AAMAS 2011

The 10th International Conference on Autonomous Agents and Multiagent Systems

May 2–6, 2011 • Taipei, Taiwan

Proceedings

Volume II

IFAAMAS

IFAAMAS
International Foundation for Autonomous Agents and Multiagent Systems
www.ifaamas.org

ISBN-10: 0-9826571-6-1
ISBN-13: 978-0-9826571-6-4

Introduction

The Autonomous Agents and MultiAgent Systems (AAMAS) conference series brings together researchers from around the world to share the latest advances in the field. It provides a marquee, high-profile forum for research in the theory and practice of autonomous agents and multiagent systems. AAMAS 2002, the first of the series, was held in Bologna, followed by Melbourne (2003), New York (2004), Utrecht (2005), Hakodate (2006), Honolulu (2007), Estoril (2008), Budapest (2009) and Toronto (2010). You are now about to enter the proceedings of AAMAS 2011, held in Taipei, Taiwan, as AAMAS celebrates its 10th anniversary as the successful merger of three related events that had run for some years previously.

In addition to the general track for the AAMAS 2011 conference, submissions were invited to three special tracks: a Robotics track, a Virtual Agents track and an Innovative Applications track. The aims of these special tracks were to give researchers from these areas a strong focus, to provide a forum for discussion and debate within the encompassing structure of AAMAS, and to ensure that the impact of both theoretical contributions and innovative applications were recognized. Each track was chaired by a leader in the field: Maria Gini for the robotics track, James Lester for the virtual agents track, and Peter McBurney for the innovative applications track. The special track chairs provided critical input to selection of Program Committee (PC) and Senior Program Committee (SPC) members, and to the reviewer allocation and the review process itself. The final decisions concerning acceptance of papers were taken by the AAMAS 2011 Program Co-chairs in discussion with, and in full agreement with the special track chairs.

Only full paper submissions were solicited for AAMAS 2011. The general, robotics, virtual agents, and innovative applications tracks received 452, 31, 51, and 41 submissions respectively, for a total of 575 submissions.

After a thorough and exciting review process, 126 papers were selected for publication as Full Papers each of which was allocated 8 pages in the proceedings and allocated 20 minutes in the Program for oral presentation. Another 123 papers were selected as Extended Abstracts and allocated 2 pages each in the proceedings. Both Full Papers and Extended Abstracts are presented as posters during the conference.

Of the submissions, more than half (338) have a student as first author, which indicates an exciting future for the field. Representation under all submissions of topics (measured by first keyword) was broad, with top counts in areas such as teamwork, coalition formation, and coordination (31), distributed problem solving (30), game theory (30), planning (26), multiagent learning (24), and trust, reliability and reputation (17).

We thank the PC and SPC members of AAMAS 2011 for their thoughtful reviews and extensive discussions. We thank Maria Gini, James Lester and Peter McBurney for making the Robotics, the Virtual Agents and the Innovative Applications tracks a success. We thank Michael Rovatsos for putting together the proceedings. Finally, we thank David Shield for his patience and support regarding Confmaster during every stage between the submission process and the actual AAMAS 2011 event. The Program represents the intellectual motivation for researchers to come together at the Conference, but the success of the event is dependent on the many other elements that make up the week especially the tutorials, workshops, and doctoral consortium. We thank all members of the Conference Organising Committee for their dedication, enthusiasm, and attention to detail, and wish to particularly thank Von-Wun Soo as Chair of the Local Organising Committee for his contributions.

Kagan Tumer and Pınar Yolum,
AAMAS 2011 Program Co-Chairs

Peter Stone and Liz Sonenberg,
AAMAS 2011 General Co-Chairs

Organizing Committee

General Chairs

Liz Sonenberg (The University of Melbourne, Australia)
Peter Stone (The University of Texas at Austin, USA)

Program Co-Chairs

Kagan Tumer (Oregon State University, USA)
Pınar Yolum (Bogazici University, Turkey)

Robotics Track Chair

Maria Gini (University of Minnesota, USA)

Virtual Agents Track Chair

James Lester (North Carolina State University, USA)

Innovative Applications Chair

Peter McBurney (University of Liverpool, UK)

Local Arrangements Chair

Von-Wun Soo (National Tsing Hua University, Taiwan)

Local Arrangements Committee

Tzung-Pei Hong (National University of Kaohsiung, Taiwan)
Churn-Jung Liau (Academia Sinica, Taiwan)
Chao-Lin Liu (National Chengchi University, Taiwan)
Soe-Tsyr Yuan (National Chengchi University, Taiwan)

Finance Chair

Nancy Reed (University of Hawaii, USA)

Publicity Chair

Iyad Rahwan (Masdar Institute, UAE)

Publications Chair

Michael Rovatsos (The University of Edinburgh, UK)

Tutorials Chair

Vincent Conitzer (Duke University, USA)

Workshops Chair

Frank Dignum (Universiteit Utrecht, Netherlands)

Exhibitions Chair

Sonia Chernova (Worcester Polytechnic Institute, USA)

Demonstrations Chair

Elizabeth Sklar (City University of New York, USA)

Scholarships Co-Chairs

Matthew E. Taylor (Lafayette College, USA)
Michael Winikoff (University of Otago, New Zealand)

Doctoral Consortium Co-Chairs

Kobi Gal (Ben-Gurion University of the Negev, Israel)
Adrian Pearce (The University of Melbourne, Australia)

Sponsorship Co-Chairs

Sherief Abdallah (British University in Dubai, UAE)
Jane Hsu (National Taiwan University, Taiwan)
Daniel Kudenko (University of York, UK)
Paul Scerri (Carnegie Mellon University, USA)

Senior Program Committee

Sherief Abdallah (British University in Dubai)
Adrian Agogino (University of California, Santa Cruz)
Stéphane Airiau (University of Amsterdam)
Francesco Amigoni (Politecnico di Milano)
Ana Bazzan (Universidade Federal do Rio Grande do Sul)
Jamal Bentahar (Concordia University)
Rafael Bordini (Universidade Federal do Rio Grande do Sul)
Cristiano Castelfranchi (ISTC-CNR)
Steve Chien (Jet Propulsion Laboratory, Caltech)
Amit Chopra (University of Trento)
Brad Clement (California Institute of Technology)
Helder Coelho (Universidade de Lisboa)
Vincent Conitzer (Duke University)
Mehdi Dastani (Utrecht University)
Keith Decker (University of Delaware)
Ed Durfee (University of Michigan)
Edith Elkind (Nanyang Technological University)
Ulle Endriss (University of Amsterdam)
Piotr Gmytrasiewicz (University of Illinois at Chicago)
Jonathan Gratch (University of Southern California)
Dominic Greenwood (Whitestein Technologies)
Dirk Heylen (University of Twente)
Koen Hindriks (Delft University of Technology)
Takayuki Ito (Nagoya Institute of Technology)
Odest Jenkins (Brown University)
Gal Kaminka (Bar Ilan University)
Jeffrey Kephart (IBM Research)
Sven Koenig (University of Southern California)
Sarit Kraus (Bar-Ilan University)
Kate Larson (University of Waterloo)
João Leite (Universidade Nova de Lisboa)
Pedro Lima (Lisbon Technical University)
Michael Luck (King's College London)
Rajiv Maheswaran (University of Southern California)
Janusz Marecki (IBM Research)
Stacy Marsella (University of Southern California)
John-Jules Meyer (Utrecht University)

Daniele Nardi (Sapienza University Roma)
Ann Nowe (Vrije Universiteit Brussel)
Ana Paiva (INESC-ID)
Simon Parsons (City University of New York)
Michal Pechoucek (Czech Technical University)
Paul Piwek (The Open University)
Helmut Prendinger (National Institute of Informatics)
Iyad Rahwan (Masdar Institute)
Mark Riedl (Georgia Institute of Technology)
Thomas Rist (University of Applied Sciences Augsburg)
Juan Antonio Rodríguez-Aguilar (IIIA-CSIC)
Alex Rogers (University of Southampton)
Jeffrey Rosenschein (Hebrew University of Jerusalem)
Jordi Sabater-Mir (IIIA-CSIC)
Erol Şahin (Middle East Technical University)
Paul Scerri (Carnegie Mellon University)
Nathan Schurr (Aptima, Inc.)
Sandip Sen (University of Tulsa)
Murat Sensoy (University of Aberdeen)
Maarten Sierhuis (Palo Alto Research Center)
Carles Sierra (IIIA-CSIC)
Munindar Singh (North Carolina State University)
Elizabeth Sklar (City University of New York)
Katia Sycara (Carnegie Mellon University)
Matthew Taylor (Lafayette College)
John Thangarajah (Royal Melbourne Institute of Technology)
Simon Thompson (BT Research and Technology)
Paolo Torroni (University of Bologna)
Karl Tuyls (Maastricht University)
Wiebe van der Hoek (University of Liverpool)
M. Birna van Riemsdijk (Delft University of Technology)
Pradeep Varakantham (Singapore Management University)
Manuela Veloso (Carnegie Mellon University)
Katja Verbeeck (Katholieke Hogeschool Sint-Lieven)
Hannes Vilhjálmsson (Reykjavik University)
Michael Wellman (University of Michigan)
Steven Willmott (3scale Networks)
Michael Wooldridge (University of Liverpool)
Neil Yorke-Smith (American University of Beirut)
R. Michael Young (North Carolina State University)
Shlomo Zilberstein (University of Massachusetts Amherst)

Program Committee

Thomas Ågotnes (University of Bergen)
Noa Agmon (University of Texas, Austin)
H. Levent Akin (Bogaziçi University)
Marco Alberti (New University of Lisbon)
Huib Aldewereld (Utrecht University)
Natasha Alechina (University of Nottingham)
Martin Allen (University of Wisconsin-La Crosse)
Christopher Amato (Aptima Inc.)
Leila Amgoud (Institut de Recherche en Informatique de Toulouse)

Bo An (University of Massachusetts Amherst)
Giulia Andrighetto (ISTC-CNR)
Luis Antunes (Universidade de Lisboa)
Alexander Artikis (NCSR Demokritos)
Itai Ashlagi (Massachusetts Institute of Technology)
Katie Atkinson (University of Liverpool)
James Atlas (University of Delaware)
Ruth Aylett (Heriot-Watt University)
Yoram Bachrach (Microsoft Research)
Byung-Chull Bae (Samsung Advanced Institute of Technology)
Quan Bai (Tasmanian ICT Centre, CSIRO)
Matteo Baldoni (University di Torino)
João Balsa (Universidade de Lisboa)
Bikramjit Banerjee (University of Southern Mississippi)
Laura Barbulescu (Carnegie Mellon University)
Cristina Baroglio (University of Torino)
Tony Barrett (Jet Propulsion Laboratory)
Christian Becker-Asano (University of Freiburg)
Reinaldo Bianchi (FEI)
Mauro Birattari (Université Libre de Bruxelles)
Olivier Boissier (Ecole des Mines de Saint-Etienne)
Andrea Bonarini (Politecnico di Milano)
Tibor Bosse (Vrije Universiteit Amsterdam)
Luis Botelho (Instituto Universitŕio de Lisboa)
Sylvain Bouveret (ONERA)
Emma Bowring (University of the Pacific)
Ronen Brafman (Ben Gurion University)
Lars Braubach (University of Hamburg)
Joost Broekens (Delft University of Technology)
Brett Browning (Carnegie Mellon University)
Paul Buhler (College of Charleston)
Bernard Burg (Panasonic Laboratories)
Juan Burguillo (University of Vigo)
Birgit Burmeister (Daimler AG)
Lucian Busoniu (Delft University of Technology)
Zhongtang Cai (Oracle)
Daniele Calisi (Sapienza University of Rome)
Monique Calisti (Martel Consulting)
Tran Cao Son (New Mexico State University)
Javier Carbo (University Carlos III)
Alan Carlin (University of Massachusetts)
Stefano Carpin (University of California, Merced)
José Cascalho (Universidade dos Açores)
Marc Cavazza (University of Teesside)
Jesus Cerquides (IIIA-CSIC)
Brahim Chaib-draa (Laval University)
Georgios Chalkiadakis (University of Southampton)
Wei Chen (Intelligent Automation)
Xiaoping Chen (University of Science and Technology of China)
Shih-Fen Cheng (Singapore Management University)
Yun-Gyung Cheong (IT University of Copenhagen)
Sonia Chernova (Worcester Polytechnic Institute)
Carlos Iván Chesñevar (Universidad Nacional del Sur)
Federico Chesani (University of Bologna)
Maria Chli (Aston University)

Robin Cohen (University of Waterloo)
Nikolaus Correll (University of Colorado, Boulder)
Jacob Crandall (Masdar Institute)
Dominik Dahlem (Massachusetts Institute of Technology)
Esther David (Ashkelon College, Israel)
Célia da Costa Pereira (Université de Nice Sophia-Antipolis)
Antônio Carlos da Rocha Costa (Universidade Federal do Rio Grande)
Paulo Pinheiro da Silva (University of Texas, El Paso)
Scott DeLoach (Kansas State University)
Yves Demazeau (LIG-CNRS)
Louise Dennis (University of Liverpool)
Patrick De Causmaecker (Katholieke Universiteit Leuven)
Steven de Jong (Maastricht University)
Stefan De Wannemaecker (Katholieke Universiteit Leuven)
Mathijs de Weerdt (Delft University of Technology)
Mary Bernardine Dias (Carnegie Mellon University)
Frank Dignum (Utrecht University)
Virginia Dignum (Delft University of Technology)
Oğuz Dikenelli (Ege University)
Jürgen Dix (Clausthal University of Technology)
Dmitri Dolgov (Google)
Klaus Dorer (Offenburg University)
Prashant Doshi (Univ of Georgia)
Paul Dunne (University of Liverpool)
Marc Esteva (IIIA-CSIC)
Piotr Faliszewski (AGH University of Science and Technology)
Alessandro Farinelli (University of Verona)
Shaheen Fatima (Loughborough University)
Sevan Ficici (Natural Selection)
Felix Fischer (Harvard SEAS)
Klaus Fischer (DFKI)
Nicoletta Fornara (Universita della Svizzera Italiana)
Alex Fukunaga (University of Tokyo)
Naoki Fukuta (Shizuoka University)
Alberto Valero Gómez (University Carlos III de Madrid)
Thomas Gabel (Albert-Ludwigs-University Freiburg)
Kobi Gal (Ben-Gurion University of the Negev)
Nicola Gatti (Politecnico di Milano)
Patrick Gebhard (DFKI)
Enrico Gerding (University of Southampton)
Aditya K. Ghose (University of Wollongong)
Marco Gilles (Goldsmiths, University of London)
Paolo Giorgini (University of Trento)
Andrea Giovannucci (Princeton University)
Claudia Goldman (General Motors, Israel)
Valentin Goranko (Technical University of Denmark)
Guido Governatori (NICTA)
Adela Grando (University of Edinburgh)
Gianluigi Greco (University of Calabria)
Rachel Greenstadt (Drexel University)
Nathan Griffiths (University of Warwick)
Davide Grossi (University of Amsterdam)
Marek Grzes (University of Waterloo)
Mingyu Guo (University of Liverpool)
Christian Guttmann (EBTIC)

Jomi Hübner (Federal University of Santa Catarina)
James Hanson (IBM Research)
James Harland (Royal Melbourne Institute of Technology)
Paul Harrenstein (Technische Universität München)
Hiromitsu Hattori (Kyoto University)
Christopher Hazard (North Carolina State University)
Tarek Helmy (King Fahd University of Petroleum and Mineral)
Annerieke Heuvelink (TNO, Netherlands)
Sarah Hickmott (RMIT University)
Martin Hofmann (Lockheed Martin)
Mark Hoogendoorn (Vrije Universiteit)
Ian Horswill (Northwestern University)
Kaijen Hsiao (Willow Garage)
Michael Huhns (University of South Carolina)
Joris Hulstijn (Vrije Universiteit Amsterdam)
Luca Iocchi (Sapienza University Roma)
Michal Jakob (FEE Czech Technical University)
Wojciech Jamroga (University of Luxembourg)
Gaya Jayatilleke (Royal Melbourne Institute of Technology)
Arnav Jhala (University of California Santa Cruz)
Catholijn Jonker (Delft University of Technology)
Meir Kalech (Ben-Gurion University)
Marcelo Kallmann (University of California, Merced)
Ece Kamar (Microsoft Research)
Sachin Kamboj (University of Delaware)
Georgia Kastidou (University of Waterloo)
Takahiro Kawamura (Toshiba)
Michael Kipp (DFKI)
Alexandra Kirsch (Technische Universität München)
Franziska Klügl (Örebro University)
Alexander Kleiner (University of Freiburg)
Tomas Klos (Delft University of Technology)
Matthias Klusch (DFKI)
Matthew Knudson (Oregon State University)
Robert Kohout (DARPA)
Martin Kollingbaum (University of Aberdeen)
Sebastien Konieczny (CRIL-CNRS)
Stefan Kopp (Bielefeld University)
Gerhard Kraetzschmar (Bonn-Rhine-Sieg University of Applied Science)
Emiel Krahmer (Tilburg University)
Brigitte Krenn (Austrian Research Institute for Artificial Intelligence)
Daniel Kudenko (University of York)
Ugur Kuter (University of Maryland)
Miguel Ángel López Carmona (Universidad de Alcala)
Michail Lagoudakis (Technical University of Crete)
Sebastien Lahaie (Yahoo! Research)
Luis Lamb (UFRGS)
Martin Lauer (Karlsruher Institut für Technologie)
Alessandro Lazaric (SequeL)
Samuel Leong (Microsoft Research)
Yves Lespérance (York University)
Victor Lesser (University of Massachusetts, Amherst)
Maxim Likachev (Carnegie Mellon University)
Wei Liu (University Western Australia)
Yaxin Liu (Google)

Brian Logan (University of Nottingham)
Alessio R. Lomuscio (Imperial College London)
Emiliano Lorini (Institut de Recherche en Informatique de Toulouse)
Bryan Kian Hsiang Low (National University of Singapore)
Zakaria Maamar (Zayed University)
Brian Magerko (Georgia Institute of Technology)
Roger Mailler (University of Tulsa)
Wenji Mao (Chinese Academy of Sciences)
Vangelis Markakis (Athens University of Economics and Business)
Lino Marques (University of Coimbra)
Viviana Mascardi (Universita' degli Studi di Genova)
Shigeo Matsubara (Kyoto University)
Tokuro Matsuo (Yamagata University)
Nicolas Maudet (University Paris-Dauphine)
Francisco Melo (INESC-ID/Instituto Superior Técnico)
Felipe Meneguzzi (Carnegie Mellon University)
Pedro Meseguer (IIIA CSIC)
Tomasz Michalak (University of Southampton)
Martin Michalowski (Adventium Labs)
Simon Miles (King's College London)
Dejan Milutinovic (University of California Santa Cruz)
Sanjay Modgil (King's College London)
Iqbal Mohomed (IBM Research)
Luis Moniz (Universidade de Lisboa)
Marco Montali (University of Bologna)
Bradford Mott (North Carolina State University)
Abdel-Illah Mouaddib (University of Caen Basse-Normandie)
Hideyuki Nakanishi (Osaka University)
Yukiko Nakano (Seikei University)
Nanjangud Narendra (IBM Research)
Toyoaki Nishida (Kyoto University)
Pablo Noriega (IIIA-CSIC)
Timothy Norman (University of Aberdeen)
Colm O'Riordan (National University of Ireland)
Magalie Ochs (CNRS)
Jean Oh (Carnegie Mellon University)
Frans Oliehoek (Massachusetts Institute of Technology)
Nir Oren (University of Aberdeen)
Mehmet Orgun (Macquarie University)
Charlie Ortiz (SRI International)
Sarah Osentoski (Brown University)
Sascha Ossowski (University Rey Juan Carlos)
Liviu Panait (Google)
Mario Paolucci (ISTC-CNR)
Dmitrii Pasechnik (Nanyang Technological University)
Terry Payne (University of Liverpool)
Catherine Pelachaud (Telecom ParisTech)
Johannes Pellenz (Federal Office of Defence Technology)
Marek Petrik (IBM Research)
Stacy Pfautz (Aptima)
Maria Silvia Pini (University of Padova)
Michael Pirker (Siemens)
Jeremy Pitt (Imperial College London)
Alexander Pokahr (University of Hamburg)
Daniel Polani (University of Hertfordshire)

Faruk Polat (Middle East Technical University)
Maria Polukarov (University of Southampton)
Enrico Pontelli (New Mexico State University)
Ronald Poppe (University of Twente)
Daniele Porello (University of Amsterdam)
Han La Poutré (Centrum Wiskunde and Informatica)
Rui Prada (INESC-ID and Instituto Superior Técnico)
Henry Prakken (Utrecht University)
Doina Precup (McGill University)
Ariel D. Procaccia (Harvard University)
Scott Proper (Oregon State University)
David Pynadath (University of Southern California)
Michael Quinlan (Univeristy of Texas, Austin)
Anita Raja (University of North Carolina at Charlotte)
Célia Ghedini Ralha (University of Brasília)
Sarvapali Ramchurn (University of Southampton)
Matthias Rehm (Aalborg University)
Dennis Reidsma (University of Twente)
Fenghui Ren (University of Wollongong)
Marcello Restelli (Politecnico di Milano)
Fernando Ribeiro (Universidade do Minho)
Alessandro Ricci (University of Bologna)
Debbie Richards (Macquarie University)
Giovanni Rimassa (Whitestein Technologies AG)
David Roberts (North Carolina State University)
Valentin Robu (University of Southampton)
Odinaldo Rodrigues (King's College London)
Nico Roos (Maastricht University)
Antonino Rotolo (University of Bologna)
Michael Rovatsos (University of Edinburgh)
Zachary Rubinstein (Carnegie Mellon University)
Wheeler Ruml (University of New Hampshire)
Vasile Rus (University of Memphis)
Zsófia Ruttkay (Moholy-Nagy University of Arts and Design Budapest)
Fariba Sadri (Imperial College London)
Alessandro Saffiotti (Orebro University)
Ken Satoh (National Institute of Informatics, Japan)
Martijn Schut (Vrije Universiteit)
Steven Shapiro (RMIT University)
Alexei Sharpanskykh (Vrije Universiteit Amsterdam)
Onn Shehory (IBM Research)
Dylan Shell (Texas A&M University)
Jiaying Shen (SRI International)
Mei Si (Rensselaer Polytechnic Institute)
Marius Silaghi (FIT)
Barry G. Silverman (University Pennsylvania)
Guillermo Simari (Universidad Nacional del Sur)
Stephen Smith (Carnegie Mellon University)
Leen-Kiat Soh (University of Nebraska-Lincoln)
Matthijs Spaan (Instituto Superior Técnico)
Mudhakar Srivatsa (IBM Research)
Jordan Srour (American University of Beirut)
Eugen Staab (IMC)
Sebastian Stein (University of Southampton)
Gerald Steinbauer (Graz University of Technology)

Bas Steunebrink (Dalle Molle Institute for Artificial Intelligence)
Nathan Sturtevant (University of Denver)
Gita Sukthankar (University of Central Florida)
Evan Sultanik (Drexel University)
Pedro Szekely (USC Information Sciences Institute)
Juan Tapiador (University of York)
Luke Teacy (University of Ulster)
Andrea Tettamanzi (Università degli Studi di Milano)
Michael Thielscher (The University of New South Wales)
Andrea Thomaz (Georgia Institute of Technology)
Ingo Timm (University of Trier)
Viviane Torres da Silva (Universidade Federal Fluminense)
Jan Treur (Vrije Universiteit Amsterdam)
Paulo Trigo (Instituto Superior de Engenharia de Lisboa)
Nicolas Troquard (University of Liverpool)
Khiet Truong (University of Twente)
Takahiro Uchiya (Nagoya Institute of Technology)
Joel Uckelman (University of Amsterdam)
Paulo Urbano (FCUL)
Greet Vanden Berghe (Katholieke Hogeschool Sint-Lieven)
Kees van Deemter (University of Aberdeen)
Ielka van der Sluis (Trinity College Dublin)
Leendert van der Torre (University of Luxembourg)
Jurriaan van Diggelen (TNO, The Netherlands)
Betsy van Dijk (University of Twente)
Hans van Ditmarsch (University of Sevilla)
H. Van Dyke Parunak (Vector Research Center)
Rogier van Eijk (Utrecht University)
Wamberto Vasconcelos (University of Aberdeen)
Kristen Brent Venable (University of Padova)
Laurent Vercouter (Ecole des Mines de Saint-Etienne)
Jose Vidal (University of South Carolina)
Vinoba Vinayagamoorthy (BBC Research & Development)
Ubbo Visser (University of Miami)
Yevgeniy Vorobeychik (Sandia National Labs)
George Vouros (University of the Aegean)
Peter Vrancx (Vrije Universiteit Brussel)
Yonghong Wang (Carnegie Mellon University)
Nick Webb (University at Albany)
Gerhard Weiss (University of Maastricht)
Danny Weyns (Katholieke Universiteit Leuven)
Michael Winikoff (University of Otago)
Cees Witteveen (Delft University of Technology)
Yang Xu (University Electronic Science and Tech of China)
Osher Yadgar (SRI International)
Hamdi Yahyaoui (Kuwait University)
Hirofumi Yamaki (Nagoya University)
Gaku Yamamoto (IBM Japan)
William Yeoh (University of Massachusetts)
Makoto Yokoo (Kyushu University)
Muhammad Younas (Oxford Brookes University)
Jie Zhang (Nanyang Technological University)
Minjie Zhang (The University of Wollongong)
Rong Zhou (PARC)
Ingrid Zukerman (Monash University)

Auxiliary Reviewers

John Augustine
Azizi ab Aziz
Haris Aziz
Aijun Bai
Tim Baarslag
Alexandros Belesiotis
Elizabeth Black
Thomas Bolander
Fiemke Both
Christoph Broschinski
Hendrik Buschmeier
Laurent Charlin
George Christelis
Evan Clark
Matt Crosby
Phan Minh Dung
Eliseo Ferrante
Francesco Figari
Jan-Gregor Fischer
Alexander Grushin
David Ben Hamo
Andreas Hertle
Greg Hines
Yazhou Huang
S. Waqar Jaffry
Thomas Keller
Eliahu Khalastchi
Yoonheui Kim
Ramachandra Kota
Rianne van Lambalgen
Steffen Lamparter
Viliam Lisý
Mentar Mahmudi
Robbert-Jan Merk
João Messias
Victor Naroditskiy
Christian Pietsch
Giovanni Pini
Matthijs Pontier
Evangelia Pyrga
Shulamit Reches
Inmaculada Rodriguez
Amir Sadeghipour
Hans Georg Seedig
Becher Silvio
Alexander Skopalik
Roni Stern
Ali Emre Turgut
Iris van de Kieft
Natalie van der Wal
Meritxell Vinyals
Wietske Visser
Grant Weddell
Matthew Whitaker
Simon Williamson
Feng Wu
Jiongkun Xie
Ramind Yaghubzadeh
Zongzhang Zhang

Sponsors

We would like to thank the following for their contribution to the success of this conference:

The International Foundation for Autonomous Agents and Multiagent Systems

Platinum Sponsor

Artificial Intelligence Journal

Gold Sponsor

Agreement Technologies

Asian Office of Aerospace Research & Development

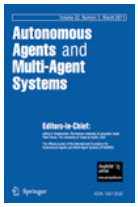
Silver Sponsors

Etisalat British Telecom Innovation Centre

Foundation for Intelligent Physical Agents

IBM Research

Journal of Autonomous Agents and Multi-Agent Systems



Wiley-Blackwell

Other Sponsors

IOS Press

Local Sponsors

National Science Council

Ministry of Education,
Republic of China (Taiwan)

Ministry of Foreign Affairs,
Republic of China (Taiwan)

National Tsing Hua University

Taiwanese Association of Artificial Intelligence

Contents

Session B4 – Game Theory and Learning

Session C4 – Teamwork

Session A5 – Learning Agents

Session B5 – Auction and Incentive Design

Session C6 – Voting Protocols

Session D6 – Trust and Organisational Structure

Session A7 – Argumentation and Negotiation

Session B7 – Planning

Session C7 – Game Theory II

Session D7 – Virtual Agents II

Main Program – Extended Abstracts

Red Session

Blue Session

Green Session

Demonstrations

Doctoral Consortium Abstracts

Virtual Agents I

Culture-related differences in aspects of behavior for virtual characters across Germany and Japan

Birgit Endrass
Elisabeth André
Human Centered Multimedia
Augsburg University
Universitätsstr. 6a
D-86159 Augsburg, Germany
endrass,
andre@hcm-lab.de

Matthias Rehm
Department of Media Technology
Aalborg University
Niels-Jernes Vej 14
DK-9220 Aalborg, Denmark
matthias@imi.aau.dk

Afia Akter Lipi
Yukiko Nakano
Dept. of Computer and Information Science
Seikei University
Musashino-shi, Tokyo,
180-8633 Japan
y.nakano@st.seikei.ac.jp
afiaakhter@hotmail.com

ABSTRACT

Integrating culture as a parameter into the behavioral models of virtual characters to simulate cultural differences is becoming more and more popular. But do these differences affect the user's perception? In the work described in this paper, we integrated aspects of non-verbal behavior as well as communication management behavior into the behavioral models of virtual characters for the two cultures of Germany and Japan in order to find out which of these aspects affect human observers of the target cultures. We give a literature review pointing out the expected differences in these two cultures and describe the analysis of a multi-modal corpus including video recordings of German and Japanese interlocutors. After integrating our findings into a demonstrator featuring a German and a Japanese scenario, we presented the virtual scenarios to human observers of the two target cultures in an evaluation study.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*; I.6.7 [**Simulation and Modeling**]: Model Development

General Terms

Human Factors, Design, Experimentation

Keywords

Virtual Agents, Multiagent Systems, Culture, Communication Management, Nonverbal Behavior

1. MOTIVATION

A vast part of our communication happens non-verbally. While we might be thinking about what we want to communicate verbally, we manage our non-verbal behavior mostly subconsciously. Thereby, we integrate our personality, emotional state and cultural background into our behavior. How this behavior is interpreted depends on the listener's social and personal background as well. Enormous effort has been done so far in integrating these personal or social factors into the behavior models of virtual characters.

Culture has come in focus lately as another important factor that influences the success of an interaction with a virtual character. How different culture-specific behavior patterns of virtual characters are perceived and interpreted across different cultures has not been studied so far. In this paper, we integrated findings about such culture-specific behavior patterns into the behavior model of virtual characters. Our main goal is to find out which aspects of behavior result in a positive or negative impression on the user. Thus, the interpretation of different behavioral aspects is tested in isolation, making use of the same underlying dialog. For the implementation and evaluation, we choose the basic behavioral dimensions of communication management in terms of pauses and overlaps between turns as well as gestural expressivity and body posture. The former have been shown to be basic structuring mechanisms for face to face communications [9], the latter have been shown to differ broadly between cultures [5]. In addition, all have been attributed as provoking misunderstandings in inter-cultural communications [27].

For that task, several challenges had to be solved. A standardized video corpus was collected in the participating cultures [26] and the data was analyzed in the target cultures simultaneously with equal quality [24], [8] and [20]. In order to integrate the findings into a multiagent system, on the one hand the virtual characters' appearances have to be adapted to their cultural background on the other hand different behavioral models have to be built in order to match the cultural-background. To evaluate these models, studies have been set up in the participating cultures. In our previous work, we concentrated on either the analysis of behavioral differences or evaluation studies in only one culture. The aim of this paper is to find out which behavioral aspects have an effect on the perception of human observers of the two target cultures and whether participants prefer agent behavior that was designed for their own cultural background.

This paper is organized as follows: In the following section (Section 2), we discuss related work in the research

Cite as: Culture-related differences in aspects of behavior for virtual characters across Germany and Japan, B. Endrass, M. Rehm, A.A. Lipi, Y. Nakano and E. André, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 441-448.

field of integrating culture into virtual agent applications. In the next chapter (Section 3), we introduce some theoretical background and state our expectations about differences in behavioral aspects for the two cultures of Germany and Japan drawn from the literature. In Section 4, we describe a video corpus, that was recorded in the above mentioned cultures as well as our analysis of culture-related differences. We focus on the above-mentioned basic behavioral dimensions of gestures, postures and communication management. Then, we describe the integration of our findings into a demonstrator (Section 5). Section 6 then gives details on our study, where we evaluated whether participants have preferences for agent behavior that was designed to match their own cultural background, before we conclude the paper (Section 7).

2. RELATED WORK

The aim of the work described in this paper is to integrate different aspects of culture-specific behaviors into a multiagent system in order to find out which behaviors affect the user's perceptions. In the following we summarize some related work on integrating culture into the behavioral models of virtual characters.

Only a few attempts have been made to integrate the aspect of culture into the behavioral models of virtual characters. An example includes the Tactical Language Training System (TLTS) by Johnson and colleagues [16]. In order to complete the tasks provided by the system, trainees have to learn a foreign language. So far, four versions of TLTS have been implemented: Iraqi, Dari, Pashto, and French [17]. Through interaction with the people in the virtual world, the learner is supposed to develop cultural sensitivity.

Aylett et al. [1] introduce an educational application that uses fantasy characters in order to develop intercultural empathy. Culture-related differences are expressed through different symbols and rituals. The agents adapt their behavior in a culture-specific way and interpret incoming events according to cultural background. Our aim, however, is the simulation of behavioral aspects in existing national cultures in order to find out which patterns affect the user's perception.

An approach that focuses on the perception of virtual characters simulating synthetic cultures is presented in Mascarenhas et al. [21]. For their simulation, two different groups of characters were created that differed in their rituals and cultural dimensions. A perception study showed that the subjects found significant differences in the cultures and were able to relate these differences to the phenomenon of culture.

Focusing on the different perception of virtual characters' appearances across cultures, Koda et al. [18] designed culture-specific comic-style agents to show different emotions to subjects from different cultures. The characters were perceived differently across cultures and emotions were interpreted more correctly in the corresponding culture. In [19], Koda et al. have a closer look at different regions of the face and conducted a cross-cultural study in Hungary and Japan in order to test the impact of facial regions as cues to recognize the emotions of virtual agents. In their results the authors report that Japanese subjects found facial cues in the eye region more important than Hungarians subjects, who vice versa concentrated more on facial cues in the mouth region.

An evaluation study that investigates the different perception of verbal and non-verbal behaviors is introduced by Iacobelli et al. [14]. In their work, the authors focus on ethnicity, by changing behaviors of the character and leaving the appearance constant. Ethnic identity and engagement were evaluated and their results reveal that users were able to relate the virtual agents correctly. This inspired our research and brought up the question which of the behaviors we plan on integrating affects the user's perception most.

An approach that deals with non-verbal behavior is presented in [15]. Jan et al. simulate cultural differences in non-verbals such as proxemics and gaze. In a user study, the authors evaluated whether their participants perceived differences between behaviors associated with their own cultural background and behaviors simulating a different cultural background. In a similar manner, we want to find out if users from Germany and Japan prefer behaviors that are built to match their own cultural background for the aspects of communication management behaviors, gestural expressivity and posture.

In the CUBE-G project [23], we aim on the integration of culture-specific behaviors for interaction with embodied conversational agents in order to build a training scenario for human users. Therefore, culture-specific behavior has been analyzed in a video corpus. So far, we analyzed non-verbal behaviors [25] and communication management behaviors [8] and integrated our findings into a demonstrator featuring a German and a Japanese dialog scenario. Furthermore, the impact of these behavioral differences has been partly evaluated by German observers. The studies showed that German participants preferred the German communication management scenario over the Japanese scenario. Japanese participants have not been considered yet. For non-verbal behavior, the question of whether observers prefer non-verbal behaviors in virtual scenarios that correspond to their own culture remaines still unanswered. The aim of this paper is to correlate the results in non-verbal behaviors and communication management behaviors with an evaluation study in both participating cultures.

3. THEORETICAL BACKGROUND

As we stated above, we are looking at different aspects of behavior in the two cultures of Germany and Japan. In particular, we focus on communication management behaviors (pauses in speech and overlaps), body posture and gestural expressivity. In this section, we introduce these behaviors and state our expectations in culture-related differences drawn from the literature.

As one aspect of behavior that might affect the perception of a particular conversation, we had a closer look at communication management behaviors. So-called regulators are used in order to manage communication [27]. *Vocalics* include verbal feedback signals (such as "uh-huh") as well as the usage of silence in speech or interruptions of the communication partner. Depending on the usage of these vocalics, a different rhythm of speech can evolve. *Kinesics* and *oculesics* comprise non-verbal regulators. According to [27], communication can be managed though hand gestures and body postures (kinesics) or eye and face gaze (oculesics).

These regulators are used to control the flow and pauses of a conversation and are considered culture-specific behaviors. In addition, regulators are used at a very low level of awareness since they are learned at a very young age [27].

Table 1: Hofstede's scores on the five dimensions of culture for the two cultures of Germany and Japan as well as the world average.

Culture / Dimension	Germany	World Average	Japan
PDI	35	55	54
IDV	67	64	46
MAS	66	48	95
UAI	65	61	92
LTO	31	41	80

We therefore consider regulators as an interesting aspect of behavior that might have an effect on the perception of a given conversation depending on the culture of the listener. This is in line with Ting-Toomey [27], who states that a discriminative use of regulators can cause intercultural distress or misunderstandings.

Another interesting aspect of behavior is the expressivity of non-verbal behaviors. How we exhibit a gesture can sometimes be more crucial for the observer's perception than the gesture itself. Differences in the dynamic variation can be described according to expressivity parameters [22]. The *spatial extend*, for example, describes the arm's extend toward the torso. The *speed* of a gesture and the *power* of the arm can vary as well. The *fluidity* parameter describes the continuity between consecutive gestures, while the *repetitivity* holds information about the repetition of the stroke. The last expressivity parameter, *overall activation*, counts the amount of gestures that are performed. How gestures are executed can depend on several individual and social factors such as personality, emotional state or culture.

Next in this study, we examined posture as another kind of non-verbal behavior. Posture is defined as a motion or position shift of the human body [3]. Based on previous studies, we defined four parameters to describe the characteristics of postures. The four parameters are *duration* till which a person remains in the same posture, *spatial extent* used in a posture, *rigidness* or relaxation apparent from the posture and *mirroring* as number of instances when an individual unconsciously imitates a partner's posture during a conversation. We already found that these parameters are useful in describing the culture variations in postures [20].

3.1 Culture-specific expectations

In the social sciences, culture is a well established research field. There are several approaches that define culture and describe differences in their behavior. A well-known model of culture was introduced by Hofstede [12], who built a five dimensional model in order to distinguish cultures. Over 20 different cultures were categorized in a broad empirical survey. Table 1 shows the scores of the two cultures of Germany and Japan, as published on Hofstede's web page [11]. Please note that these scores were normalized across cultures to stay between 0 and 100 in the first version and extended later, when more cultures were added and more extreme values were observed.

The Power Distance dimension (PDI) describes the extent to which a different distribution of power is accepted by the less powerful members of a culture. The Individualism dimension (IDV) describes the degree to which individuals are integrated into a group. On the individualist side ties between individuals are loose, and everybody is expected to take care for him- or herself. On the collectivist side, people are integrated into strong, cohesive in-groups. The gender or masculinity dimension (MAS) describes the distribution of roles between the genders and how masculine values are perceived. In feminine cultures, the roles differ less than in masculine cultures, while competition is rather accepted in masculine cultures and status symbols are of importance. In the uncertainty avoidance dimension (UAI), the tolerance for uncertainty and ambiguity is defined. It indicates to what extent the members of a culture feel either comfortable or uncomfortable in unstructured or unknown situations. The long-term orientation dimension (LTO) has been added afterwards, in order to explain differences between Asian and Western cultures. Values for long term orientation are, for example, thrift and perseverance; whereas examples for values for the short term orientation are respect for tradition, fulfilling social obligations, and saving one's face.

The positioning on these dimensions affects one's behavior. Taking a look at the cultural dimensions in isolation, Hofstede [13] introduces so-called synthetic cultures that find themselves on one of the extreme ends of each dimension. For these synthetic cultures he describes prototypical behavior norms. For the behavioral aspects investigated in our research, the individualism dimension and the power distance dimension are of special interest.

For collectivistic cultures, he states that silence may occur in conversations without creating tension. This observation does not hold true for individualistic cultures. In addition, he states that the usage of pauses can be a crucial feature in collectivistic cultures. Germany is a more individualistic culture than Japan (see Table 1, IDV). As a consequence, it should be more likely in the German culture that pauses in a conversation create tension and are thus tried to be avoided. In Japanese conversations, on the other hand, pauses can be considered a feature of the conversation.

Another behavioral aspect is affected by the power distance dimension. High-power cultures are described as verbal, soft-spoken and polite and interpersonal synchrony is much more important than in low-power cultures, whose members tend to talk freely in any social context [27]. One possibility to achieve interpersonal synchrony in a conversation is giving feedback. This feedback often occurs during the speaking floor of the interlocutor. This should occur more often in the Japanese culture due to their higher value on the power distance dimension (see Table 1, PDI). The individualism dimension is also related to the expression of emotions and the acceptable emotional displays in a culture. In individualistic cultures it is more acceptable to publicly display emotions than it is in collectivistic cultures [6]. This also suggests that non-verbal behavior is expressed more emotional in German conversations than in Japanese ones. We expect displaying emotions more obviously should affect the expressivity of gestures in a way that parameters such as speed, power or spatial extent are increased for a higher arousal in emotion.

Strengthening our expectations about the usage of silence in speech and overlapping speech, Ting Toomey [27] states that the beliefs expressed in talk and silence are culture-dependent. Following Hall's categorization of cultures [10] into high- and low context communication cultures, Ting Toomey [27] observes that conversation in high context com-

munication cultures relies mainly on physical context. Meaning can be transported through non-verbal cues, such as pauses, silence and prosody. In contrast, low context communication cultures tend to explicitly code information. Clear descriptions and a high degree of specificity are used commonly in these cultures. Germany is described as one of the most extreme low context cultures, while Japan finds itself on the extreme high context side [27]. Thus, communication management behaviors such as pauses in speech or overlapping speech, should occur more frequently in Japanese conversations. Verbal feedback is given in every culture but the meaning can vary with the communicative function expressed in the feedback. In Japanese conversations, for example, communication partners explicitly communicate that they are listening by using the utterance "hai hai", while the literal translation "yes - yes" would communicate more than that. Frequency and positioning of pauses and overlaps can vary across cultures, too. Overlapping speech is often considered as impolite. But feedback utterances are often performed while it is still the interlocutor's turn without wanting to gain the turn. As we stated above, acknowledgments are very common in Japanese conversations. Thus, we expect a high amount of overlapping speech in Japanese conversations that are short but frequent. In addition, Ting Toomey [27] states that silence serves as a critical communication-device in Japanese communication patterns. Pauses reflect the thoughts of the speaker and can contain strong contextual meaning.

Similar findings are described by Trompenaars and Hampden-Turner [28], who divide cultures into Western, Latin and Oriental cultures. While Germany is considered a Western culture, Japan would count as an Oriental culture (including Asian cultures). In line with Hofstede and Ting-Toomey, Trompenaars and Hampden-Turner describe Western cultures as verbal and state that their members get nervous when there are long pauses. In addition, they state that interruptions are considered as impolite. Thus, communication in Western cultures is managed as follows: interlocutors start talking after the other conversation partner stopped. In Oriental cultures silence is more important and can be considered a sign of respect. Pauses are used to process information or assure that the conversation partner gives away the speaking floor.

Summarizing our culture-specific expectations drawn from the literature, we expect more pauses in speech and overlapping speech such as in feedback behavior in Japanese conversations than in German ones. Gestures and postures should be more expressive in prototypical German behavior than in prototypical Japanese behavior.

4. EMPIRICAL VERIFICATION

Behavioral tendencies described in the literature are sometimes rather abstract. As we stated above, we expect more pauses in speech in Japanese conversations than in German ones, for example. In order to integrate our expectations into the behavior model of virtual characters, we need more details such as number or length of pauses. To answer these and other questions, we recorded and analyzed a video corpus in the two target cultures (see [23]). Three prototypical interaction scenarios were videotaped, while more than 20 subjects participated in each of the two cultures. In a total, around 20 hours of video material were collected. Subjects interacted with actors whom they did not know in advance in order to ensure that all subjects meet the same conditions and that all scenarios last for about the same time. For the first scenario, participants were asked to get acquainted with one another since they had to solve a task together later. Recordings started already during this time. The analysis described in the next section focuses on this first time meeting scenario, which lasted for around 5 minutes for each subject.

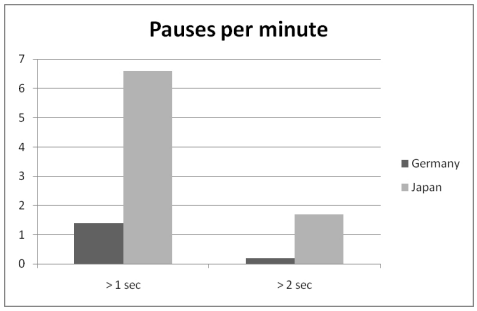
4.1 Analysis

As we stated above, we concentrate on several aspects of behavior such as the usage of pauses, overlapping speech, gestural expressivity and posture. The corpus described above was analyzed in order to find culture-related differences in these aspects [8] [24]. In the following section, we summarize our results:

For the analysis of pauses in speech, we considered as a pause the parts of the conversation where none of the conversation partners spoke and took into account the pauses that lasted for more than one second and more than two seconds respectively. In that manner, we sorted out very brief pauses that are used for breathing for example. Comparing the two cultures, we found more pauses in speech in the Japanese conversations. In the German videos, we found on average 7.1 pauses that lasted for more than one second and 1.3 pauses on average that lasted for more than 2 seconds. In the Japanese videos, we observed 31 pauses on average that lasted over 1 second and 8.4 pauses that lasted for more than 2 seconds. Figure 1 (left) shows the distribution of short (more than 1 second) and long pauses (more than 2 seconds) that were found on average per minute in each video. Comparing the amount of pauses in speech across the two cultures, using the two sided t-test, we achieved significance for both, pauses that last for more than 1 second ($p < 0.001$) and pauses that last for more than 2 seconds ($p < 0.001$).

Regarding overlapping speech, we considered time spans where both conversation partners spoke at the same time as overlapping speech. Pragmatics, such as using overlaps for feedback behavior, were not taken into account yet. The average occurrences of overlapping speech per subject per minute for the two cultures are shown in Figure 1 (right). We observed 6 overlaps per minute in German conversations on average, while in Japanese conversations 9 overlaps per minute occurred on average. Comparing the frequency of overlapping speech across the two cultures, we achieved significant results for the total amount of overlaps ($p = 0.04$). No significance was achieved for overlaps that last for more than 0.5 seconds ($p = 0.31$) and 1 second ($p = 0.12$). By trend, we observed more overlaps in the Japanese conversations for all lengths, which is in line with our expectations described above.

As we stated above, we analyzed gestures according to expressivity parameters (see Section 3). Each parameter was coded using a seven-point scale. Analyzing the two cultures, we found significant differences for all parameters (using ANOVA with $p < 0.01$ for all parameters). Figure 2 (left) shows the average ratings of the expressivity parameters for the two cultures of Germany and Japan. Gestures were performed faster and more powerfully in the German videos than in the Japanese one's. In addition, German subjects used wider space for their gestures compared to Japanese subjects who used less space. Gestures were also performed



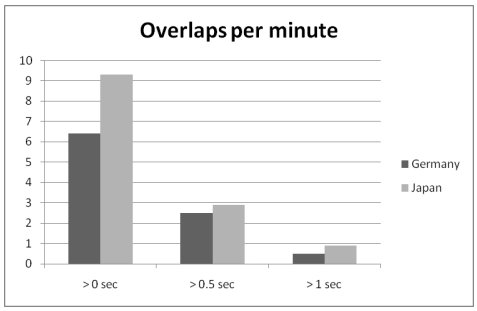
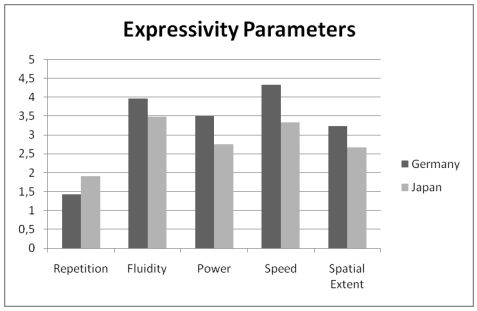


Figure 1: Pauses (left) and overlaps in speech (right) per minute, averaged over participants.



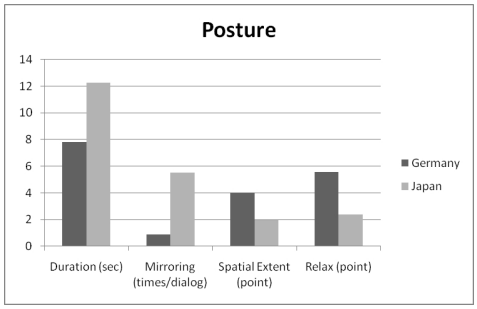


Figure 2: Ratings of expressivity parameters (left) and posture characteristics (right) in German and Japanese culture.

more fluently in the German conversations and the stroke of a gesture was repeated less in the Japanese conversations. For the analysis of posture, we used Bull's coding scheme [2] to label the posture type/shape. Figure 2 (right) shows the arm posture changes that were extracted from studying the corpus data of German and Japanese subjects. The value for duration was derived by calculating the average number of posture shifts observed in the data. To get the score for mirroring, we looked at the total number of common posture shapes of both interactors in each turn. The value for spatial extent and rigidness were assigned based on the average of 7 point scale ratings done by multiple annotators. We used the opposite word relax instead of rigidness to make the word easy to understand. Figure 2 (right) indicates that Japanese subjects remained in the same posture longer, engage in more frequent mirroring, take up less space, and display a more rigid posture in comparison to German subjects.

The postures most frequently observed in the German videos (folding the arms in front of the trunk (FAs) and putting the hands in the pockets of the trousers (PHIPt)) and in the Japanese videos (joining both hands in front of the body (JHs)) are exemplified in Figure 3 (left and middle). It is notable that ratings for postures frequently observed in the German corpus such as PHIPt and FAs were rated higher in spatial extent and lower in rigidness, compared to postures frequently observed in the Japanese data such as JHs and PHB (put hands back). Details of how values of each of the posture traits in relation to culture were obtained, are provided in [20].

5. SIMULATION

In order to simulate the behavioral tendencies described in the literature and verified by our empirical corpus study for the German and the Japanese cultures, we use the Virtual Beergarden scenario [4]. In the scenario, an arbitrary number of agents can be loaded that are able to move around in the scenario freely, exhibit gestures and communicate with each other. For the simulation of different cultures, culture-specific characters were modeled. Thus, we created prototypical German looking and prototypical Japanese looking characters (see Figure 3, left and middle) whose appearances (skin, hair or shape of the face) have been adapted to their cultural background.

Verbal behavior is realized by a text-to-speech component. For the different characters, different voices can be used, e.g. German, English or Japanese speech synthesis. Non-verbal behaviors are divided into gestures, postures and movement animations. Gestures can be culture-specific or not. An example of a culture-specific gesture is, for example, a bow for the Japanese greeting. Examples for culture-specific postures are shown in Figure 3 (left and middle). General gestures such as beat gestures can be exhibited by every agent. The performance of these gestures, however, can be customized and thus be performed in a culture-specific way. To this end, every gesture is divided into three phases: preparation, stroke and retraction. The preparation and retraction phases are used to blend the animations. A gesture could, for example, be chosen while the agent already performs another gesture or stands in a certain posture. The stroke phase can be performed in different ways taking into account the expressivity parameters. The parameter speed, for example, can be varied by playing the animation faster or slower; the parameter repetition can be changed by playing the stroke phase several times.

Figure 3: Culture-specific agents in the Virtual Beergarden (left: Germany; middle: Japan) and during the evaluation study (right: Japanese agents showing Japanese vs. German postures).

6. EVALUATION

Most misunderstandings in inter-cultural communication are caused by differences in non-verbal behavior [27]. In an evaluation study, we investigate whether the culture-related differences that we found in the literature and in our video corpus are perceived by human observers during agent interaction.

6.1 Design

In order to find out which of the behavioral aspects do have an impact on the user's perception, we simulated them in isolation. For the study conducted in Germany, the German looking characters were used and for the study conducted in Japan, we used the Japanese looking characters. In addition, we used language specific text-to-speech systems for the Western and Asian characters (German and Japanese) to match the prosody of the speech of the target culture. Thus, participants should not assume a cultural background different from their owns.

For each behavioral dimension, participants were shown two videos with face to face dialogs. In one video, the characters performed prototypical German behavior, in the other prototypical Japanese behavior for the specific behavioral aspect. In the study, participants had to state their preference by providing ratings on a 6 graded scale, containing three grades on each side, starting from "rather this video" to "by any means this video". For the two parted study, we stated the following two hypotheses:
H1: For each behavioral dimension, German participants prefer the videos showing German behavior over the Japanese versions.
H2: For each behavioral dimension, Japanese participants prefer the videos showing Japanese behavior over the German versions.

In order to avoid side effects evoked by gender, we showed mixed gender combinations in the videos. That is, one female and one male character interacted with each other in both cultures. To avoid preference for one of the videos due to the semantics of speech, we used Gibberish, a fantasy language that represents a language without any specific meaning of the words. To this end, words were generated that have the same statistical distribution of syllables as the words from the target language. The same dialog was retained during the whole study changing only aspects of the non-verbal and communication management behaviors. Keeping the dialog consistent also assured that the users' perceptions are not influenced by other linguistic features, such as the length of the sentences.

In order to get participants acquainted with the situation of listening to a Gibberish dialog, we showed a neutral conversation first. In this video, the dialog described above was performed without any non-verbal behavior or any pauses in speech or overlapping speech. After this neutral video, six pairs of videos were shown in random order, each lasting for half a minute and containing differences in one of the following aspects of behavior (see Figure 3 (right) for a sreenshot of the evaluation study as it was conducted in Japan):

- Pauses in speech: As we observed more pauses in the Japanese corpus, the simulated dialogs reflecting typical Japanese conversations contain more pauses as well. Taking into account our corpus findings, German agent dialogs contained one pause that lasted one second, whereas the Japanese version contained two pauses that lasted one second and one pause that lasted two seconds.

- Overlapping speech: Following our analysis of overlapping speech, we integrated one overlap that lasted 0.3 seconds and two overlaps that lasted 0.5 seconds into the German dialog. The Japanese dialog contained three overlaps that lasted 0.3 second, one that lasted 0.5 seconds and one that lasted one second.

- Communication management: Videos showing communication management behavior contained both: pauses and overlaps as described above.

- Speed of gestures: Our findings showed that in the German corpus gestures are performed faster than in the Japanese one. Thus, in one pair of the videos the gestures were customized according to speed. Three gestures were shown in both videos, but played faster in the German and slower in the Japanese behavior model.

- Spatial extent of gestures: Similar to gesture speed, another screen in the study contained two videos showing gestures with a different spatial extent. According to our findings, gestures had a smaller spatial extent in the Japanese models.

- Postures: The posture evaluation does not take the results on mirroring into account yet, but looks only into the interpretation of dominant body postures found in our corpus study for the two cultures.

6.2 Results and Discussion

As we stated earlier, we designed two different versions of our evaluation study. One utilizing the German-looking characters and a German text-to-speech system and another one using the Japanese-looking characters and a Japanese text-to-speech system, each showing both behavioral models. Instruction texts as well as preference questions matched the participants' mother tongue. In the German evaluation study, 15 participants took part (6 female and 9 male), while in the Japanese study 17 people participated (3 female and 14 male). All subjects were students (with one exception in the German study) in an age range between 20 and 45. In the evaluation study, participants had to decide which of the videos they liked better, assuming that participants prefer videos showing virtual characters that behave in a way that was designed for their own cultural background. In a goodness-of-fit test, we tested whether the observed pattern of events significantly differed from what we might have expected by chance alone.

Significantly more than 50% of our German participants had a preference for the version with German overlapping speech and spatial extent in gestures (both with $chi^2 = 8{,}067$ and p = 0.005 with df = 1). For pauses in speech, communication management and posture, we almost achieved significance (with $chi^2 = 3.26$ and p = 0.071 with df = 1 for all three aspects). However, by trend German participants showed a preference for the videos simulating prototypical German behavior for all aspects of behavior.

Results in the Japanese study are less strong. Significantly more than 50% of our Japanese participants had a preference for the version with Japanese posture behavior (with $chi^2 = 4.675$ and p = 0.029 with df = 1). For other behavioral patterns, we cannot claim any evidence. The results for pauses in speech and overlapping speech, however, were a bit surprising for the Japanese study as participants seemed to favor the German videos over the Japanese ones (although not significant). We attribute the missing semantics of the Gibberish dialogs as the main reason for this result, based on the following observations: On the German side pauses are generally viewed as somewhat awkward and overlaps as rude regardless of the semantic content of utterance. On the other hand, as discussions with our Japanese project partners showed afterwards, the use of pauses and overlaps in the Japanese language seems to be tight to the semantics of the utterances and is acceptable in one case and unacceptable in another. Thus, without having the necessary semantic clues at hand, Japanese participants might have been tempted to go for the "safe" solution and vote for the version with less pauses and overlaps.

This "failure" highlights a very important aspect of cross-cultural interaction in research teams. Despite frequent discussions and experience in cross-cultural projects, the developer's own cultural expectations are always present and sometimes interfere with the development. In this case, the seemingly good solution of using Gibberish for the tests, due to the arguments given above, lead us to missing an important feature of Japanese dialogs, i.e. its high context nature as Hall puts it [10].

Interestingly, the results for communication management behavior seem to be more related to the results from pause behavior than the results from overlapping behavior. We made similar observations in [7], where we considered communication management behaviors for the two cultures of Arabia and US America. The analyses suggested too, that the impact of pause behavior was stronger than the impact of overlapping behavior to human observers.

Although, we only had a limited number of participants in our study, for some cases we have significant results suggesting that behavioral patterns are preferred that were designed for the participants cultural background. However, for none or the behavioral patterns, we found evidence that more than 50% of our participants prefered behavior that did not match their cultural background.

7. CONCLUSION

In this paper, we investigated different behavioral dimensions for the two cultures of Germany and Japan, in order to find out which of these aspects affect the human observer. Focusing on parts of communication that are performed rather subconscious and where the influence of culture can play a crucial role without even realizing it, we concentrated on aspects of non-verbal behavior and communication management and did not consider semantics of speech yet. Culture-related differences have been extracted from the literature for the two cultures of Germany and Japan and strengthened by a empirical corpus study in the two target cultures. Results have been integrated into a multiagent system that demonstrates the simulation of cultural patterns of behavior.

For our evaluation study, behavioral aspects were tested in isolation. In that manner, we wanted to find out which of these aspects affect the perception of the user. Our preliminary evaluation study in Germany revealed that subjects significantly preferred the version that resembled behavior observed for their own cultural background for some of the behavioral aspects (overlapping speech and spatial extent of gestures). For all other aspects participants seemed to prefer the German versions at least by trend. In the Japanese evaluation study, we found out that Japanese subjects significantly preferred postures designed for their cultural background. Only for pauses in speech and overlapping speech we observed a controversial trend. One reason for this outcome might be the missing semantics of the shown dialogs. Since the Japanese version contained both more pauses and more overlaps in speech, but lacked the context in which they occur, participants chose the safe solution, i.e. the version with less pauses and overlaps. As a consequence, we think that pauses and overlaps need to be placed very carefully and in relation to the actual dialog.

Reflecting on our findings, we plan to refine our models in communication management by adding context. In another step, we want to combine all the aspects of behavior that we investigated in isolation and build a scenario with virtual characters that behave according to their cultural background on different channels. In that way, we want to believably simulate different cultural backgrounds and create an awareness for these differences on the user's side.

8. ACKNOWLEDGMENTS

The first author of this paper was supported by a grant

from the Elitenetzwerk Bayern (Elite Network Bavaria). This work was also partly funded by the European Commission within the 7th Framework Program under grant agreement eCute (education in cultural understanding, technologically enhanced) and the Japan Society for the Promotion of Science (JSPS) under a Grant-in-Aid for Scientific Research (C) (19500104) and (S) (19100001).

9. REFERENCES

[1] R. Aylett, A. Paiva, N. Vannini, S. Enz, E. André, and L. Hall. But that was in another country: agents and intercultural empathy. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Budapest, Hungary, 2009.

[2] P. Bull. *Posture and Gesture*. Pergamon Press, Oxford, 1987.

[3] J. Cassell, Y. Nakano, T. Bickmore, C.Sidner, and C. Rich. Non-verbal Cues for Discourse Structure. In *The 39th Annual Meeting of the Association for Computational Linguistic (ACL 01)*, pages 106–115, 2001.

[4] I. Damian, P. Huber, B. Endrass, and N. Bee. Advanced Agent Animation. In *IVA Gala 2010*, 2010.

[5] D. Efron. *Gesture, Race and Culture*. Mouton and Co, 1972.

[6] P. Ekman. *Telling Lies - Clues to Deceit in the Marketplace, Politics, and Marriage*, volume 3rd edn. Norton and Co., New York, 1992.

[7] B. Endrass, L. Huang, E. André, and J. Gratch. A data-driven approach to model Culture-specific Communication Management Styles for Virtual Agents. In *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010.

[8] B. Endrass, M. Rehm, and E. André. Culture-specific communication management for virtual agents. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Budapest, Hungary, 2009.

[9] C. Goodwin. *Conversational Organization — Interaction between Speakers and Hearers*. Academic Press, New York, 1981.

[10] E. T. Hall. *The Silent Language*. Doubleday, 1959.

[11] G. Hofstede. http://www.geert-hofstede.com/.

[12] G. Hofstede. *Culture's Consequences - Comparing Values, Behaviours, Institutions, and Organizations Across Nations*. Sage Publications, 2001.

[13] G. J. Hofstede, P. B. Pedersen, and G. Hofstede. *Exploring Culture - Exercises, Stories and Synthetic Cultures*. Intercultural Press, Yarmouth, United States, 2002.

[14] F. Iacobelli and J. Cassell. Ethnic Identity and Engagement in Embodied Conversational Agents. In C. Pelachaud, J.-C. Martin, E. André, G. Chollet, K. Karpouzis, and D. Pelé, editors, *Proc. of Conf. on Intelligent Virtual Agents (IVA 2007)*, pages 57–63. Springer, 2007.

[15] D. Jan, D. Herrera, B. Martinovski, D. Novick, and D. Traum. A Computational Model of Culture-Specific Conversational Behavior. In C. Pelachaud, J.-C. Martin, E. André, G. Chollet, K. Karpouzis, and D. Pelé, editors, *Intelligent Virtual Agents (IVA 2007)*, pages 45–56. Springer, 2007.

[16] W.-J. Johnson, S. Marsella, and H. Vilhjálmsson. The DARWARS Tactical Language Training System. In *Interservice / Industry Training, Simulation, and Education Conference*, 2004.

[17] W.-L. Johnson and A. Valente. Tactical Language and Culture Training Systems: Using Artificial Intelligence to Teach Foreign Languages and Cultures. In *Innovative Applications of Artificial Intelligence (IAAI 2008)*, pages 1632–1639. Association for the Advancement of Artificial Intelligence (AAAI), 2008.

[18] T. Koda, M. Rehm, and E. André. Cross-cultural evaluations of avatar facial expressions designed by western designers. In H. Prendinger, J. Lester, and M. Ishizuka, editors, *Proc. of Conf. on Intelligent Virtual Agents (IVA 2008)*, pages 245–252. Springer, 2008.

[19] T. Koda, Z. Ruttkay, Y. Nakagawa, and K. Tabuchi. Cross-cultural study on facial regions as cues to recognize emotions of virtual agents. In T. Ishida, editor, *Culture and Computing*, pages 16–27. Springer, 2010.

[20] A.-A. Lipi, Y. Nakano, and M. Rehm. Culture and Social Relationship as Factors of Affecting Communicative Non-verbal Behaviors. *Japanese Society of Artificial Intelligence*, 25(6):712–722, 2010.

[21] S. Mascarenhas, J. Dias, N. Afonso, S. Enz, and A. Paiva. Using rituals to express cultural differences in synthetic characters. In Decker et al., editor, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, 2009.

[22] C. Pelachaud. Multimodal expressive embodied conversational agents. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 683–689, 2005.

[23] M. Rehm, E. André, Y. Nakano, T. Nishida, N. Bee, B. Endrass, H.-H. Huan, and M. Wissner. The CUBE-G approach - Coaching culture-specific nonverbal behavior by virtual agents. In I. Mayer and H. Mastik, editors, *ISAGA 2007: Organizing and Learning through Gaming and Simulation*, 2007.

[24] M. Rehm, Y. Nakano, E. André, T. Nishida, N. Bee, B. Endrass, M. Wissner, A.-A. Lipi, and H.-H. Huang. From observation to simulation: generating culture-specific behavior for interactive systems. *AI & Society*, 24(3):209–211, 2009.

[25] M. Rehm, Y. Nakano, E. André, and T. Nishida. Culture-specific first meeting encounters between virtual agents. In *Intelligent Virtual Agents 2008 (IVA 2008)*, pages 223–236, 2008.

[26] M. Rehm, Y. Nakano, H.-H. Huang, A.-A. Lipi, Y. Yamaoka, and F. Grueneberg. Creating a standardized corpus of multimodal interactions for enculturating conversational interfaces. In *IUI-Workshop on Enculturating Interfaces (ECI)*, 2008.

[27] S. Ting-Toomey. *Communicating across Cultures*. The Guilford Press, New York, United States, 1999.

[28] F. Trompenaars and C. Hampden-Turner. *Riding the waves of culture - Understanding Cultural Diversity in Business*. Nicholas Brealey Publishing, London, 1997.

Controlling Narrative Time in Interactive Storytelling

Julie Porteous, Jonathan Teutenberg, Fred Charles and Marc Cavazza
School of Computing,
Teesside University,
Middlesbrough TS1 3BA,
United Kingdom
{j.porteous,j.teutenberg,f.charles,m.o.cavazza}@tees.ac.uk

ABSTRACT

Narrative time has an important role to play in Interactive Storytelling (IS). The prevailing approach to controlling narrative time has been to use implicit models that allow only limited temporal reasoning about virtual agent behaviour. In contrast, this paper proposes the use of an explicit model of narrative time which provides a control mechanism that enhances narrative generation, orchestration of virtual agents and number of possibilities for the staging of agent actions. This approach can help address a number of problems experienced in IS systems both at the level of execution staging and at the level of narrative generation. Consequently it has a number of advantages: it is more flexible with respect to the staging of virtual agent actions; it reduces the possibility of timing problems in the co-ordination of virtual agents; and it enables more expressive representation of narrative worlds and narrative generative power. Overall it provides a uniform, consistent, principled and rigorous approach to the problem of time in agent-based storytelling. In the paper we demonstrate how this approach to controlling narrative time can be implemented within an IS system and illustrate this using our fully implemented IS system that features virtual agents inspired by Shakespeare's *The Merchant of Venice*. The paper presents results of an experimental evaluation with the system that demonstrates the use of this approach to co-ordinate the actions of virtual agents and to increase narrative generative power.

Categories and Subject Descriptors

H5.1 [**Multimedia Information Systems**]: Artificial, augmented and virtual realities

General Terms

Algorithms

Keywords

Interactive Storytelling, Agents in games and virtual environments, Narrative Modelling, Planning

Cite as: Controlling Narrative Time in Interactive Storytelling, J. Porteous, J. Teutenberg, F. Charles and M. Cavazza, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 449-456.

1. INTRODUCTION

Time plays a central role in many aspects of narration [22] both at the story and at the discourse level. Time determines pace, dramatic tension as well as the aesthetic of story visualisation and staging. Existing Interactive Storytelling (IS) systems have emphasised the causal aspects of agents' actions but have not incorporated time in their narrative generation mechanism in a principled fashion.

The prevailing approach in IS has been to use AI planning for narrative generation and empirical solutions for the synchronisation of agents' actions, often arrived at by a process of trial and error or using deliberately underspecified representations that assume uniform execution time of agent actions. These approaches can work well, as demonstrated by a number of successful IS prototype systems, including [2, 27, 19], but they miss an opportunity to use action duration as an element of story presentation at the discourse level.

An alternative approach to controlling narrative time is to extend the representation of narrative actions to reincorporate temporal aspects (such as duration, concurrency, overlap and so on) in the planning process that is used for narrative generation. This would ensure that generated narratives contained explicit information about the timing of agent actions which could be used during the staging of the narrative. While the IS research community has enthusiastically embraced AI planning due to its capability for propagating causality, to date, there has been no use of dedicated temporal planning architectures. Yet these architectures are potentially useful for IS since it is likely that there are narrative situations that require dedicated temporal architectures (ones which are similar to the *temporally expressive* problems documented in the AI planning literature [8]). Applying temporal planning to narrative generation would provide a sound and principled approach to further increase the generative power of IS systems and to expand the range of stories that can be generated.

The use of temporal planning within the process of narrative generation is an approach that neatly re-incorporates aspects of the problem that have tended to be solved by trial and error. Clear benefits of this approach include: (i) it will enable the generation of story and discourse from shared principles; (ii) it will simplify development and production; (iii) it will improve integration of action and motion at the technical level. In addition, we anticipate that the approach will yield the following advantages: (i) help improve system reliability, e.g. by overcoming problems associated with timing and co-ordination of virtual agent actions; (ii) provide a wider range of possibilities for staging and cinematographic

aspects of virtual agent actions; and (iii) increase the generative power of the system, i.e. the range of agent situations and narratives that can be generated.

Throughout the paper we illustrate our discussion with examples taken from an interactive narrative that we have developed which features virtual agents and situations inspired by Shakespeare's play *The Merchant of Venice* [24].

The paper is organised as follows. In the next section we consider related work. This is followed in section 3 with discussion of issues related to the explicit temporal representation of actions and narratives. Section 4 gives an overview of our approach to generating temporal narratives. The results of an evaluation using our implemented system are presented in section 5. Section 7 summarises our conclusions.

2. RELATED WORK

2.1 Interactive Storytelling

A number of prototype IS systems have been developed that use AI planning for narrative generation [2, 27, 19]. These systems ignore the staged execution time of agent actions during narrative generation. Instead, they have adopted a range of solutions to the handling of temporal aspects at the staging level. One such approach is the use of *executability conditions* [15] to specify conditions for successful execution of actions [4]. This approach has been used to co-ordinate the actions of virtual agents but its failure to reason about temporal aspects such as staged execution time can make it unreliable. It also requires time-consuming empirical solutions for the actual production of interactive narratives thereby limiting its scalability.

A form of executability condition is used in the execution management architecture ZOCALO [27] to ensure that actions are executed in legal world states. The system makes some allowance for the time taken for actions to execute (a state of *executing* is maintained) and action effects are not activated until actions have successfully completed. However there is no explicit reasoning about action duration during narrative generation and this could make the system unreliable. For example, this omission may only become apparent during staged execution when an agent arrives too late to co-ordinate with another agent.

The LOGTELL system [19] also features an overall manager of the IS system which is responsible for controlling the staging of a partially-ordered plot output by their IPG generator. The system makes use of temporal logic as a representation for the state of the system, which can be used in particular when authoring the narrative. However no mention is made of its use for resolving the problems of temporal dynamics faced by narrative generation.

HPTS [11] is a system that reasons about time to handle the synchronisation of behavioural agents. Reactive behaviours are described within a runtime environment to handle parallel state machine execution and synchronisation of agents. This approach orchestrates the synchronisation of low level action execution (sometimes referred to as the motion level), such as motion blending and interruption.

An alternative approach is the use of Petri Nets which has been explored to handle the unfolding of story plots and the co-ordination of virtual agent behaviour [3]. However the behaviour of such a system is reactive and only includes deliberation about localised temporal aspects of the problem. Also localised in its approach is the use of cascaded Finite State Machines in SCENEMAKER [14]. This represents an orchestrated approach to temporal and synchronisation issues but its static strategy is rather inflexible and temporal reasoning is at the "microscopic" level not the planning level.

2.2 Research in Automated Planning

On the other hand temporal planning is a very active research topic in the field of AI planning which has generated multiple approaches, targeted specifically at temporal problems. These include logic based planning [1], partial order planning (ZENO [20], VHPOP [28]), hierarchical planning (NONLIN [25], OPLAN [12]), extended state space progression search planning (SAPA [10], SGPLAN [6]) and hybrid planning systems combining features of different temporal planning architectures (TEMPO [8], CRIKEY [7]). Early systems such as ZENO could tackle complex temporal problems but they suffered from performance limitations. More recently systems such as SGPLAN, TEMPO and CRIKEY have overcome efficiency problems to the point where they now have potential for application to IS.

3. REPRESENTING NARRATIVE TIME

IS systems that use planning for narrative generation use a representation of the narrative world that includes information about virtual agent behaviours represented as pre- and post-condition *actions*. These actions detail the way that the agent action is expected to change the state of the narrative world when it is staged in a visual environment. Not only can these actions describe the capabilities of an agent, but they can also describe properties inherent in the process itself – in particular their staged execution time. This notion of execution time may be represented either *explicitly* or *implicitly*: an implicit representation enabling the narrative generator to reason about relative orderings of actions; an explicit representation extending this to enable reasoning about complex temporal interactions[1].

3.1 Narrative Action Representation

In an implicit representation no temporal information is included in the description of agent actions and the assumption is that the effects of actions are instantaneous (the classical STRIPS assumption [13]). In contrast, explicit reasoning about the duration of actions makes it possible to take into account the more sophisticated interplay between the occurrence of actions themselves, not just their consequences. It shows the continuous evolution of the story world over time as actions unfold rather than merely showing actions as their consequences. This explicit *durative* representation provides a means to represent conditions that can be used for agent synchronisation: before an agent is able to start an action (e.g. in order for an agent to start to listen to another agent, they must be within earshot); at the end of the action (e.g. in order for an agent to make a selection between a number of alternatives, they must have reached their decision); or must remain true over the duration of the action as an invariant (e.g. during the time an agent listens to an agent singing they must stay in earshot). The durative action representation also makes it possible to specify which narrative effects occur immediately, as a virtual agent starts to perform an action (e.g. when a virtual agent sings,

[1]We note the correspondence between implicit and explicit models [17] and qualitative and quantitative models [7].

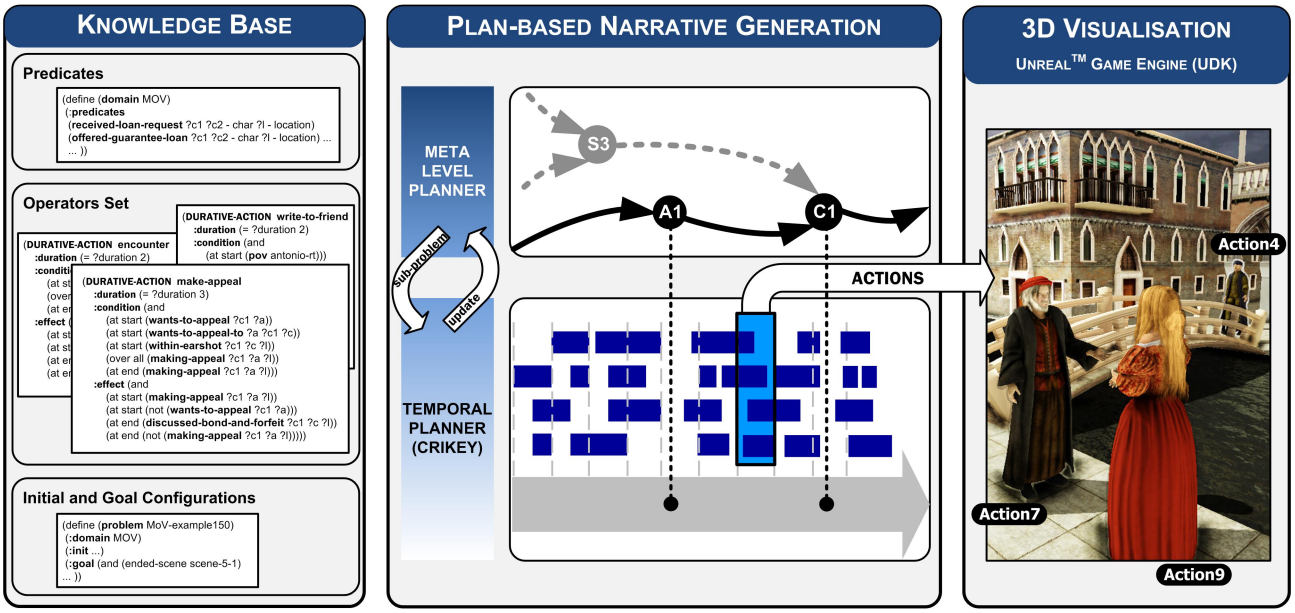


Figure 1: System Architecture: input is a domain model (knowledge base) represented temporally; the plan-based generator builds the narrative incrementally by decomposing the problem into a series of sub-problems which are then tackled in turn using a temporal planner; for 3D visualisation, the temporal narrative actions output by the planner map to UnrealScript action descriptions.

the sound starts immediately), and which are delayed until the agent finishes the action (e.g. an agent spends time persuading another agent, the effect of having been persuaded is activated at the end).

An illustration of the need for temporal reasoning is provided by act III scene ii of our *Merchant of Venice* system. In the scene there are specific narrative actions that require an informed decision by an agent. These actions must unfold whilst the agent acquires additional information through other actions (e.g. conversations). One such action is the selection of a casket by a character, Bassanio, in an attempt to win the hand in marriage of another character, the wealthy heiress Portia. A durative representation of the action is[2]:

```
(:durative-action select-casket
    :parameters (?c - char ?ca - casket ?l - location)
    :duration (= ?duration 4)
    :condition (and ....
              (over all (selecting ?c ?l))
              (at end (selecting ?c ?l))
              (at end (decided-to-select ?c ?ca)))
    :effect (and
              (at start (selecting ?c ?l))
              (at end (selected ?c ?ca ?l))
              (at end (not (selecting ?c ?l)))))
```

This illustrates the temporal properties of the action where deliberation lasts for the duration of the action (over all the character is *selecting*) but this must be finalised for the action to end when post-conditions are activated.

A non-durative version of this narrative action is cumbersome and does not capture the unfolding of agent deliberation over time. This may prevent the action from being synchronised with other agent actions or being interrupted (either by other agents or users in an interactive setting). In addition, deliberation has dramatic value in terms of staging and understandability: it enables the spectator to see agents' decision processes and the factors that influence them.

[2]We chose PDDL3.0 [16] because of its expressive power and since it is a standard action description compatible with multiple planning approaches.

3.2 Narrative Representation

Temporal narrative plans include information about the time each agent action is scheduled to start and the expected duration of each action. The following example:

```
0.001: (select-casket bassanio lead casket-room) [4.00]
0.002: (give-hint-in-song portia casket-room) [3.00]
0.003: (listen-to-song bassanio casket-room lead) [3.00]
```

is a representative example of the paradigm, showing the start time on the left of the action name and the duration on the right. This example occurs in act III scene ii of the *Merchant of Venice* where one of the characters, Bassanio, is deliberating about the selection of a casket whilst simultaneously acquiring information from hints that are given to him in song by another character. The temporal aspect of the action, namely the character's decision process (deliberation) can now be staged as an important element of discourse, as it incorporates important information on the relation between characters. Also, it allows for interference by other agents (or the user) thereby supporting further narrative generation

In contrast, capturing this in a non-temporal narrative is problematic since there is no way to specify start times and duration of actions. Actions can be left partially ordered (either generated by a partial order planner [28] or by lifting a partially ordered narrative from a totally ordered one [26]) but the required overlap between actions cannot be captured without explicit representation of time.

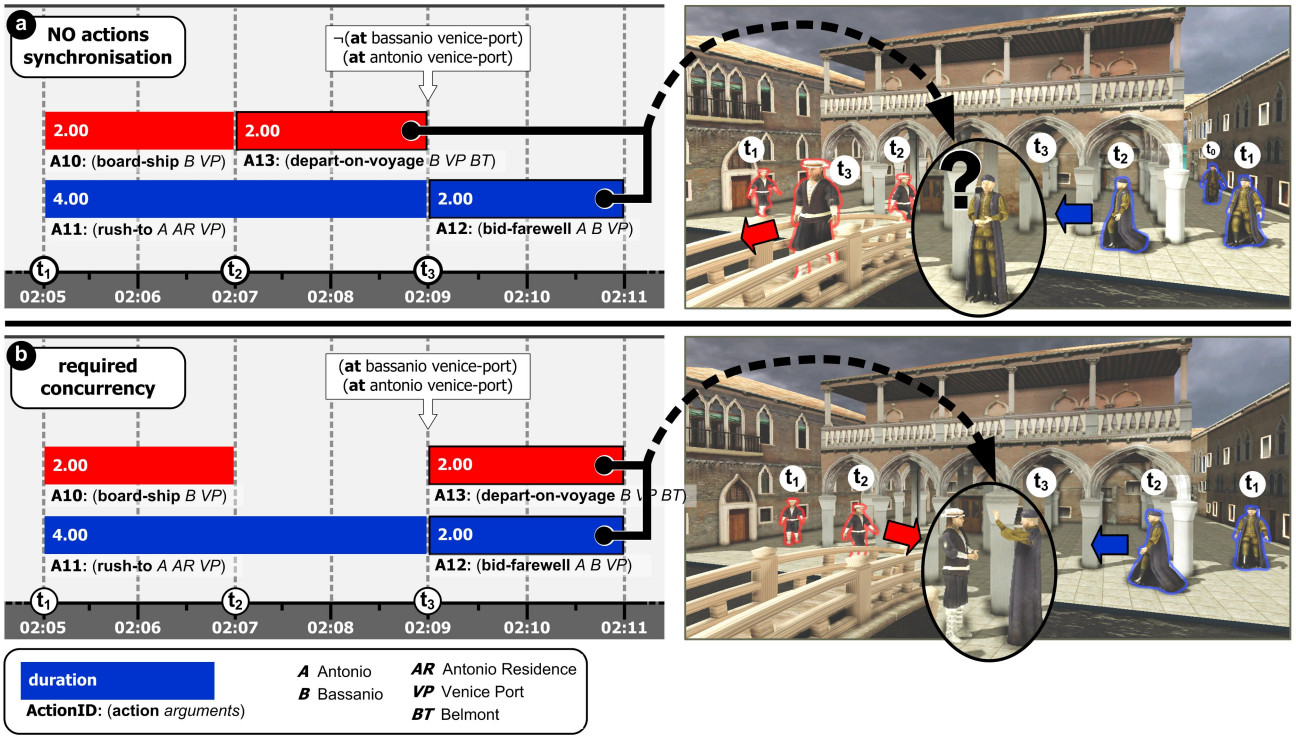


Figure 2: *Merchant of Venice* example illustrating the role of reasoning about staged execution time: (a) staging failure when narrative actions are not synchronised; (b) successful staging when reasoning about staged execution time during narrative planning identifies required concurrencies between actions.

4. NARRATIVE GENERATION

Generation of narratives that feature concurrent durative agent actions requires a planning architecture that can reason about explicit temporal information. Research in AI planning has led to the development of a number of dedicated temporal planning architectures (discussed in section 2). Recent, hybrid temporal planners such as TEMPO and CRIKEY have managed to overcome the performance limitations of earlier partial order planners and the incompleteness experienced by the extended space progression planners. Since our motivation includes being able to generate narratives that feature overlapping concurrent agent actions, we have chosen to use the CRIKEY system of Coles et al[7] in our implemented narrative generator. The system will use CRIKEY in combination with narrative structuring information since, without such information, the planner could end up generating sparse narratives or even no narrative at all [23]. The generator will use the information to guide CRIKEY towards the generation of narratives that are sufficiently rich and in keeping with the narrative genre.

The narrative structuring information represents key narrative situations that can be used like intermediate goals to guide the planner. After [21], we refer to these situations as *constraints* but they have also been described in the literature as *author goals* [23] and are similar to the notion of landmarks [18] that have featured in AI planning. The constraints for a narrative world are represented as a partially ordered set of predicates – a declarative representation which separates this information from action descriptions and may help facilitate its specification and maintenance.

Our implementation is based on the decomposition approach of [21]. This can be summarised as follows: use an input set of constraints to decompose the process of narrative generation into a sequence of sub-problems; generate a narrative for each decomposed sub-problem; and then assemble the final narrative by composition of the sequence of narratives. This approach implements a higher level of representation, where the constraints enable reasoning about narrative at the meta-level. The constraints can also be re-combined for different total orderings (as used in our experiments, see section 6).

Our contribution has been to extend their approach to handle temporal reasoning. These extensions were possible because of fundamental properties of the system that enabled the control program to be integrated with different base planners. An overview of the architecture of our implemented experimental system is shown in figure 1. The input is represented using the representation language PDDL3.0 which permits both implicit and explicit representations of the narrative domain to be input to the system. The control mechanism uses the input constraints to decompose the problem and then sends decomposed sub-problems to CRIKEY. As narrative actions are received from CRIKEY by the control mechanism they are sent to a visualisation module. The switch to temporal planning provides a direct route to mapping between planning actions and their visualisation through the transfer of PDDL3.0 temporal parameters to animation control structures (UnrealScript action descriptions).

5. QUALITATIVE EVALUATION

The objective of our evaluation was to provide data for the systematic assessment of system performance and behaviour. Here we evaluate the approach qualitatively, with reference to sample *Merchant of Venice* narratives generated by our system and shown in figures 2 and 3. These narratives provide answers to some key questions about our approach to controlling narrative time, namely: (1) can our approach help to avoid timing problems as agent actions are staged? (2) does our approach provide a mechanism to exploit information about the staging of agent actions? (3) does our approach to explicit temporal representation and reasoning increase the generative power of the system?

5.1 Avoiding Timing Problems

Failure to reason explicitly about temporal aspects of the IS domain at the point of narrative generation can cause problems that only become apparent when the virtual agent actions are staged. This can manifest itself both in real-time failure of the system and failure at the "production" level which it may be possible to repair through ad hoc local solutions. For example, if action duration isn't reasoned about during narrative generation then an agent may fail to meet up with another agent because they arrive too late, after the other agent has already left.

A scene from our *Merchant of Venice* system, shown in figure 2, illustrates how this situation can arise. In this scene one character, Antonio, is endeavouring to reach another character, Bassanio, in time to bid him farewell before he departs to sea. In principle, it is possible to generate a narrative for this scenario without reasoning about the staged execution time of the actions and then to use executability conditions (as used in [4]) to try to synchronise agents by testing that conditions for successful execution of agent actions hold. In this example the actions for Antonio are to rush to the port and then bid farewell to Bassanio as he leaves; the actions for Bassanio are to board the boat and then depart on his voyage. The first action for Bassanio has him boarding the ship and since this is independent of the first action for Antonio, rushing to the port, they can be staged and visualised in a concurrent fashion (which also gives interesting opportunities for exploration of automated camera control). The executability conditions for Bassanio's next action, departing aboard ship, do not mention anything about Antonio's location. Hence the action can start being visualised irrespective of the actual on-stage localisation of Antonio. Depending on how long Antonio takes to arrive at the port, it can happen that this doesn't occur until Bassanio has completely departed from the port, making it impossible for Antonio's final action, that of bidding his friend farewell, to be executed in the visual environment. This situation is depicted in figure 2.

How would explicit reasoning about time at the point of narrative generation mean such situations were avoided? The critical consequence of reasoning about the staged execution time of these agent actions is the recognition of the requirement that Bassanio must still be at the port when Antonio bids farewell to him, in other words that these actions are staged at the same time. This is shown in figure 2: the narrative generator has considered the duration of the actions, identified the required concurrency between them and forced them to overlap.

5.2 Providing Information for Staging

Our use of an explicit model of time results in generated narratives that include scheduled start times for each agent action and their duration, precisely the information that can be utilised for staging actions in different ways.

Act I scene (iii) of the *Merchant of Venice* provides an illustration of the generation of this staging information. The narrative for this scene (figure 3) shows the scheduled actions for the characters named Antonio, Bassanio and Shylock. The start of the narrative includes actions which bring them together on the Rialto ready to discuss the loan of a sum of money and subsequently seal a bond committing them to this arrangement. The red line drawn through the narrative in figure 3 shows the point at which this scene begins in the original play – opening with Bassanio and Shylock in conversation on the Rialto and continuing with the arrival of Antonio who joins them in conversation. This use of scene changes in classical theatre can be seen as a "tweak" which enables characters to appear at different locations as and when needed with no need to reason about their actions during the elapsed time (this tweaking of time has also been used in IS systems to avoid reasoning about agent actions whilst they are "off-screen" [21]).

However, in IS the objective is to provide different possible directions for the narrative and if there is a possibility that agent actions may need to be staged then they must be reasoned about. In our *Merchant of Venice* example, this means that earlier portions of the narrative (i.e. those before the start of the original scene from the play) need to be reasoned about during narrative generation. Consequently, the narrative in figure 3 also includes agent actions for the time before they enter into conversation. This allows for multiple ways of staging these actions, for example, focussing on one agent and their actions and motivations prior to the conversation, rather than cutting directly to them.

5.3 Generative Power

There are narratives that can only be generated with an explicit temporal approach. The scene depicted in figure 3 where the character Bassanio is enquiring about a loan and Shylock is simultaneously listening can be used to illustrate this. The action of Bassanio enquiring about the loan requires that Shylock listens to Bassanio for the whole of the enquiry. The action can be represented as:

```
(:durative-action listen-to-enquiry
  :parameters (?c1 ?c2 - char ?l - location)
  :duration (= ?duration 2)
  :condition (and
     (at start (at ?c1 ?l))  ...
     (over all (listening-to-enquiry ?c1 ?c2 ?l))
     (at end (listening-to-enquiry ?c1 ?c2 ?l)))
  :effect (and
     (at start (listening-to-enquiry ?c1 ?c2 ?l)) ...
     (at end (not (listening-to-enquiry ?c1 ?c2 ?l)))))
```

which captures the ongoing nature of the listening process with the condition *(listening-to-enquiry ?c1 ?c2 ?l)* that is activated at the start of the action and is maintained over the duration. However, in a non-durative version of this action, time would be compressed[3] and this condition would not be made true. This is problematic since the action of

[3]A compressed version of a durative action can be formed by setting the effects of the action to be the result of applying the start effects followed by the end effects and then setting

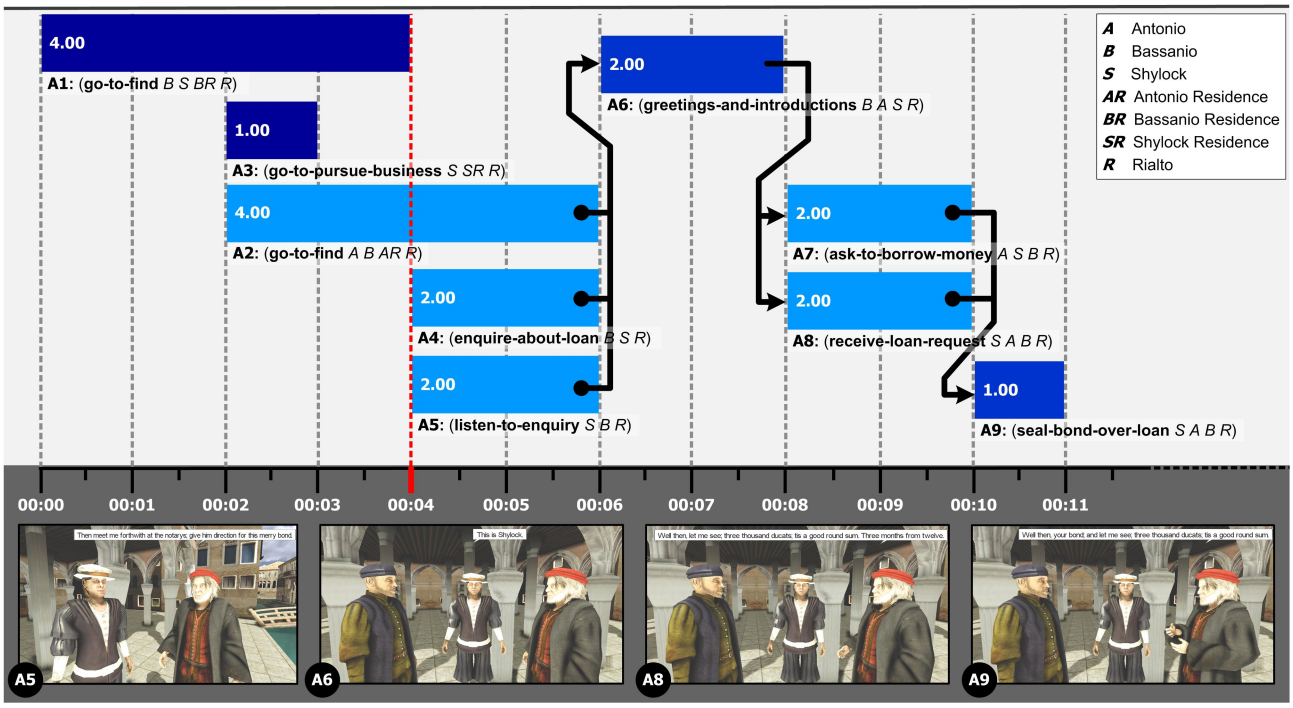


Figure 3: Example *Merchant of Venice* Narrative with overlapping durative actions: multiple possibilities for staging are introduced by temporal reasoning before the start of the scene in the original play (red line).

Bassanio enquiring about a loan requires Shylock to be listening (even in a compressed version this would remain as a pre-condition). The only way to handle scenarios such as this would be somewhat clumsy and would involve coercing the conversational exchange to take place at a given stage.

This example demonstrates the increased generative power of a temporal approach: narratives can be generated that require interactions over the duration of actions and these cannot be generated by compressed versions of the same actions. This is discussed further in the next section.

6. RESULTS

As demonstrated in the previous section, there are narratives which can only be properly generated and staged using narrative actions which have duration. Here we assess how this could affect real-world IS narrative generation problems, by examining the capacity these representations have for generating narratives for the different sub-problems that result from applying our decomposition approach in our experimental *Merchant of Venice* domain. These experiments focussed on narrative generation and consequently were performed off-line, without visualisation. The inclusion of staging would not significantly alter these results, and if anything, temporal planning would be less adversely affected given that the resolution of temporal factors is handled prior to visualisation.

In the course of one run of the IS system, user interaction could force the story to enter a broad range of unforeseeable world states. To simulate this, we generated a set of 20 potential initial states of the narrative domain by sampling randomly from the set of facts that are relevant to the different story sub-problems (where a fact is relevant if it can appear in a causal chain for achieving the sub-problem). A typical example of one of the randomly generated initial states contains the following facts:

```
(at bassanio venice-rialto)
(at antonio venice-street )
(decided bassanio lead-casket)
(enquired-about-loan bassanio shylock antonio)
```

In addition to facts specifying virtual agents' initial locations, in this state Bassanio has decided to choose the correct casket prior to travelling to Belmont, and has already discussed potential loans with Shylock. It should be noted that spurious facts, such as *(decided antonio gold-casket)* are never included in the generated initial states, as they are not deemed to be relevant facts (i.e. in this case, Antonio is not a suitor, and therefore has no reason for selecting caskets).

For each of the initial states, two narrative plans were generated, built up from 10 decomposed sub-problems. The first of these narratives was constructed using non-durative agent actions, and the second with durative ones. A cumulative count of the number of sub-problems successfully achieved was kept for each run. If one approach failed to achieve a sub-problem its state was changed to that reached by the other approach, and the system was then permitted to continue narrative generation from that point. This strategy was adopted in order to avoid unfairly penalising an approach for failing to achieve a sub-problem especially early in the narrative. Figure 4 shows the mean rate of sub-problem achievement for durative and non-durative actions. The solid lines indicate the mean number of sub-

the action pre-conditions to be the start conditions of the durative action along with all end conditions and invariants that are not achieved by the start effects [7].

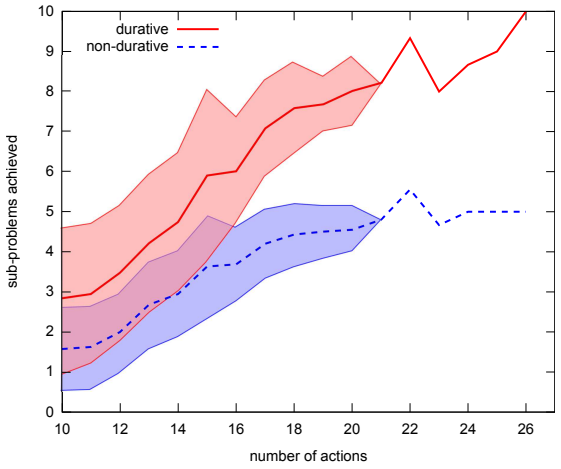


Figure 4: Count of the number of successfully generated narratives for decomposed sub-problems (with and without durative actions). Corridors show one standard deviation. (See text for further details.)

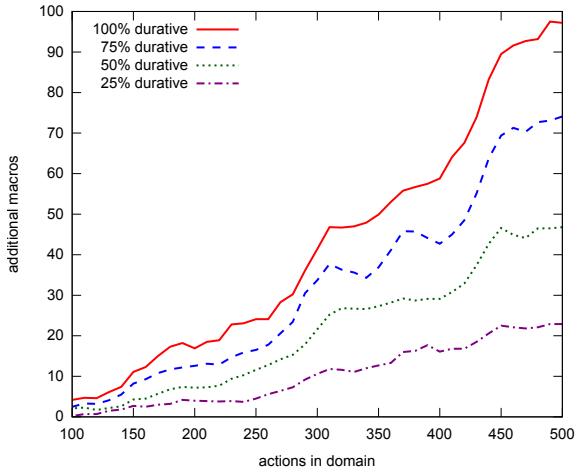


Figure 5: Increase in generative power resulting from the use of durative action representation. Lines show the increase in potential macros depending on domain size and percentage of durative actions.

problems achieved at each point in a narrative, and the corridor around each shows one standard deviation.

It is immediately clear that in this real-world example of an IS problem, generativity issues can have a significant effect on its execution. The graph shows results on output narratives of more than 10 actions, since narratives shorter than this are deemed too brief to be meaningful. For narratives of increasing length there is a clear difference in the number of sub-problems that can be achieved with the use of a temporal approach. Each failed sub-problem represents a point at which a real-world IS system must either sacrifice logical consistency of the narrative, or apply hand-crafted repair rules that jeopardise its scalability and reliability.

In addition to quantifying the expected rate of failure to achieve constraints after arbitrary user interaction, we also want to quantify the increase in generative power that temporal representations provide. As a measure of generative power, we consider the potential for non-trivial interactions between narrative actions of a domain. The simplest examples of these interactions can be seen in producer-consumer relationships between agent actions, such as when a condition that is added by one action is then deleted by another; or when a fact that is deleted by an action is then replaced by another. In IS, these sorts of interaction appear, for example, in conversations between characters, or when movement between locations is performed. An illustration is provided by the agent action *(board-ship bassanio venice-port)* that covers the movement of Bassanio from the port and interacts with actions that move Bassanio to the port (the "producers" in the relationship). Similar interactions occur between conversational actions, which feature agents entering and exiting the conversation through different actions. Most importantly, actions that do *not* interact in this way provide no scope for the generation of novel interesting narrative situations (similar to the idioms described in [5]).

The identification of these macros is performed in a phase of static domain analysis [9]. For the macros considered here the macro action sequence must be valid (i.e. the preconditions for each action are not violated by prior actions) and the post-conditions of the macro as a whole must differ from the union of its parts.

As a measure of these interesting narrative situations or idioms we counted the number of additional macro actions (i.e. all sets of actions with non-trivial interactions) that result as a consequence of using an explicit temporal representation. We created a set of test domains to measure the presence of macro actions with varying numbers of durative actions. The domain objects and facts were the same as those in our *Merchant of Venice* domain. The number of actions in each domain was similar to that used in the previous evaluation – between 100 and 500. These actions were randomly generated from the domain facts, and had the same number of pre- and post- conditions as those found in the *Merchant of Venice* IS domain. Figure 5 shows the number of additional macro operators present when 25%, 50%, 75% or 100% of the agent actions in the domain were defined as durative actions, and the remainder were compressed, non-durative versions of them (as described in section 5.3).

The results show that the fundamental nature of the durative representation of actions gives rise to a significant increase in the number of possible interesting interactions. For a 500 action domain, almost 100 additional macro actions were seen to appear from the switch to a pure temporal representation – each of which is a new, potential situation or idiom. When moving to domains with larger sets of actions (e.g. planning for the entire *Merchant of Venice* rather than the sub-plot used to illustrate this paper), the number of additional macros relative to the number of actions can be seen to grow at a super-linear rate. As seen in figure 5, applying the durative representation to only a subset of a domain can still realise this increase in generative power.

7. CONCLUSIONS

In this paper we presented the case for the use of an explicit approach to controlling narrative time in IS. This approach involves extensions to the representation of agent actions to include their staged execution time. It also includes

a shift to planning architectures that can schedule agent actions with required concurrency. The approach is applicable to a wide variety of different genres: those where timing or pace play a role, those where staging needs to be explored and those where story and discourse may have complex relationships. Overall the approach provides a uniform, consistent, principled and rigorous approach to the problem of time in agent-based storytelling

Our evaluation clearly demonstrated the advantages of a temporal IS approach: at the level of staging, it has been shown to overcome problems of timing of agent actions and provides a mechanism to exploit information about the staging of agent actions; and at the level of narrative generation, it has been shown to increase the generative power of the system. In addition the principled nature of the approach will be advantageous in system production since it removes the time consuming search for empirical solutions.

8. ACKNOWLEDGMENTS

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).

9. REFERENCES

[1] J. Allen. Planning as Temporal Reasoning. In *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, 1991.

[2] R. Aylett, J. Dias, and A. Paiva. An affectively-driven planner for synthetic characters. In *Proc. of 16th Int. Conf. on Automated Planning and Scheduling*, 2006.

[3] D. Balas, C. Blom, A. Abonji, and J. Gemrot. Hierarchical Petri Nets for Story Plots Featuring Virtual Humans. In *Proc. of the 4th AI and Interactive Digital Entertainment Conference*, 2008.

[4] M. Cavazza, F. Charles, and S. Mead. Generation of Humorous Situations in Cartoons through Plan-based Formalisations. In *Proc. of the ACM CHI Workshop on Humour Modelling in the Interface*, 2003.

[5] F. Charles and M. Cavazza. Exploring the scalability of character-based storytelling. In *Proc. of 3rd Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, pages 872–879, NY, USA, 2004.

[6] Y. Chen, B. Wah, and C. Hsu. Temporal Planning using Subgoal Partitioning and Resolution in SGPlan. *Journal of Artificial Intelligence Research*, 26:323–369, 2006.

[7] A. I. Coles, M. Fox, K. Halsey, D. Long, and A. Smith. Managing concurrency in temporal planning using planner-scheduler interaction. *Artificial Intelligence*, pages 1–44, 2009.

[8] W. Cushing, S. Kambhampati, Mausam, and D. Weld. When is Temporal Planning *Really* Temporal. In *Proc. of the 20th Int. Joint Conf. on AI (IJCAI)*, 2007.

[9] C. Dawson and L. Siklossy. The Role of Preprocessing in Problem Solving Systems "An Ounce of Reflection is Worth a Pound of Backtracking". In *Proc. of the 5th Int. Joint Conference on AI (IJCAI)*, 1977.

[10] M. Do and S. Kambhampati. SAPA: A Multi-objective Metric Temporal Planner. *Journal of Artificial Intelligence Research*, 20:155–194, 2003.

[11] S. Donikian. HPTS: A Behaviour Modelling Language for Autonomous Agents. In *Proc. of the 5th Int. Conference on Autonomous Agents*, 2001.

[12] B. Drabble and A. Tate. The Use of Optimistic and Pessimistic Resource Profiles to Inform Search in an Activity Based Planner. In *Proc. of the 2nd. Conf. on AI Planning Systems (AIPS-94)*, 1994.

[13] R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[14] P. Gebhard, M. Kipp, M. Klesen, and T. Ritt. Authoring Scenes for Adaptive, Interactive Performances. In *Proc. of the 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003.

[15] C. W. Geib and B. Webber. A Consequence of Incorporating Intentions in Means-End Planning. In *Working Notes – AAAI Spring Symposium Series: Foundations of Automatic Planning: The Classical Approach and Beyond*, 1993.

[16] A. Gerevini and D. Long. BNF Description of PDDL3.0. Technical report, 2005. http://www.cs.yale.edu/homes/dvm/papers/pddl-bnf.pdf.

[17] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[18] J. Hoffmann, J. Porteous, and L. Sebastia. Ordered Landmarks in Planning. *Journal of Artificial Intelligence Research (JAIR)*, 22:215–278, 2004.

[19] B. Karlsson, A. Ciarlini, B. Feijó, and A. Furtado. Applying a Plan-Recognition/Plan-Generation Paradigm to Interactive Storytelling. In *ICAPS Workshop on AI Planning for Computer Games and Synthetic Characters*, 2006.

[20] J. Penberthy and D. Weld. Temporal Planning with Continuous Change. In *Proc. of the 12th Nat. Conf. on AI (AAAI-94)*, 1994.

[21] J. Porteous, M. Cavazza, and F. Charles. Narrative generation through characters' point of view. In *Proc. of 9th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2010)*, 2010.

[22] P. Ricoeur. *Time and Narrative, Volume 1*. University Of Chicago Press, 1990.

[23] M. Riedl. Incorporating Authorial Intent into Generative Narrative Systems. In *Proc. of AAAI Spring Symposium on Intelligent Narrative Technologies*, Palo Alto, California, 2009.

[24] W. Shakespeare. *The Merchant of Venice*. Penguin Popular Classics (New Edition), 2007.

[25] A. Tate. Generating Project Networks. In *Proc. of the Int. Joint Conf. on AI (IJCAI-77)*, 1977.

[26] M. Veloso, M. Pérez, and J. Carbonell. Nonlinear Planning with Parallel Resource Allocation. In *Proc. of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, 1990.

[27] T. Vernieri. *A Web Services Approach to Generating and Using Plans in Configurable Execution Environments (M.S. thesis)*. PhD thesis, North Carolina State University, 2006.

[28] H. Younes and R. Simmons. VHPOP: Versatile Heuristic Partial Order Planner. *Journal of Artificial Intelligence Research*, 20:405–430, 2003.

ESCAPES - Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison

Jason Tsai[1], Natalie Fridman[2], Emma Bowring[3], Matthew Brown[1], Shira Epstein[1], Gal Kaminka[2], Stacy Marsella[4], Andrew Ogden[1], Inbal Rika[2], Ankur Sheel[1], Matthew E. Taylor[5], Xuezhi Wang[1†], Avishay Zilka[2], and Milind Tambe[1]

[1]University of Southern California, Los Angeles, CA 90089
{jasontts, matthew.a.brown, spepstei, aogden, asheel, tambe} @usc.edu, [†]littlexxxx@163.com
[2]Bar Ilan University, Israel
{fridman, galk} @cs.biu.ac.il, {avish12, inbalrika} @gmail.com
[3]University of the Pacific, Stockton, CA 95211, ebowring@pacific.edu
[4]USC ICT, Playa Vista, CA 90094, marsella@ict.usc.edu
[5]Lafayette College, Easton, PA 18042, taylorm@lafayette.edu

ABSTRACT

In creating an evacuation simulation for training and planning, realistic agents that reproduce known phenomenon are required. Evacuation simulation in the airport domain requires additional features beyond most simulations, including the unique behaviors of first-time visitors who have incomplete knowledge of the area and families that do not necessarily adhere to often-assumed pedestrian behaviors. Evacuation simulations not customized for the airport domain do not incorporate the factors important to it, leading to inaccuracies when applied to it.

In this paper, we describe ESCAPES, a multiagent evacuation simulation tool that incorporates four key features: (i) different agent types; (ii) emotional interactions; (iii) informational interactions; (iv) behavioral interactions. Our simulator reproduces phenomena observed in existing studies on evacuation scenarios and the features we incorporate substantially impact escape time. We use ESCAPES to model the International Terminal at Los Angeles International Airport (LAX) and receive high praise from security officials.

Categories and Subject Descriptors

I.6.3 [**SIMULATION AND MODELING**]: Applications

General Terms

Security

Keywords

Innovative Applications, Evacuation, Crowd Simulation

1. INTRODUCTION

From large-scale citywide evacuations to small-scale evacuations of buildings, emergency evacuations are unfortunately a perpetual fixture in society. Fire drills and other 'mock evacuations' generally used today fail to accurately prepare us for evacuations in which life-threatening danger is immediate and, in fact, are very often ignored altogether [5]. Thus, designing security policies based on them do not accurately account for actual human behavior. Simulations can provide an additional method of evaluating security policies that gauge the impact of different environmental, emotional, and informational conditions. In any evacuation, the layout of the area, the population composition, level of urgency, and the behavior of authority figures all play a role in the safety and speed of an evacuation. The ESCAPES system is a multiagent evacuation simulation tailored to the needs of airport security officials based on existing psychological and evacuation research.

Office buildings and railway stations, which are often the subject of evacuation studies, possess largely homogenous crowds of business people that are very familiar with the environment. Airports, however, have a large presence of families and first-time visitors which are major considerations for security officials [3]. Families present a completely different model of human behavior, as they no longer follow the often-assumed 'self-preservation first' edict and often seek to ensure the safety of family members first [19]. Travelers' uncertainties about the environment logically lead to increased reliance on authority figures for directions and necessitates a realistic model of information-spread about events and exits as well as a model of behavior when no exit locations are known.

These features that officials have identified as especially important to airport evacuations have not been specifically addressed by existing commercial and academic simulators. Legion Software[1], for example, is used by security forces in many areas to evaluate the expected speed of traffic flow through an area. However, it does not model agent types such as families and authority figures or realistic knowledge spread about the environment and events. Other evacuation simulators in academia explore more detail and even base their agents on psychological models, such as Pelechano et al. [18]. However, their work does not model the behaviorial dynamics unique to family units, nor the emotional contagion of the crowd as fear levels rise during the evacuations.

In our meetings with security experts affiliated with Los Angeles International Airport, they discussed the importance of agent types, the presence of fear, and realistic knowledge spread. In addition, a

[1]www.legion.com

Cite as: ESCAPES - Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison, J. Tsai et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 457-464.

Phenomenon	Ref.	Feature
People forget their entrance	[2]	Misc.
First-time Visitors	[3]	SoK / SCT
Heightened emotions -> chaos	[21]	Emotions / EC
Herding behavior	[10]	SCT / Families
Pre-evacuation delay	[4, 14]	SoK / Families
Families gather before exiting	[19]	Families
Authorities calm people	[21]	Auth / Emotions

Table 1: Phenomena modeled in ESCAPES

strong 3D visualization was emphasized for the purpose of visual conditioning during security personnel training. Thus far, airport security officials have been forced to use general simulations to answer questions about authority figure placement, number, and policy. Our work aims to fill this gap by tailoring a system to the particular needs of an airport evacuation and other similar scenarios with a solid grounding in psychological and evacuation research.

We discuss our multiagent evacuation simulation system, ESCAPES, in two parts: individual agent types and agent interactions. ESCAPES includes regular travelers, authority/security figures, and families, as these have been documented as having the most impact in an airport evacuation [3]. Another major aspect of evacuations is fear. Although there is substantial debate on the existence of 'panic' in evacuations, the presence of fear is undisputed [20]. For the purposes of our work, we focus on a baseline implementation of fear and its impacts. Finally, in discussions with airport security officials, incomplete knowledge of the environment was cited as a major concern. Thus, we also give agents incomplete knowledge of the world by restricting their knowledge of the exits and the event causing the evacuation.

ESCAPES agent interactions include three separate phenomena: spread of knowledge, emotional contagion, and social comparison. Evacuation literature shows that the crucial seconds people spend before actively moving towards an exit greatly impact their survivability and is largely due to uncertainty about the nature of the evacuation [4]. Thus, we include a 'Spread of Knowledge' (SoK) component, which realistically models the spread of information about an event and that an evacuation is truly necessary. Emotional Contagion (EC) is the well-documented phenomena that causes one person's emotional state to be impacted by neighboring people's emotional state [9]. We incorporate EC in our system as a logical byproduct of our inclusion of fear in the presence of crowds. Finally, in a situation where people don't have all the information, following others is a commonly seen phenomenon. Social Comparison Theory (SCT) is a theory of how one person impacts another at a broad level, positing that people perceived to be similar to each other will mimic each other [6]. We use SCT to direct people's actions when they have no knowledge of the environment.

Existing evacuation simulations fail to take these factors into account in a cohesive fashion, resulting in visually appealing but ultimately inaccurate simulations of airport evacuations. In ESCAPES, we model agents based on key features identified by LAX officials and attributes from evacuation literature and explore the impacts of these factors on the speed and smoothness of evacuation. In particular, we include emotions that impact behavior, authorities, family units, realistic spreading of knowledge about an emergency, emotional contagion, and social comparison. We describe each of these components in more detail in Sections 3 and 4 and explore their impacts on evacuations in great depth in Section 5. We show that inclusion of these factors leads to a number of emergent behaviors documented in literature, as summarized in Table 1. Finally, we conduct tests on a model of a terminal at Los Angeles International Airport and begin to provide answers to security officials' questions about authority figure policies.

2. RELATED WORK

Early work in pedestrian dynamics noted the similarity between crowd behavior and well-understood phenomena observed in physics. These observations led to the development of models based on fluid-dynamics [12]. Another approach to force-based crowd simulation is built off the idea of social forces [11]. Instead of being based on the physical properties of water or gas, social forces represent the attractive and repulsive forces felt by a pedestrian toward various aspects of its environment. Yet another approach involves the use of cellular automata (CA). In CA-based models [1], the environment is divided into a grid consisting of cells. At each time step, a cell transitions to a new state based upon its current state and the states of the neighboring cells. However, in both forced-based and CA-based models, it is difficult to simulate goal-driven and heterogeneous behavior. Thus, the specific crowd phenomenon we are looking at are not typically modeled with these approaches.

Agent-based models allow for each pedestrian to be modeled as an autonomous entity. Under this model, pedestrians are represented as agents capable of perceiving and interacting with their environment as well as other agents. While being the most computationally expensive modeling technique, agent-based models are capable of a higher degree of expressivity and fidelity. The ability to represent cognitive information and model complex and heterogeneous behaviors has opened the possibility for new avenues of research that had not been attempted with previous methods.

As a result, there has been a shift toward the use of agent-based models for evacuation simulations. However, much of this research has been focus solely on modeling the physical interactions between agents[16]. The EXODUS[2] system represents the state-of-the-art for these systems with versions specifically for various types of large-scale scenarios and additional modules that can model phenomena such as toxic gas and fire spread. The system does move slightly beyond physical interactions to include informational aspects such as signage and exit familiarity, but still does not attempt to use psychologically-based decision-making in their agents.

Despite this trend, there has been some interest in incorporating emotional as well as the informational interactions into agent-based models. The complex relationship between the spread of information and the spread of emotion was explored from a theoretical modeling perspective in [13]. [17] focuses on creating agents with sophisticated psychological models. Our research is less concentrated on individual agents and more concerned with the interactions between agents and the resulting group dynamics. Additionally, ESCAPES is focused on a different set of domains including airports, malls, and museums. To accurately represent these types of environments, we believe it is particularly important to model the influence of families, emotional contagion, social comparison, and spread of knowledge, which past work has not cohesively addressed.

3. AGENT DESIGN

The ESCAPES system is a two-part system comprised of a 2D, OpenGL environment based in the open-source project OpenSteer[3] and a 3D visualization component using Massive Software[4]. The 2D module consists of agents as described below, outputting their

[2]http://fseg.gre.ac.uk/exodus

[3]http://opensteer.sourceforge.net

[4]http://www.massivesoftware.com

physical and behaviorial information into files that are then imported into customized Massive extensions to generate 3D movies of the scenarios. The 2D module can be used for efficient statistical analysis of different security policies. As mentioned previously, the 3D visualization is a key component for airport security officials, as it provides a superior training medium to their current tools. Screenshots in Figure 3 show the children models as well as some people running in different directions (denoted in the white circle) when an evacuation begins. Here we describe the architecture of the 2D module, first introducing the individual traveler agent, then detailing two special agent categories (families, authorities), and finally discussing interaction level dynamics (spread of knowledge, emotional contagion, social comparison).

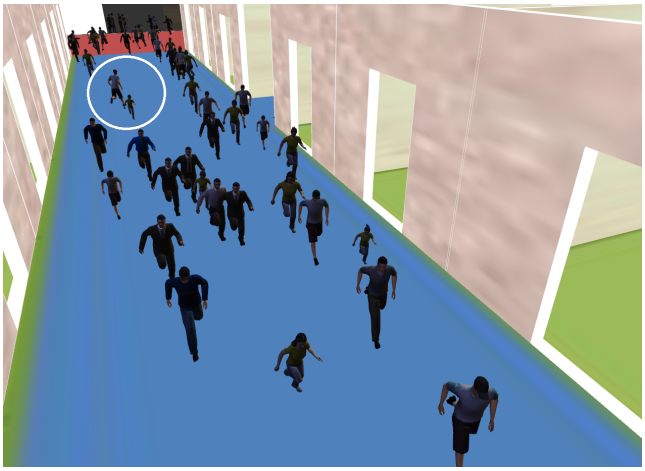


Figure 1: ESCAPES 3D visualization

3.1 Individual Travelers

All agents share a common architecture based in a BDI framework, possessing varying degrees of knowledge about the world and their neighbors. Each agent has access to a subset of the 14 available behaviors, any one of which may be active at a given time, where the behavior is selected via a probabilistic weighting scheme. The weighting scheme is a combination of 6 'Cognitive Mechanisms,' each of which prioritize some of the agent's desires. For example, there is a Cognitive Mechanism that prioritizes the basic desire of an agent to 'Wander' through his environment or 'Shop' in the stores. On the other hand, we have another Cognitive Mechanism that prioritizes an agent's desire to survive by evacuating through an exit once an event has occurred via one of the escape behaviors ('Run to Nearest Exit', 'Run to My Exit', and 'Search for Exit'). During execution of these behaviors, individual travelers may move at integer speeds from 0 to 3.

Each agent also has specific levels of emotions and information about the environment. Studies have shown that emotional stress causes changes in decision-making and may even cause someone to forget where he/she entered a building from [2]. Combined with the incomplete knowledge of a person that is in a place for the first time, which occurs extremely frequently in the airport scenario that we model, an evacuation suddenly becomes much more difficult to manage. Thus every agent has a fear level, an event certainty level, as well as a list of known exits. A more extended discussion of these attributes will take place in Section 4, but we briefly mention their implementation here first.

Fear is modeled as an integer value between 0 and 2 (*FearFactor*), 0 indicating that the agent has no fear. Higher levels of fear lead to higher movement speeds to get out of the area as soon as possible. Each agent's fear is a result of a number of factors such as their proximity to the event, the presence of authority figures nearby (as a result of documented impact of authority figures on evacuees [3, 21]) and the level of fear in neighbors and family members (as a result of Contagion [9]).

Event certainty is modeled as an integer value between 0 and 2 (*EventCertainty*), designating how aware the agent is that an event has occurred and that, therefore, an evacuation is necessary. An event certainty level of 2 is generated only by people close to the event, who immediately run directly away from the event before beginning active exiting behavior. Further away agents may have 1, which immediately triggers exiting behavior. Agents furthest away have an EventCertainty of 0 and continue their normal behavior, as they are unaware of any need to evacuate. Each agent's EventCertainty level is dictated by their proximity to the event, the presence of authority figures nearby that would inform them of the event, and the event certainty of neighbors via the Spread of Knowledge mechanism discussed in Section 4.1. The importance of uncertainty about an event has been noted in evacuation literature as a major cause of delay and, therefore, casualties [4].

Exit knowledge is modeled as a binary value indicating whether or not an agent knows about a given exit. Given a list of known exits, if an agent decides to evacuate, he will choose the nearest one. Exit knowledge is dictated by where they entered from, a random chance to forget that exit, and the presence of authority figures nearby that would inform them of exits. A person's knowledge of exits are clearly of paramount importance in any evacuation situation, especially in airport scenarios where many people are first-time visitors and are unaware of the environment layout.

3.2 Family Agents

Evacuations in some environments pose additional challenges as a result of the population present. In the airport scenario that we focus on, families have been identified as an important facet of the environment that must be modeled to more realistically portray the situation [3]. One can see how this might differ from the evacuation of an office building where only knowledgeable adults are present. For instance, children often rely on their parents to lead them and parents will undoubtedly seek out each other and their children before exiting, oftentimes disobeying authority instructions [19].

We model the presence of family units composed of 2 parents and 2 children with behaviors and cognitive mechanisms not applicable to general agents. Prior to an evacuation, children usually execute the 'Follow Parent' behavior, except occasionally executing the 'Drag into Shop' behavior which leads their parents into nearby stores that they find interesting. To enhance realism, we also restrict children to slower movement speeds (maximum of 2), which parents leading them will inevitably match. Parents that are not with their children heavily prioritize finding them via the 'Find Child' behavior, and put some emphasis on the 'Find Other Parent' behavior (they may also Wander or Shop). When an evacuation occurs, however, parents immediately seek each other out to gather the family together before proceeding to an exit, as has been shown to occur in real evacuations [19]. After an evacuation is underway, children will no longer execute the 'Drag into Shop' behavior, resorting exclusively to 'Follow Parent'.

3.3 Authority and Security Agents

Studies have shown that some authority figures have a very strong calming effect on people in an evacuation situation [21]. This can come through implicit calm at the sight of other people that appear calm via emotional contagion and may be enhanced due to the uni-

formed authorities having a stronger contagion effect due to their leadership role [9]. Also, by simply being there everyday, authorities know the environment and are trained to properly direct people to the nearest exits in the event of an emergency.

In our simulator, under normal conditions, authority agents 'Wander' or 'Patrol' the environment. After an event occurs that necessitates an evacuation, all authority figures switch to 'Patrol' in an attempt to inform everyone of the event and where nearby exits are located. We also set the FearFactor of authority figures very low and keep it constant to mimic well-trained security personnel that can maintain a level head in volatile situations. The calming effect they have on other agents is modeled by overriding nearby agents' FearFactor with the authority figure's FearFactor. The practical effect of this is to slow agents down (since FearFactor directly impacts travel speed), which may increase the total evacuation time, but also reduces the severity of colliding and the level of chaos. Also, authorities know all exit and event locations and pass this information to agents that are nearby.

4. AGENT INTERACTIONS

With the existence of crowds, agent interactions are a fundamental aspect of our evacuation simulation. Thus, we base our agent interactions on existing evacuation and social psychology research. We incorporate a realistic 'Spread of Knowledge' of events and exits, an Emotional Contagion module to model the infectious nature of emotions, as well as a social comparison component to capture people's mimicry of others.

4.1 Spread of Knowledge

As mentioned, while unimportant for office building or railway station simulations, realistic knowledge spread to model the behavior of first-time visitors is a necessary component in an airport simulation. Thus, we model the spread of two types of knowledge in our system: Exit Knowledge and Event Knowledge.

4.1.1 Exit Knowledge

People entering an environment for the first time will possess incomplete knowledge of exit locations. Thus, they must rely on authorities, signs, and following the crowd to make their way towards the nearest exit if there is one closer than the one they entered from. It has been shown that in times of high emotional stress, people even forget where they entered [2].

Our simulator includes this level of realism, giving agents knowledge of their entry location and a random chance that they forget this knowledge. In contrast, authority figures begin with and maintain full knowledge of all exit locations and pass a limited subset of this to nearby agents to simulate their redirection of passersby to the nearest exits. Also, family members will inform each other of exits they find out about, but otherwise, agents do not communicate exit knowledge to each other. Agents are also able to use the 'Search for Exit' behavior to find a way out on their own or some may choose to simply follow nearby, similar agents via the SCT module's 'Follow Most Similar Agent' behavior.

4.1.2 Event Knowledge

In real emergency situations, pre-evacuation delay has been cited as a major cause of slower evacuations and, therefore, deaths [4, 14]. This delay is largely due to a lack of knowledge about the emergency, both in disbelief of the severity of the situation as well as a desire to find out more about what has occurred. Pre-evacuation delay has been noted to persist despite verbal warnings and physical cues in the environment [14].

In our simulation, agents that are near the event as it occurs will have full knowledge of what has occurred, whereas agents far away have no idea are unaware that anything is wrong. As civilians pass each other, they communicate their level of certainty to each other, raising awareness of the situation. As civilians become more aware, they are more likely to run towards the exit as their self-preservation desires take precedent over all other desires.

Authority figures are assumed to instantly know when something has occurred, simulating an immediate radio notification from central security personnel. This does not necessarily translate into an immediate announcement to the general public, since oftentimes the appropriate response is not immediately obvious. Authority figures also communicate their certainty of the event to nearby agents, mimicking an actual authority figure telling people to evacuate.

4.2 Emotional Contagion

Emotional contagion is the effect of one person's emotional state on the emotional state of people around him/her both explicitly and implicitly [9]. It has been observed in families, small-scale interactions as well as large crowds [7, 9]. Researchers continue to develop theories on the phenomenon and are still exploring the various factors that are believed to influence the level of contagion as well as its effect on decision-making.

In an evacuation scenario, fear abounds, due both to uncertainty of the situation as well as concern for one's own safety [21]. As a result of emotional contagion, bystanders that are unaware of the event may develop otherwise inexplicably high levels of fear as well. Their subsequent decisions and behaviors as a result of this 'inherited' fear have not been explored in the context of a crowd or evacuation simulation. We therefore propose a baseline implementation and analysis of a model of emotional contagion.

Specifically, we have two components that spread emotions amongst agents. First, as agents pass by each other, they inherit the highest level of fear of neighboring agents. This is the baseline emotional contagion model that conforms with a theory of emotional contagion in which the highest level of emotion is transferred to all surrounding agents and inherited at full effect [9]. Second, as agents pass by authority figures, their level of fear is reduced to the authority figure's fear level. This simulates the implicit and explicit calming effect of authorities and conforms with a theory of emotional contagion that allows for specific agent types to reduce the level of emotion of surrounding agents (e.g., an agent that is greatly respected by all surrounding agents [9]).

4.3 Social Comparison (SCT)

Social Comparison Theory [6] is a social psychology theory, initially presented by Festinger. It states that humans, when facing uncertainty, compare themselves to others that are similar to them, and act towards reducing the differences found. Social comparison is considered a general cognitive process, which underlies human social behavior. During emergencies, individuals face greater uncertainty, and thus the weight of social comparison in human decision-making is increased [15].

We find the utilization of the computational model of social comparison [8] helpful in developing agents with the social skills that are crucial to the accurate simulation of different crowd behaviors. The SCT computational model can be used, for instance, by agents who wish to urgently exit an area. If they do not know the location of a close exit, they may turn to mimicking others hoping that they will lead them to safety.

For the simulation, SCT was implemented as follows. First, the agent compares itself to others around it by measuring the similarity in a set of features, including speed, emotional state, distance,

etc.. The similarity values are combined, and the agent that is most similar (within bounds) is selected. The agent executing SCT takes actions to reduce dissimilarities to the selected agent. In this simulation, SCT increases the tendency to mimic someone else's behavior, whereas emotional contagion transfers emotions regardless of what different behavior will be chosen based on it.

5. EVALUATION

We conducted extensive testing using a generic scenario to evaluate the impact of the emotional and informational phenomena modeled in ESCAPES. The scenario takes place in a generic airport setting consisting of 2 gates, 3 hallways, and 14 shops. There is an exit in each gate as well as the end of one of the hallways. Unless otherwise noted, the experiments for the generic scenario feature the following: 100 travelers which includes 10 families, 10 authority figures, emotional contagion, spread of knowledge, and social comparison. Simulated evacuations are typically evaluated by examining the rate at which people evacuate. While, evacuation rate is obviously important there are other metrics which can also provide insight as to how an evacuation proceeded. In Sections 5.1-6, we analyze the results from these experiments using the metrics which best highlight the effect of the various phenomena. Additionally, we modeled Tom Bradley International Terminal at Los Angeles International Airport and ran proof-of-concept tests on this to evaluate our performance on a real domain. A description of the scenario and accompanying results is provided in Section 5.7

In all of our experiments, an event occurs during the 14th time step and travelers have until the 300th time step to evacuate. It is assumed that by this time, airport officials will have managed to coordinate in response and issue a general order to evacuate through their emergency broadcast system. All the results in this section have been averaged over 30 independent simulations.

5.1 General Testing

As mentioned in previous sections, current evacuation simulators tend to focus on the physical interactions of agents. The agents in these simulations are typically homogeneous, rational, and omniscient. In contrast, ESCAPES agents are heterogeneous, emotional, and limited in both knowledge and perception. In Figure 2, we compare the evacuation rates from simulations in which the population of travelers is modeled as homogeneous, omniscient agents to those in which the population is modeled as ESCAPES agents including authority figures and families. The y-axis represents the percentage of travelers who have yet to evacuate. This percentage will decrease over time and the slope of the line signifies the current rate at which travelers reached safety. For example, after 85 time steps we can see all travelers have evacuated in the physical interaction model whereas 25% of travelers have yet to evacuate in the physical, emotional, and informational model.

When modeling omniscient agents, simulations consist of travelers with complete knowledge who are not influenced by their emotions. The only relevant interaction between travelers occurs when there is congestion due to an area becoming overcrowded. When the event occurs, all travelers are able to perceive it instantaneously and begin to head for an exit. We see a steep decline in the number of unevacuated travelers, as those close to an exit evacuate rapidly. There is then a temporary decrease in the rate of evacuation as those travelers who were far away from an exit rush towards it. Once those travelers start reaching the exits, the rate of evacuation picks up again until everyone has evacuated. While these models can provide a good first order approximation, they fail to capture much of the underlying complexity present in evacuations.

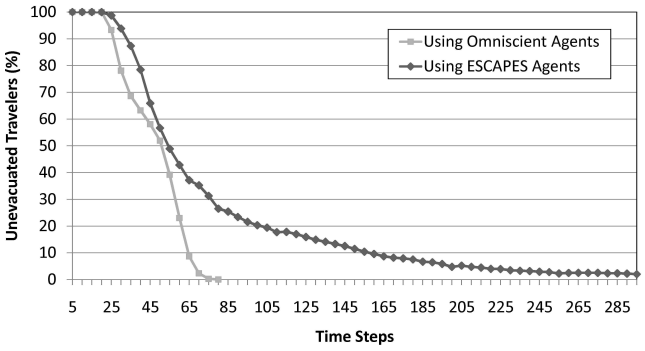


Figure 2: Effect of Modeling Physical, Emotional, and Informational Interactions on Evacuation Rate

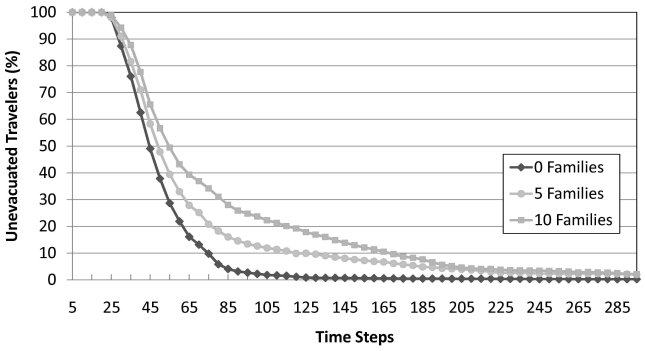


Figure 3: Effect of Families on Evacuation Rate

With travelers who are more realistic, the evacuation rate is slower. This is due to a multitude of factors such as families taking time to find their loved ones, travelers never learning about the event, or travelers having limited knowledge about exits. Unlike when travelers are modeled as omniscient agents, situations arise with ESCAPES agents where there are travelers who are unable to evacuate in time. However, it is important to examine these situations because it is exactly these scenarios where the potential for danger is greatest were they to occur in real life. Models using omniscient agents provide best-case scenarios and a lower bound on evacuation times. While this information is useful, a system that is capable of modeling unforeseen worst-case scenarios, such as ESCAPES, will be more effective as a training and policy-making tool.

5.2 Families

Studies have shown that the presence of the families results in slower evacuation times [19]. We tested the effect of families on evacuation rate by comparing the results from simulations with varying numbers of families. Figure 3 shows that increasing the number of families slows the overall rate of evacuation. After 85 time steps, simulations starting with 10 families had 30% of travelers remaining, whereas the simulations with 5 families had 15% remaining, and simulations with no families had only 5%. This slow down is a consequence of two main factors. First, instead of heading towards a known exit immediately upon learning of the event, parents first seek out the other members of their family. As a result, parents will often ignore known information and perform actions which are suboptimal from an individual perspective. Second, once family members have found each other, they stay grouped together. Due to children moving more slowly, as mentioned in Section 3.2, family units move slower than typical travelers.

5.3 Emotional Contagion

The spread of emotions through crowds as a result of emotional contagion has been well-documented [9]. In the simulations, emotional contagion is used to propagate fear. Travelers with high levels of fear pass on their FearFactor to travelers with lower levels of fear. Higher values of FearFactor activate a flight response in travelers. At the crowd level, this phenomenon causes travelers to collide into each other. The overall number of collisions can then be view be as a measure of the level of chaos in an evacuation. By modeling emotional contagion, we would expect to see an increased levels of fear which in turn will produce a higher number of collisions between travelers.

To isolate the impact of emotional contagion we ran experiments without authority figures. Without the calming influence of authority figures, there is nothing to impede the dissemination of fear through emotional contagion. Specifically, we compared the number of high-speed collisions that occurred over the course of an evacuation both with and without emotional contagion. High-speed collisions are defined as collisions that occur while a traveler has a speed of 2 or greater. Focus is placed on these collisions as they are more likely to cause injury or falls in real evacuations. When emotional contagion is modeled, evacuations average 6932 high-speed collisions, whereas evacuations without emotional contagion average 2701 high-speed collisions. From these results, we can see that modeling emotional contagion results in more chaotic evacuations with an increased number of high-speed collisions.

5.4 Spread of Knowledge

Agent-based evacuation simulations often start after an incident has occurred and assume that all agents are instantaneously aware of the need to evacuate. ESCAPES is geared towards domains where this is likely not the case. It is then important to model how knowledge of an event would spread throughout a crowd. In the simulations, EventCertainty represents the level of a traveler's knowledge of the event. Higher values of EventCertainty reflect greater knowledge about the event. The average EventCertainty over all unevacuated travelers is a good way to measure the level of knowledge of those who are still in danger.

In Figure 4, we contrast our model for the spread of knowledge against a model in which instantaneous knowledge is assumed. The y-axis represents the average EventCertainty for all unevacuated travelers, while the x-axis represents the time step. With instantaneous knowledge, travelers are able to fully perceive the event immediately after it occurs regardless of where they are situated in the environment. Accordingly, the average EventCertainty jumps from 0 (no knowledge) to 2 (full knowledge) and remains at this level for the duration of the simulation. When knowledge is spread, the situation is much different. Immediately after the event, EventCertainty is low as only the travelers close by know that is has occurred. As time passes, knowledge of the event propagates through the crowd as travelers with information disseminate it to their neighbors. As a result, EventCertainty rises until it reaches a point where almost all travelers are fully aware of the event. From this point, EventCertainty decreases as travelers with knowledge of the event are able to evacuate leaving an increasingly higher proportion of travelers who are unaware of the event.

Throughout the evacuation, authority figures are patrolling for travelers to inform. However, if a traveler is particularly isolated they may never come into contact with an authority figure. Instantaneous knowledge is a common assumption in agent-based evacuation models, but humans are not omniscient. In comparison, our model for the spreading of knowledge provides a more realistic approximation of knowledge diffusion through crowds.

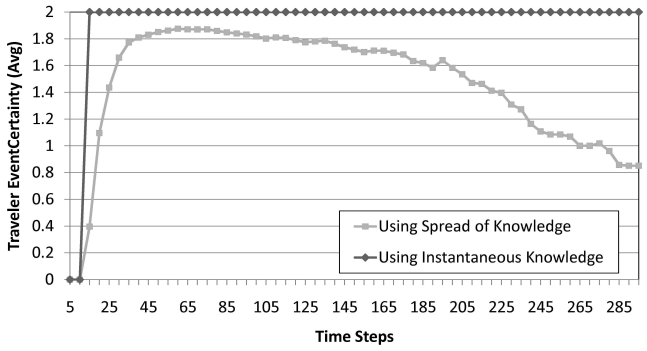


Figure 4: Effect of Knowledge Tranfer on EventCertainty

5.5 Authorities

Authority figures have been shown to exhibit a calming effect over crowds [21]. In the simulations, authority figures always have a low level of fear (FearFactor=1) and the highest level of knowledge about the event (EventCertainty=2). They then help to calm the crowd by passing these values onto all travelers they come into contact with. Thus, the presence of authority figures in the simulations should result in a lower level of fear among travelers. We can use the percentage of unevacuated travelers with the highest level of fear (FearFactor=2) as an inverse measure on the ability of authority figures to calm the crowd.

Figure 5 shows the effect of varying the number of authority figures on the FearFactor of travelers over the course of the evacuation. The y-axis represents the percentage of unevacuated travelers with FearFactor=2. Initially, there are no travelers with FearFactor=2. At the 15th time step, the percentage increases to include all travelers close to the event. This percentage continues to climb as a result of the contagion effect until it reaches a maximum between the 35th and 50th time steps. As time progresses, the effect of emotional contagion is balanced out by the influence of authority figures and the successful evacuation of travelers with FearFactor=2. From the results, we can see that increasing the number of the authority figures results in a lower percentage of travelers with FearFactor=2. With 6 authority figures, the percentage of travelers with FearFactor=2 reaches a maximum of 47%, whereas simulations with 8 and 10 authority figures reach maximums of 36% and 27%, respectively. Given that authority figures are distributed evenly, this is a logical result, as more authority figures provide for better spacial coverage. This in turn, increases both the likelihood and speed in which authority figures will inform travelers about the event. Thus, we have shown that authority figures in the simulations display a calming effect on travelers and increasing the number of authority figures only strengthens this effect.

5.6 SCT

It has been observed that Social Comparison leads people in close proximity to mimic the actions of the those around them [6]. In a crowd setting this would logically result in a grouping effect. The phenomenon of grouping within crowds has been well documented in research on pedestrian dynamics [10]. To measure the prevalence of localized grouping in the simulations, we introduce the notion of connectivity. A traveler's connectivity is equal to the number of neighboring travelers plus one. Travelers are considered to be neighbors if they are within a specified distance of each other. Thus, a traveler with a connectivity of 1 is considered to be isolated. As connectivity is a measure of grouping, we would expect to see an increase in the overall level of traveler connectivity by modeling

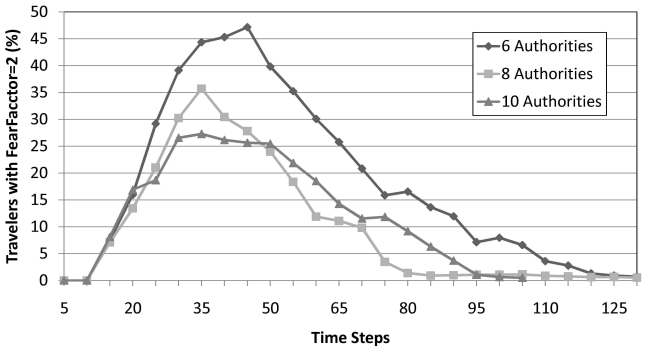


Figure 5: Effect of Authority Figures on FearFactor

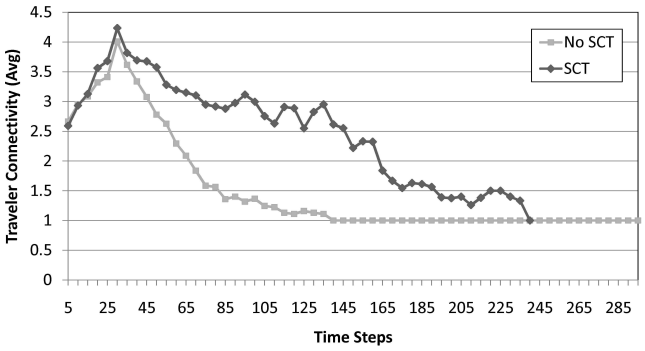


Figure 6: Effect of SCT on Connectivity

Social Comparison. The impact of Social Comparison on the average connectivity of all unevacuated travelers can be seen in Figure 6. Connectivity, both with and without Social Comparison, rises in the moments leading up to and following the event. Without Social Comparison, the level of connectivity then steadily drops as travelers begin to disperse and exit the terminal. This continues until the average level of connectivity reaches 1, which represents travelers being isolated. With Social Comparison, the level of connectivity declines at a much slower rate before also reaching 1. These results indicate that Social Comparison increases the level of connectivity and thus the amount of grouping displayed by travelers.

5.7 Los Angeles International Airport

Finally, we modeled the Tom Bradley International Terminal (TBIT) at Los Angeles International Airport as a realistic test scenario for our simulation environment. The scenario is approximately 55 times larger than the test case used in Section 5. Ideally, we would have liked to experiment on the full scenario and compare results with data from LAX, however, such data is not available. While lack of data is a major issue for most simulations in academia, the security domain presents an added level of difficulty due to confidentiality and national security concerns surrounding such data. Thus, for the tests in this section, we focused on one end of the terminal (the hallway and two gates, with one exit in each gate) and examined the impact of various authority policies with the aim of generating policy recommendations. We used 200 pedestrians, including 20 families of four, variable number of authorities, and two exits as the default case.

As a baseline test, we first ran experiments to examine the impact of increasing the number of authority figures as well as removing one exit from the scenario. We would expect that increasing the number of authority figures creates a calmer evacuation and removing an exit creates a more chaotic evacuation as more people squeeze towards fewer exits. Figure 7 shows the number of collisions (in thousands) under different parameter settings, where the number indicates the number of authorities in the setup and More/Less indicates whether an exit was added or removed from the base scenario. Higher bars indicate a more chaotic evacuation. All differences within a single authority setting, with the exception of 2-authority More vs 2-authority Normal, were statistically significant. As can be seen by the fact that the results are higher as we move to the right within a single authority setting, fewer exits lead to more chaotic evacuations. Comparing across authority settings, all differences within a single exit setting were statistically significant, with the exception of 4-authority vs 6-authority Less. As can be seen, fewer authorities leads to more chaotic evacuations as well. Both of these results are in line with expectation.

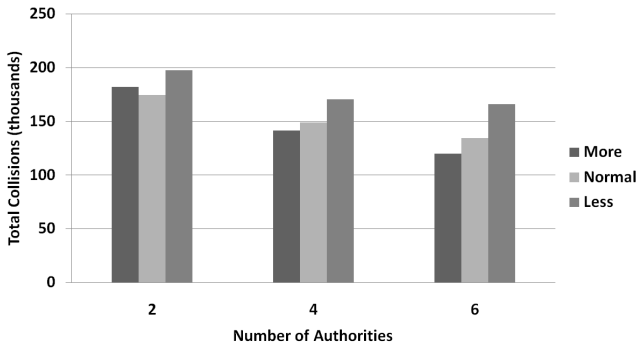


Figure 7: Effect of adding exits and authorities

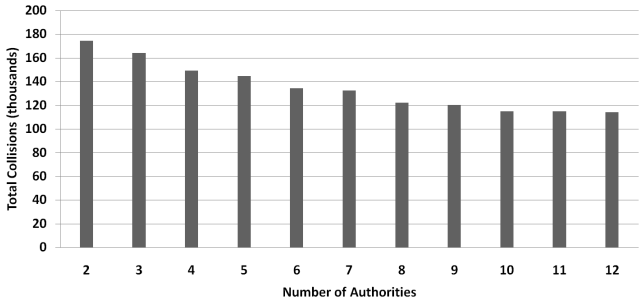


Figure 8: Effect of more authorities

Next, as per security officials' interest, we examined the impact of having more authority figures to aid in recommending how many are needed to safely evacuate this space. Figure 8 shows the number of collisions over the course of the evacuation (in thousands), with the number of authorities listed on the x-axis. T-Tests revealed that settings of more than 8 authority figures did not produce statistically significantly different results from the 8-authority case. This result implies that for this particular space, using more than 8 authorities would not produce better results.

We also ran tests with an alternate patrolling strategy. The default strategy is to proceed to a randomly chosen 'patrol point', the list of which is predefined to be the corners of each area in the scenario. The alternate strategy we tested was to have authority figures patrol the perimeters of the waiting areas and hallways. Results pertaining to the number of collisions were not statistically significantly different, implying no benefit to either strategy. However, further analysis revealed another trend.

Specifically, we looked at what percentage of the population would be reached by patrolling authorities on average within the first 300 time steps of the simulation. Figure 9 shows the per-

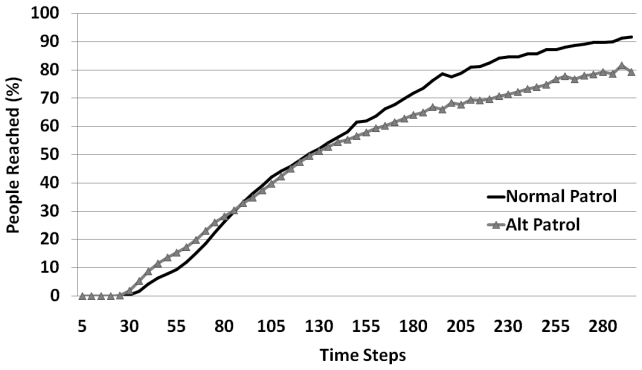


Figure 9: Effect of alternate patrol

centage of people that were reached by authorities within 300 time steps. We show only the case of 6 authority figures, but all like comparisons showed the same results (although varying in degree of the difference). Namely, the alternate strategy lines were always steeper at the beginning of the evacuation, but flattened out, implying that initially the alternate strategy was superior, but as fewer and fewer people remained, the point-to-point strategy was superior. Patrolling the edge of the room is effective to reach agents on the outskirts and more evenly distributes authority figures, but due to the large size of the waiting areas, crossing the room to reach different corners ultimately covers more ground. These results imply that a coordinated authority policy that intelligently covers the ground would be superior to both.

6. CONCLUSION

In this paper, we describe ESCAPES, a multiagent evacuation simulation tool that incorporates four key features: (i) different agent types; (ii) emotional interactions; (iii) informational interactions; (iv) behavioral interactions. These features are grounded in social psychology and evacuation research and tailored towards the needs of an airport security official (as well as other situations with similar features such as a mall, where homogenous agents are a poor approximation). Furthermore, as shown in Table 1, the features result in a breadth of emergent behaviors that have been observed in the literature, implying increased fidelity of our simulation as a result of their inclusion. We also show results based on a model of Los Angeles International Airport's Tom Bradley International Terminal with concrete recommendations that can be produced with our simulation.

In discussions with security officials affiliated with LAX, ESCAPES received high praise. Officials mentioned that the 3D visualization we provide is far superior for training and planning to other systems they have tried in the past. The inclusion of families and authorities as well as realistic knowledge spread about event and exits were specifically mentioned by them as being important and something they have not yet seen. The ability to adjust the number of families, pedestrians, and authorities in each zone was crucial. Overall, ESCAPES was very well received by security officials affiliated with LAX.

7. REFERENCES

[1] C. Burstedde, A. Kirchner, K. Klauck, A. Schadschneider, and J. Zittartz. Cellular automaton approach to pedestrian dynamics-application. In *Pedestrian and Evacuation Dynamics*, pages 87–97. Springer Berlin Heidelberg, 2002.

[2] J. M. Chertkoff and R. H. Kushigian. *Don't Panic: The Psychology of Emergency Egress and Ingress*. Praeger Publishers, 1999.

[3] J. Diamond, M. McVay, and M. W. Zavala. Quick, Safe, Secure: Addressing Human Behavior During Evacuations at LAX. Master's thesis, UCLA Department of Public Policy, June 2010.

[4] D.S.Mileti and J.L.Sorensen. Communication of emergency public warnings: A social science perspective and state-of-the-art assessment. 1990.

[5] R. F. Fahy and G. Proulx. Human behavior in the world trade center evacuation. In *International Association for Fire Safety Science, Fifth International Symposium*, pages 713–724, 1997.

[6] L. Festinger. A theory of social comparison processes. *Human Relations*, pages 117–140, 1954.

[7] J. P. Forgas. Affective influences on individual and group judgments. *European Journal of Social Psychology*, (20):441–453, 1990.

[8] N. Fridman and G. A. Kaminka. Comparing human and synthetic group behaviors: A model based on social psychology. In *ICCM-09*, 2009.

[9] E. Hatfield, J. T. Cacioppo, and R. L. Rapson. Cambridge University Press, 1994.

[10] D. Helbing, I. J. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[11] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282–4286, 1995.

[12] L. Henderson. On the fluid mechanics of human crowd motion. *Transportation research*, 8(6):509–515, 1974.

[13] M. Hoogendoorn, J. Treur, C. v. d. Wal, and A. v. Wissen. An Agent-Based Model for the Interplay of Information and Emotion in Social Diffusion. In *In IAT-10*, pages 439–444, New York, USA, 2010.

[14] J.L.Bryan. Behavioral response to fire and smoke. In *SFPE Handbook of Fire Protection Engineering*, pages 3315–3341. National Fire Protection Association, third edition, 2002.

[15] J. A. Kulik and H. I. M. Mahler. Social comparison, affiliation, and emotional contagion under threat. In *Handbook of social comparison: Theory and research*. New York: Plenum, 2000.

[16] Y. Lin, I. Fedchenia, B. LaBarre, and R. Tomastik. Agent-based simulation of evacuation: An office building case study. In *Pedestrian and Evacuation Dynamics 2008*, pages 347–357. Springer Berlin Heidelberg, 2010.

[17] N. Pelechano. Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation*, pages 21–30, 2005.

[18] N. Pelechano, J. Allbeck, and N. Badler. *Virtual Crowds: Methods, Simulation, and Control*. Morgan & Claypool Publishers, 2008.

[19] G. Proulx and R. F. Fahy. Human behavior and evacuation movement in smoke. *ASHRAE Transactions*, July 2008.

[20] H. E. Russell and A. Beigel. *Understanding Human Behavior for Effective Police Work*. Basic Books, 1976.

[21] C. A. Smith and P. C. Ellsworth. Patterns of cognitive appraisal in emotion. *Journal of Personality and Social Psychology*, 4(48):813–838, 1985.

Agent Communication

Commitments with Regulations: Reasoning about Safety and Control in REGULA

Elisa Marengo
Università degli Studi di Torino
emarengo@di.unito.it

Matteo Baldoni
Università degli Studi di Torino
baldoni@di.unito.it

Cristina Baroglio
Università degli Studi di Torino
baroglio@di.unito.it

Amit K. Chopra
Università degli Studi di Trento
chopra@disi.unitn.it

Viviana Patti
Università degli Studi di Torino
patti@di.unito.it

Munindar P. Singh
North Carolina State Univ.
singh@ncsu.edu

ABSTRACT

Commitments provide a flexible means for specifying the business relationships among autonomous and heterogeneous agents, and lead to a natural way of enacting such relationships. However, current formalizations of commitments incorporate conditions expressed as propositions, but disregard (1) temporal regulations and (2) an agent's control over such regulations. Thus, they cannot handle realistic application scenarios where time and control are often central because of domain conventions or other requirements.

We propose a new formalization of commitments that builds on an existing representation of events in which we can naturally express temporal regulations as well as what an agent can control, including indirectly as based on the commitments and capabilities of other agents. Our formalization supports a notion of commitment safety. A benefit of our consolidated approach is that by incorporating these considerations into commitments we enable agents to reason about and flexibly enact the regulations.

The main contributions of this paper include (1) a formal semantics of commitments that accommodates temporal regulations; (2) a formal semantics of the notions of innate and social control; and (3) a formalization of when a temporal commitment is safe for its debtor. We evaluate our contributions using an extensive case study.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; H.1.0 [**Information Systems**]: Models and Principles—*General*

General Terms

Theory

Keywords

Business process modeling, business protocols

1. INTRODUCTION

Previously, commitments have been studied over propositional languages [6, 7, 11, 19]. But in a number of practical settings, commitments involve rich temporal structure. Consider the following examples taken from a healthcare setting.

EXAMPLE 1. *An insurance company commits to reimbursing a covered patient for a health procedure provided the patient obtains approval from the company prior to the health procedure. Presumably, the patient would delay going in for the procedure until after having obtained an approval.*

EXAMPLE 2. *An insurance company commits to paying an in-network surgeon for a procedure only after a covered patient has undergone the procedure. Presumably, the surgeon would bill the insurance company after performing the procedure.*

As the following examples illustrate, temporal commitments can also involve more than two parties.

EXAMPLE 3. *A physician commits to a patient that if the patient has any sign of heart trouble after signing up with him, then the patient will be immediately referred to a laboratory for tests, the results of which will be evaluated by a specialist.*

EXAMPLE 4. *A pharmacy commits to provide medicine only if the patient obtains a prescription for that medicine.*

EXAMPLE 5. *For an out-of-network surgeon, an insurance company commits to paying the patient (instead of the surgeon) but only after the surgeon performs the procedure, the patient pays the surgeon, and the patient submits receipts to the insurance company.*

Temporal constraints such as those alluded to in Examples 1–5 are traditionally captured as procedural workflows. Instead, following recent approaches [8, 16], we think of such constraints more broadly as regulations and express them more flexibly in a logical notation. The commitments among autonomous parties capture their business relationships naturally. In contrast with existing approaches [3, 8, 10], we incorporate regulations as contents of commitments. By thus reifying regulations into business relationships, we bring normative force to the specification, thereby providing a clear basis for the participants to guide their actions locally and to judge the compliance of their counterparties. For example, if a regulation says that a physician's referral should precede a surgeon's procedure, it is not clear whether the physician is responsible for moving first or the surgeon is responsible for moving second. By placing the regulations in commitments, we make it explicit that it is the debtor of the commitment who needs to ensure its satisfaction. Further, in doing so, we can capture the business relationships (and concomitant regulations) in a flexible manner that avoids unnecessarily coupling or constraining the participants.

Cite as: Commitments with Regulations: Reasoning about Safety and Control in REGULA, E. Marengo, M. Baldoni, C. Baroglio, A. K. Chopra, V. Patti, M. P. Singh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 467-474.

The expression $\mathsf{C}(debtor, creditor, antecedent, consequent)$ means that the debtor commits to the creditor that if the antecedent holds, the consequent will hold. The antecedent and the consequent, i.e., the contents of a commitment, are typically logical expressions over states of the world [19], although some have added an explicit temporal component for expressing deadlines [5, 13].

In the present development, the content is specified over events. We use '$\cdot$' (center dot) as *before*, our main temporal operator on events, where $a \cdot b$ means that event a occurs before event b (though both occur eventually). Then we may express the commitments in Example 1 and 2 as $\mathsf{C}(ins, pat, approve \cdot perform, reimburse)$ and $\mathsf{C}(ins, sur, perform \cdot bill, pay)$, respectively. The commitment in Example 3 would be $\mathsf{C}(phy, pat, signup \cdot heartTrouble, test \cdot evaluate)$.

1.1 Challenges: Progression, Control, Safety

Placing temporal regulations within commitments enables us to identify precisely the responsibilities of the agents individually, and offer flexibility in terms of how the agents enact their commitments. By contrast, a purely temporal approach, such as Singh's [16], curtails the autonomy of the participants once the desired computations are specified. However, placing regulations inside commitments, as we did in the above examples, leads to new challenges. One, we must formalize the progression that is, the life cycle, of commitments, bearing in mind the events that have occurred. For example, we say that an active commitment $\mathsf{C}(x, y, r, u)$ progresses to *discharged* when u occurs. Analogously, we would like to say that $\mathsf{C}(x, y, r, e \cdot f)$ progresses to $\mathsf{C}(x, y, r, f)$ when e occurs. The challenge is to formalize general progression rules for an expressive event language. Two, a regulation expresses a constraint over the occurrence of events in a distributed system. The capability for bringing about the events, that is, the *control* of the events, would also generally be distributed among the agents. In Example 3, the physician commits to the patient for both testing and evaluation, but has control of neither—he must rely on a laboratory and a specialist for these tasks. Further, the physician commits to the coordination constraint that the testing will occur before the evaluation. Clearly, the physician is committing to activities over whose performance he apparently has no control. What would make the physician's commitment reasonable?

In general, an agent would want to commit only to temporal conditions over which it exercises adequate control. We distinguish between two kinds of control: *innate* and *social*. In the above example, the laboratory and the specialist have innate control over testing and evaluation, respectively. Social control ties in with commitments: for regulations specified over events that the debtor does not have control over, the debtor would need the appropriate commitments from those who have control. The physician would have control over testing and evaluation if he could get the appropriate commitments from the laboratory and the specialist. For example, $\mathsf{C}(lab, phy, \top, test)$ and $\mathsf{C}(specialist, phy, \top, evaluate)$ would give the physician social control over the two events; however, the physician really needs $\mathsf{C}(specialist, phy, \top, test \cdot evaluate)$ from the specialist in order to ensure the appropriate event order.

Control in turn motivates the notion of the *safety* of a commitment. A commitment is safe if its debtor has established sufficient control to guarantee being able to discharge it (assuming others discharge commitments of which they are the debtors). For example, without the above commitments from the laboratory and the specialist, the physician's commitment to the patient would be unsafe. As our examples illustrate, the notions of control and safety are especially relevant for understanding engagements among more than two parties. How can we determine whether the debtor of a commitment is able to apply the requisite control so as to ensure that its commitments have the support of the other agents so that together they satisfy a given regulation, thereby accomplishing the coordination envisaged by the regulation?

1.2 Contributions

Our contributions may be summarized as follows. First, we formalize commitments with regulations in a simple but expressive event-based model. We formalize rules that capture the *progression* of a commitment over runs (sequences of events), using a previous sound and complete residuation reasoner. Second, we formalize *control* and *safety* in the same event-based model. In particular, we formalize a notion of social control via commitments, which naturally matches multiagent settings. Third, we *declaratively formalize* the *life cycle* of commitments, captured by Theorem 1. Moreover, we connect the notion of commitment progression with the notions of control and of safety (Theorems 2 and 3). Specifically, as long as the agents cause events that are expected by the application of the definition of safety itself, safety is preserved. We evaluate the proposed notions by formalizing Robert's Rules of Order [14] (RONR), one of the best known set of laws for managing the proceedings of democratic parliamentary assemblies.

Organization

The paper is organized as follows: Section 2 presents the theoretical background necessary for our formalization; Section 3 contains the main theoretical results, concerning the notions of progression, control and safety; Section 4 reports our case study; Section 5 concludes with a discussion and a review of the relevant literature.

2. TECHNICAL FRAMEWORK

Previous works on events and on commitments are relevant here. However, the approaches of interest are not mutually compatible at a technical level. On the one hand, to reason incrementally about control and progression, we need a powerful notion of events and residuation whose semantics is given with respect to an event run [16]. On the other hand, to represent conditional, active commitments, we need an approach based on a state-based semantics given with respect to a state and an index on it [17]. Thus one of the challenges our framework addresses is reconciling the above. As a result, although our approach borrows ideas from Singh [17], our formal model and its details are novel to this paper.

2.1 Precedence logic

Precedence logic is an event-based logic [16]. It has three primary operators for specifying requirements about the occurrence of events: '$\vee$' (choice), '$\wedge$' (concurrence), and '$\cdot$' (before). The *before* operator enables one to express specifications such as $approve \cdot perform$: both $approve$ and $perform$ must occur and in the specified order. The specifications are interpreted over runs. Each run is a sequence of events. Figure 1 shows a schematic of our model. The transitions correspond to event occurrences (the $\bullet$ symbols merely identify place holders between consecutive events: on each run, each $\bullet$ corresponds to an index). The model shows several runs, of which it identifies τ_0, τ_1, and τ_2, which all begin with ab. Additional runs include all the suffixes of τ_0, τ_1, and τ_2—for example, bef and bcd_x (an event subscript indicates which agent has the capability to perform the event; thus, d_x means that x has the capability to perform d). The same point may be identified with different indices on different runs. For example, the point after b has index 2 on $\tau_0 = abcd_x \ldots$ and index 0 on $gh \ldots$ (the top branch).

Let e be an event. Then $\overline{e}$, the complement of e, is also an event. Initially, neither e nor $\overline{e}$ hold. On any run, either e or $\overline{e}$ may oc-

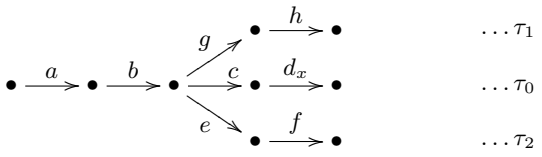


Figure 1: A schematic of runs with common prefixes.

cur, not both. We assume that events are nonrepeating. In practice, transaction IDs or timestamps differentiate multiple instances of the same event. This yields the following advantages. First, since we want to talk about precedence, it helps avoid the confusion where a preceding b would be consistent with b preceding a. Second, it supports negative events as occurring, and distinct from (and stronger than) a positive event not having occurred yet. Third, it facilitates a simpler language and logic that is nevertheless adequate for capturing several regulations of practical interest.

2.2 Language

We distinguish between physical and social (or institutional) events. Our specifications are limited to the physical events (those that are publicly performed by an agent). For example, the events in Figure 1 are physical events: think of a as a waiter pushing a menu card to a customer over a counter in a diner and b as the customer pushing \$5 back to the waiter. However, we supplement physical events with a set of means axioms, which capture the notion of *counts as* [2, 15]. We take the social events corresponding to a physical event to occur concurrently with the physical event. In this manner, we respect Goldman's notion of (conventional) *generation* [12]. For example, a in Figure 1 may mean the creation of an offer to sell a coffee for \$5, which thus happens simultaneously with a. Likewise, b may mean accepting the offer created by a.

We propose a language, REGULA, in which the antecedents and consequents of commitments are event expressions. In intuitive terms, a commitment itself remains a state expression and we do not express it directly in our language of events. Instead, we think of the operations of commitments as first-class entities, i.e., as events, and let the resulting commitments stay in the background. In other words, we can think of an operation such as $\mathsf{Create}(x, y, r, u)$ as a social event that brings about the corresponding commitment.

Below, $\mathcal{X}$ yields agent names, **E** yields event types, and *param* yields domain values using which an event instance is specified from an event type. In conceptual terms, an event type may be either (1) of sort *physical*, in which case we optionally specify the agent who has the capability to perform it (an unspecified agent indicates we do not care about the agent) or (2) of sort *social*, in which case it is an operation on commitments. For brevity, "event" means "event instance" throughout. The syntax of REGULA is:

$\textsc{Regula} \longrightarrow \mathit{axiom}\{\ ,\mathit{axiom}\ \}$
$\mathit{axiom} \longrightarrow \langle\langle \mathit{physical}\ \textbf{means}\ \mathit{social}\rangle\rangle$
$\mathit{physical} \longrightarrow \mathbf{E}\langle[\mathcal{X},]\mathit{param}^*\rangle$
$\mathit{social} \longrightarrow \mathit{op}(\mathcal{X}, \mathcal{X}, \mathit{regulations}, \mathit{regulations})$
$\mathit{op} \longrightarrow \mathsf{Create} \mid \mathsf{Cancel} \mid \mathsf{Release} \mid \mathsf{Assign} \mid \mathsf{Delegate}$
$\mathit{regulations} \longrightarrow \mathit{regulation}\ \{\ \wedge\ \mathit{regulation}\ \}$
$\mathit{regulation} \longrightarrow \mathit{sequence}\ \{\ \vee\ \mathit{sequence}\ \}$
$\mathit{sequence} \longrightarrow 0 \mid \top \mid \mathit{physical} \mid \mathit{physical}\ \cdot\ \mathit{physical}$

That is, a REGULA specification is a set of axioms asserting which physical events count as which social events. In well-formed axioms, we require that the performing agent of the physical and corresponding social event be the same.

We use the following conventions: x, etc. are agents, e, etc. are physical events, r, s, u, etc. are regulations, τ, etc. are runs, and i, etc. are indices into runs. We drop agent names when they are understood. The above grammar limits sequences to two events each to simplify our formalization. However, in practice we write longer sequences because $e_1 \cdots e_n \equiv (e_1 \cdot e_2) \wedge \ldots \wedge (e_{n-1} \cdot e_n)$.

2.3 Model and Semantics

We now describe the semantics of REGULA in terms of a model, $M = \langle \mathbb{E}, \mathbb{T}, \mathbb{C}, \mathbb{D}, \mathbb{X}, \mathbb{V}\rangle$. Here $\mathbb{E}$ and $\mathbb{T}$ describe the physical layer, $\mathbb{C}$ describes the social (commitment) layer.

- $\mathbb{E}$ is a denumerable set of possible event instances closed under complementation. That is, $e \in \mathbb{E}$ if and only if $\overline{e} \in \mathbb{E}$. For simplicity in our notation, we identify e and $\overline{\overline{e}}$; thus wherever we write e in the semantics, it applies both to e and $\overline{e}$. The set $\mathbb{E}$ can itself be generated from event types and their parameters, as described above. Further, we introduce a special symbol ϵ for a null event.

- $\mathbb{T} = \{\tau | \tau : \mathbb{N} \mapsto \mathbb{E} \cup \{\epsilon\}$, τ is (1) injective, and (2) $(\forall i, j : \tau_i \neq \overline{\tau_j})\}$ is the set of possible event runs (we write the i^{th} event in τ as τ_i). $\mathbb{N}$ is the set of natural numbers. Thus each member of $\mathbb{T}$ is a sequence of events. The above constraints restrict $\mathbb{T}$ to *legal* runs [16] wherein (1) no event repeats and (2) no event and its complement both occur.

  We use the null event ϵ to indicate the termination of a run. If ϵ ever occurs on a run, all subsequent events are ϵ. That is, $(\forall i, j : j \geq i$ and $\tau_i = \epsilon \Rightarrow \tau_j = \epsilon)$. Below $|\tau|$ is the length of τ, and equals the smallest index i for which $\tau_i = \epsilon$ if i exists and is ω otherwise (indicating an infinite run).

  Notice that $\mathbb{T}$ is suffix-closed, meaning that if a run belongs to $\mathbb{T}$, then so does each suffix of it. Formally, using s as the successor function for $\mathbb{N}$ and $\odot$ as functional composition, we have $\{\tau \odot s | \tau \in \mathbb{T}\} \subseteq \mathbb{T}$. Below, $[i, j]$ refers to the subrun between the i^{th} and the j^{th} events, both inclusive. Likewise, $\mathbb{T}$ is prefix-closed. That is, (using the fact that all natural numbers are less than ω), we have $\{\tau_{[0,j]} | \tau \in \mathbb{T}$ and $j \leq |\tau|\} \subseteq \mathbb{T}$.

- $\mathbb{C} : \mathbb{T} \times \mathbb{N} \times \mathcal{X} \times \mathcal{X} \times \wp(\mathbb{T}) \mapsto \wp(\wp(\mathbb{T}))$ is the standard for (active) commitments. That is, at each index on each run, for each debtor-creditor (ordered) pair of agents, $\mathbb{C}$ assigns to a set of runs a set of set of runs. The intuition is that $\mathbb{C}$ determines which conditional commitment is active from a debtor to a creditor at an index in a run. Given a potential antecedent, each of the consequents is placed in the set that is the output of this function. If the output set is empty at a run and an index that means no commitments are active there. We lack the space to include additional semantic (closure) constraints on $\mathbb{C}$ along the lines of Singh [17].

  (For readability, we place the agents as subscripts.) If two runs are equal until i then $\mathbb{C}$ yields the same result for each of them at index i. Formally, $(\forall \tau, \tau', i, R \subseteq \mathbb{T} : \tau_{[0,i]} = \tau'_{[0,i]} \Rightarrow \mathbb{C}_{x,y}(\tau, i, R) = \mathbb{C}_{x,y}(\tau', i, R))$.

- $\mathbb{D}, \mathbb{X}, \mathbb{V}$ with the same signature as $\mathbb{C}$ are respectively the standards for discharged, expired, and violated commitments.

The following constraints on our model capture some of the essential intuitions about commitments. Let $\textbf{cone}(\tau, i) = \{\tau' | \tau_{[0,i]} = \tau'_{[0,i]}\}$. In intuitive terms, the cone of a run at an index includes all possible future branches given the history (the part of the run up to the index). Because $\mathbb{T}$ contains all possible legal runs, the intuition

is that when a regulation is true (respectively, false) on all runs on a cone, then it is definitely true (respectively, definitely false). For example, e is false at index i of a run τ if it has not occurred yet; it is definitely false if e would occur on no runs in τ's cone at i, meaning that $\overline{e}$ must already have occurred.

- $U \in \mathbb{D}_{x,y}(\tau, i, R)$ only if $\tau_{[0,i]} \in U$

  The consequent of a discharged commitment must be true.

- $U \in \mathbb{X}_{x,y}(\tau, i, R)$ only if $\mathbf{cone}(\tau, i) \subseteq (\mathbb{T} \setminus R)$

  An expired commitment must have its antecedent definitely false. In other words, we do not just want that the antecedent is not yet true, we want it to never become true given the run so far.

- $U \in \mathbb{V}_{x,y}(\tau, i, R)$ only if $\mathbf{cone}(\tau, i) \subseteq (\mathbb{T} \setminus U)$ and $\tau_{[0,i]} \in R$

  A commitment is violated if its antecedent holds but its consequent is false.

- $U \in \mathbb{C}_{x,y}(\tau, i, R)$ only if $U \notin (\mathbb{D}_{x,y}(\tau, i, R) \cup \mathbb{X}_{x,y}(\tau, i, R) \cup \mathbb{V}_{x,y}(\tau, i, R))$

  An active commitment is one that is not discharged, expired, or violated.

Semantic postulates M_1–M_5, loosely based on Singh [16], address the temporal aspects of our language.

M_1. $\tau \models_i \top$

M_2. $\tau \models_i e$ iff $(\exists j \geq i : \tau_j = e)$, where e is a physical event

M_3. $\tau \models_i r \vee u$ iff $\tau \models_i r$ or $\tau \models_i u$

M_4. $\tau \models_i r \wedge u$ iff $\tau \models_i r$ and $\tau \models_i u$

M_5. $\tau \models_i r \cdot u$ iff $(\exists j \geq i : \tau \models_{[i,j]} r$ and $\tau \models_{[j+1,|\tau|]} u)$

It is helpful to expand the notion of complementation to apply to regulations, not just to physical events. To this end, we define complementation via a set of inference rules as follows: (1) $\overline{r \wedge u} = \overline{r} \vee \overline{u}$; (2) $\overline{r \vee u} = \overline{r} \wedge \overline{u}$; (3) $\overline{r \cdot u} = \overline{r} \vee \overline{u} \vee u \cdot r$; and (4) $\overline{\overline{e}} = e$. The interesting rule is the one for $\overline{r \cdot u}$, which captures that $r \cdot u$ may fail to occur exactly when one of its components does not occur or they both occur but in the reverse order.

2.4 Residuation

In simple terms, residuation is a way for us to track progress in the real world. The residual of a regulation with respect to an event is the "remainder" regulation that would be left over from the original after the event, and whose satisfaction would guarantee the satisfaction of the original regulation.

For example, let $r = \overline{a} \vee b \cdot a$ be a regulation under consideration; r means that either a cannot occur, because $\overline{a}$ occurred, or b and a both occur with b preceding a. If we residuate r by an event g, which does not occur in r, the result is the same as r, indicating that the desired regulation is unaffected by irrelevant events. Residuating r by b yields a, meaning that going forward a remains the only possibility. A subsequent occurrence of a would residuate this to $\top$, meaning that the regulation is satisfied on this execution. Residuating r directly by a yields 0, meaning that the occurrence of a has caused a violation of the regulation.

Following Singh [16], we can define the residual of a regulation r with respect to a physical event e as the maximal (most flexible) regulation such that an occurrence of e followed by an occurrence of the residual guarantees the original regulation. A benefit of using the event-based semantics is that it supports a set of simple equations or rewrite rules through which we can symbolically calculate the residual of a regulation given an event. The following equations are due to Singh [16]. Here, r is a sequence expression, and e is a physical event or $\top$. Below, Γ_u is simple the set of literals and their complements mentioned in u. Thus $\Gamma_e = \{e, \overline{e}\} = \Gamma_{\overline{e}}$ and $\Gamma_{e \cdot f} = \{e, \overline{e}, f, \overline{f}\}$.

$0/e \doteq 0$
$\top/e \doteq \top$
$(r \wedge u)/e \doteq ((r/e) \wedge (u/e))$
$(r \vee u)/e \doteq ((r/e) \vee (u/e))$
$(e \cdot r)/e \doteq r$, if $e \notin \Gamma_r$
$r/e \doteq r$, if $e \notin \Gamma_r$
$(e' \cdot r)/e \doteq 0$, if $e \in \Gamma_r$
$(\overline{e} \cdot r)/e \doteq 0$

The above equations characterize the progression of regulations under physical events. They have some important properties, including that (1) regulations not mentioning an event are independent of that event; (2) conjoined or disjointed regulations can be treated modularly; and (3) regulations can be incrementally progressed: hence a residuated regulation embodies the relevant history and no additional history need be represented.

We define the *intension* of r as the set of runs where it is true on index 0: $[\![r]\!] = \{\tau | \tau \models_0 r\}$. As an auxiliary definition, for a set of runs R, let $R \downarrow e = \{\nu \in \mathbb{T} | (\forall \upsilon : \upsilon \in [\![e]\!] \Rightarrow \upsilon\nu \in R)\}$. Then, following Singh [16], we can capture residuation semantically as $[\![r/e]\!] = [\![r]\!] \downarrow e$.

2.5 Commitments

When the antecedent is $\top$ (*true*), we refer to the commitment as being *unconditional*. An unconditional commitment usually arises because a conditional commitment was detached. For example, $\mathsf{C}(merchant, customer, paid, goods)$ gives rise to $\mathsf{C}(merchant, customer, \top, goods)$ when *paid* holds. We briefly mention some important stages in the life cycle of a commitment, that is, its progression. A commitment holds either because of an explicit Create operation by the debtor or because an existing commitment was detached. It is considered *expired* if the antecedent has expired (it cannot be satisfied anymore), meaning that the creditor did not take up the offer entailed by the commitment. A commitment is considered *violated* if it is unconditional and its consequent has expired, meaning that the debtor did not fulfill the offer. Alternatively, if the consequent holds, it is considered *discharged*.

Although we adopt many of the intuitions of the previous works, our technical development is significantly different in that we model commitments in an event-based framework. Doing so is nontrivial but yields rewards in an improved characterization of the progression of commitments than previously available.

Now we enhance the above development to accommodate commitments. The commitment operator C is not included in our language but provides a useful basis for the social operations.

M_6. $\tau \models_i \mathsf{C}(x, y, r, u)$ iff $[\![u]\!] \in \mathbb{C}_{x,y}(\tau, i, [\![r]\!])$

The meaning of the physical events in terms of social events is defined as follows. This simply states that whenever a physical event occurs the corresponding social event occurs as well. We leave open the possibility that a social even could occur implicitly without the matching physical event.

M_7. $\tau \models_i e_x \; means \; Op(x, y, r, u)$ iff $(\forall j \geq i : \tau \models_j e_x \Rightarrow \tau \models_j Op(x, y, r, u))$

Now we can state the meanings of the operations in terms of how they change the social state by manipulating commitments. We describe Create for concreteness and omit the rest for brevity.

M_8. $\tau \models_i \mathsf{Create}(x, y, r, u)$ iff $\tau \not\models_i \mathsf{C}(x, y, r, u)$ and $\tau \models_{i+1} \mathsf{C}(x, y, r, u)$

We impose a restriction on our model capturing that commitments persist until they are discharged, expired, or violated. Formally, $\forall \tau, \tau', i, R, U \subseteq \mathbb{T}$ if $U \in \mathbb{C}_{x,y}(\tau, i, R)$ and $\tau_i = e$, then:

1. $U \downarrow e \in \mathbb{X}_{x,y}(\tau, i, R \downarrow e)$ if $R \downarrow e = \emptyset$;
2. $U \downarrow e \in \mathbb{V}_{x,y}(\tau, i, R \downarrow e)$ if $R \downarrow e = \mathbb{T}$ and $U \downarrow e = \emptyset$;
3. $U \downarrow e \in \mathbb{D}_{x,y}(\tau, i, R \downarrow e)$ if $U \downarrow e = \mathbb{T}$;
4. $U \downarrow e \in \mathbb{C}_{x,y}(\tau, i, R \downarrow e)$ otherwise.

3. THEORETICAL RESULTS

We now present the main theoretical results on REGULA.

3.1 Residuation for Commitment Progression

Although commitment expressions are not event expressions in REGULA, we can use residuation to compute the progression of a commitment. For example, consider a commitment $c_1 = \mathsf{C}(x, y, b, d \cdot c \cdot a)$ and assume that events d, c, a occur in this order. In intuitive terms, after d, the antecedent of c_1 would be unaffected whereas its consequent would progress to $c \cdot a$. If c were to occur then, the antecedent would still be unaffected, but the consequent would reduce to a; then when a occurs, the consequent would reduce to $\top$, indicating that c_1 is discharged. Alternatively, assume that events d, a, b, c occur in this order. Now after d and a, the antecedent would still be unaffected, but the consequent would reduce to 0, indicating that commitment c_1 is violated.

In essence, the idea is that a commitment progresses as its antecedent and consequent are residuated by the events as they occur. The foregoing intuition can be thought of as distributing residuation into the antecedent and consequent of a commitment. This leads us to Theorem 1 on commitment progression. This theorem is technically trivial but important because it shows how a commitment progresses. Here, we assume that operators exp, vio, and dis are defined analogously to C though based on $\mathbb{X}$, $\mathbb{V}$, $\mathbb{D}$, respectively.

THEOREM 1. *If* $\tau \models_i \mathsf{C}(x, y, r, u)$ *and* $\tau_i = e$, *then*

$$\tau \models_{i+1} \begin{cases} \mathsf{exp}(x, y, r/e, u/e) & \text{if } r/e \doteq 0 \\ \mathsf{vio}(x, y, r/e, u/e) & \text{if } r/e \doteq \top,\ u/e \doteq 0 \\ \mathsf{dis}(x, y, r/e, u/e) & \text{if } u/e \doteq \top \\ \mathsf{C}(x, y, r/e, u/e) & \text{otherwise} \end{cases}$$

Proof sketch: Trivial by construction of $\mathbb{C}$, $\mathbb{X}$, $\mathbb{V}$, and $\mathbb{D}$.

3.2 Control

Consider $\mathsf{C}(x, y, \top, a_x \cdot b_y)/a_x$, yielding $\mathsf{C}(x, y, \top, b_y)$. Now x is committed to b_y, for which it depends on y. The key challenge here is one of *control*, whether an agent can bring about an event or complex action so as to detach or discharge a given commitment. Our intuition is that control is a combination of capability and opportunity. An agent may control an event innately, i.e., based on which events it can perform, or socially, i.e., based on the commitments of others and what they control. We define the intuitive notion of control $\xi(.,.)$ (our primary definition) in two mutually recursive parts. First, we capture the base cases of control through $\zeta(.,.)$, which captures the notion of innate control.

M_9. $\tau \models_i \zeta(x, \top)$

M_{10}. $\tau \models_i \zeta(x, e_x)$ iff $(\exists \tau' \in \mathbf{cone}(\tau, i) : \tau' \models_i e_x)$

Notice we refer to $\mathbf{cone}(\tau, i)$ above because it serves as a surrogate for notion of state, which is otherwise not present in our framework. Thus, merely finding a τ' on which event e_x occurs at index i would not be enough.

M_{11}. $\tau \models_i \zeta(x, e_y)$, where $x \neq y$, iff $(\exists r : \tau \models_i \xi(x, r)$ and $\tau \models_i \mathsf{C}(y, x, r, e_y)$ and $\tau \models_i \xi(y, e_y))$

Second, we formulate control recursively using the above.

M_{12}. $\tau \models_i \xi(x, r \vee u)$ iff $\tau \models_i \xi(x, r)$ or $\tau \models_i \xi(x, u)$

M_{13}. $\tau \models_i \xi(x, r \wedge u)$ iff $\tau \models_i \xi(x, r)$ and $\tau \models_i \xi(x, u)$

M_{14}. $\tau \models_i \xi(x, r \cdot u)$ iff $\tau \models_i \xi(x, r)$ and $(\forall \tau' \in \mathbf{cone}(\tau, i) : \tau' \models_i r \Rightarrow \tau' \models_i (r \cdot \xi(x, u)))$

M_{15}. $\tau \models_i \xi(x, r)$ iff $\tau \models_i \zeta(x, r)$, when r is not of the form $r \vee u$, $r \wedge u$, or $r \cdot u$

Given that an agent controls a regulation, the question is whether it is possible that such control be propagated along the execution, as the regulation is residuated based on the occurred events.

THEOREM 2. *If* $\tau \models_i \xi(x, r)$ *then* $(\exists \tau', e : \tau' \in$ ***cone***(τ, i) *and* $\tau'_i = e$, *and* $\tau' \models_{i+1} \xi(x, r/e))$.

Proof sketch: Follows directly from the definition of control.

Notice that the preservation of control requires some cooperation of the involved agents. For instance, $\xi(x, e_y)$ requires that y continues to support x. In other words, either y causes e_y at some point or y does not cancel its commitment to x to execute e_y when some condition becomes true.

Informally, by a Create, the debtor provides social control to the creditor; by performing a Cancel the debtor takes back social control; by performing a Release the creditor relinquishes social control. Likewise, Assign and Delegate transfer control suitably.

3.3 Safety

Safety is a property of commitments. Since the regulations embedded in commitments involve many actors, it is important for the potential debtor to understand when it is "reasonable" for it to commit. Intuitively, this is reasonable when the agent controls the events that are part of the regulation. In other words, a commitment is safe for its debtor when the coordination necessary to fulfill the regulation is supported by commitments by the other agents involved. We can thus define the *safety of a commitment* $\mathsf{C}(x, y, r, u)$ *for the debtor agent* x as $\sigma(x, \mathsf{C}(x, y, r, u))$ as follows:

M_{16}. $\tau \models_i \sigma(x, \mathsf{C}(x, y, r, u))$ iff $(\forall \tau' \in \mathbf{cone}(\tau, i)$ and $(\tau' \models_i \xi(x, \overline{r})$ or $(\mu j \geq i : \tau' \models_{[i,j]} r \Rightarrow \tau' \models_j \xi(x, u/\tau'_{[i,j]}))))$

Residuation by subrun $\tau'_{[i,j]}$ is a shorthand notation standing for the residuations, in sequence, by all the events in the subrun. By μj we refer to a generalized quantifier that selects the least such j index. So, a commitment is safe for its debtor if either the debtor controls the negation of the antecedent or whenever the antecedent holds, the debtor controls the residuation of the consequent. In the former case, the debtor can act so as to avoid letting the commitment become active. In the latter case, instead, when the commitment becomes active, there is a way to satisfy it. Residuation is necessary because at j some event that is in the consequent might have occurred already.

Notice that the definition of safety does not depend directly on the given run but consider all runs of the same history. The intuition here is to capture the fact that safety is essentially a state property, even though we express it in an event-based model. The definition is symmetric between all the runs that have the same history as τ.

Moreover, safety does not mean that no matter how bad a decision an agent takes, success is guaranteed; just that the agent is not subject to the whims of another agent. In other words, the agent can prevent a bad situation, not that a bad situation is impossible.

THEOREM 3. *If* $\tau \models_i \sigma(x, \mathsf{C}(x, y, r, u))$ *then* $(\exists \tau' \in \textbf{\textit{cone}}(\tau, i)$, e *and* $\tau' \models_i \overline{r}$ *or* $(\tau' \models_i r$ *and* $\tau'_i = e$ *and* $\tau' \models_{i+1} \sigma(x, \mathsf{C}(x, y, r/e, u/e))))$.

In words, Theorem 3 states that if at some point on a run a commitment is safe for an agent, then there is a possible continuation such that the residuation of the commitment remains safe.

Proof: Follows from Theorem 2 and the definition of safety. Suppose that $\tau \models_i \sigma(x, \mathsf{C}(x, y, r, u))$ holds. Therefore, $(\forall \tau' \in \textbf{cone}(\tau, i)$ and $\tau' \models_i \xi(x, \overline{r})$ or $(\mu j \geq i : \tau' \models_{[i,j]} r \Rightarrow \tau' \models_j \xi(x, u/\tau'_{[i,j]})))$ by M_{16}. We have two cases. *First case.* Let us suppose that $\tau' \models_i \xi(x, \overline{r})$. By Theorem 2, $(\exists \tau'' \in \textbf{cone}(\tau', i), e$ and $\tau''_i = e$, and $\tau'' \models_{i+1} \xi(x, \overline{r}/e))$ and this proves the case. *Second case.* Let us suppose that $(\mu j \geq i : \tau' \models_{[i,j]} r \Rightarrow \tau' \models_j \xi(x, u/\tau'_{[i,j]}))$. Thus, whenever $\tau' \models_{[i,j]} r$ we have $\tau' \models_j \xi(x, u/\tau'_{[i,j]})$. By Theorem 2, $(\exists \tau'' \in \textbf{cone}(\tau', j+1), e$ and $\tau''_i = e$, and $\tau'' \models_{j+1} \xi(x, (u/\tau'_{[i,j]})/e))$. Let us consider the previous τ'' and e and assume that $\tau'' \models_{[i,j+1]} r/e$ holds. Then, we have that $\tau'' \models_i \xi(x, (u/\tau'_{[i,j]})/e)$. This proves the theorem.

4. CASE STUDY

We apply our approach on *Robert's New Rules of Order* (RONR) [14], a system of parliamentary laws. RONR posits two roles: *chair* and *participants*. The activity of the assembly consists of discussing a motion at a time, and then voting. The rules are aimed at guaranteeing that the assembly works in a democratic way. Among other rules, in particular, it specifies that voting will not take place until all the participants who raised their hand for expressing their opinion have spoken. Different members are not allowed to speak at the same time and, in particular, in order to speak one must have the floor. As long as everybody *behaves* according to the rules, the assembly works in a democratic way. In other terms, RONR not only specifies the actions but also governs the behavior of the participants and the chair (specifying the contexts in which the execution of actions makes sense) so as to guarantee the success of the assembly if all the agents behave according to RONR. Each participant autonomously decides whether to conform to the rules, but doing so confers some rights on the participant.

The first column of Table 1 lists physical events that can occur during an enactment of RONR (the subscript indicates which agent directly controls the event: c stands for chair and p_i generically stands for participant). Recall that given two agents p_1 and p_2, the event instance e_{p_1} is different from e_{p_2}. So, for instance, $askFloor_{p_1}$ is different from $askFloor_{p_2}$. A possible specification of the semantics of events is given in terms of their effects on the social state. The second column of Table 1 reports event effects in terms of commitments that are created by their occurrence. Antecedents and consequents are written in REGULA. The social effects are operations on commitments. Besides Create, in the example, we also use Assign and Delegate: a participant can delegate its vote or assign its time slot for speaking to another participant.

Notice that if the meaning were given using propositional commitments, one could not express temporal regulations. For instance, to express that the floor is given after it is asked for, the commitment $\mathsf{C}(c, p_i, askFloor_{p_i}, giveFloor_c\langle p_i\rangle)$ would be inadequate since it does not ensure that the two events occur in the expected order. Potentially, the chair could give the floor to p_i before p_i asked for it and the commitment would be discharged. Such apparent flexibility may be desirable in some settings but not where it violates a regulation.

Physical event	Means these social events
openAssembly$_c$	$\forall p_i \in P$, Create($\mathsf{C}(c, p_i, \top$, $exposeMotion_c\langle m\rangle \cdot openDebate_c\langle m\rangle)) \wedge$ $\forall p_i, p_j \neq p_i \in P$, Create($\mathsf{C}(c, p_i$, $discuss_{p_j} \wedge \overline{giveFloor_c\langle p_j\rangle}$ $\vee discuss_{p_j} \cdot giveFloor_c\langle p_j\rangle$, $punish_c\langle p_j\rangle)))$
openDebate$_c\langle$m$\rangle$	$\forall p_i \in P$, Create($\mathsf{C}(c, p_i, askFloor_{p_i}$, $askFloor_{p_i} \cdot giveFloor_c\langle p_i\rangle))) \wedge$ $\forall p_i \in P$, Create($\mathsf{C}(c, p_i$, $askFloor_{p_i} \cdot giveFloor_c\langle p_i\rangle \cdot discuss_{p_i}$ $\vee \overline{askFloor_{p_i}}$, $askFloor_{p_i} \cdot giveFloor_c\langle p_i\rangle \cdot$ $discuss_{p_i} \cdot cfv_c$ $\vee \overline{askFloor_{p_i}} \cdot cfv_c))$
cfv$_c$	none
enterAssembly$_p$	Create($\mathsf{C}(p, c, cfv_c, cfv_c \cdot vote_p)$)
askFloor$_p$	Create($\mathsf{C}(p, c, \top, discuss_p)$)
exposeMotion$_c\langle$m$\rangle$	none
discuss$_p$	none
giveFloor$_c\langle p\rangle$	none
passFloor$_{p_i}\langle p_j\rangle$	Assign($p_j, \mathsf{C}(c, p_i, \top, giveFloor_c\langle p_i\rangle)$)
vote$_p$	none
delegateVote$_{p_i}\langle p_j\rangle$	Delegate($p_j, \mathsf{C}(p_i, c, \top, vote_{p_i})$)
close_cfv$_c$	none
closeAssembly$_c$	none
punish$_c\langle$p$\rangle$	none

Table 1: RONR physical events mapped to their social effects. Here p_i are participants, c the chair, and m a motion.

The following is an example commitment whose antecedent or consequent use all the allowed operators of REGULA. In this case, c commits to each p_i that c would punish any other p_j if p_j starts speaking when the chair refused to give it the floor or speaks before having the floor.

$$C_1 = \forall p_i, p_j \neq p_i \in P, \mathsf{C}(c, p_i, discuss_{p_j} \wedge \overline{giveFloor_c\langle p_j\rangle} \vee discuss_{p_j} \cdot giveFloor_c\langle p_j\rangle, punish_c\langle p_j\rangle)$$

4.1 Simulation of a Possible Enactment

In order to explain the notions of safety and of control, let us suppose that, instead of the commitments in Table 1, the physical event *openDebate* creates the following commitments:

$$\forall p_i \in P,\ C_2(p_i) = \mathsf{C}(c, p_i, \top, askFloor_{p_i} \cdot giveFloor_c\langle p_i\rangle \cdot discuss_{p_i} \cdot cfv_c \vee \overline{askFloor_{p_i}} \cdot cfv_c)$$

Given that P denotes the set of all participants to the assembly, the formula specifies the set of *unconditional* commitments of c to all the participants to the assembly to call for votes (cfv_c) after each participant has either (1) asked for the floor, obtained it, and discussed or (2) declined the possibility to speak.

Figure 2 shows a tree corresponding to some of the possible runs that can be obtained by RONR. Since the RONR events have no preconditions, all interleavings where each event instance occurs at most once are possible. The chair, by executing $openDebate$ commits unconditionally to the regulation $u = askFloor_{p_i} \cdot giveFloor_c$

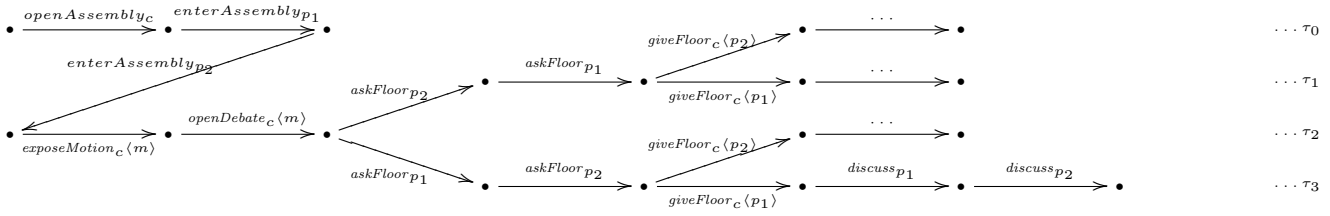


Figure 2: A schematic of some runs with common prefixes for RONR.

$\langle p_i \rangle \cdot discuss_{p_i} \cdot cfv_c \vee \overline{askFloor_{p_i}} \cdot cfv_c$. Let us consider the bottom run, which involves the chair and two participants (p_1 and p_2): participant p_1 asks for the floor, receives it and speaks; instead, p_2 asks for the floor but starts speaking *before* the chair gives it the floor. This causes a violation of the commitment of the chair. This violation is due to p_2, who, however, did not have any commitment to wait for the floor before speaking. Therefore, the chair has no right to expect this behavior by p_2. Indeed, it was not safe for the chair to adopt the above commitment by opening the debate. More formally, we can show that $\tau' \not\models_5 \sigma(c, C_2(p_2))$. Safety holds iff $(\forall \tau'' \in \textbf{cone}(\tau', 5) : (\exists j \geq 5 : \tau'' \models_{[5,j]} \top \Rightarrow \tau'' \models_j \xi(c, u/\tau''_{[5,j]})))$. With $j = 5$ this simplifies to $(\forall \tau'' \in \textbf{cone}(\tau', 5) : \tau'' \models_5 \xi(c, u))$, which does not hold. The participant has not adopted any commitment toward the chair to execute any of the actions under its control. In particular, by repeatedly applying the definition of control, it is easy to see that, in order for the commitment to be safe, it is necessary that after a certain step, for all τ, $\tau \models_i askFloor_{p_2} \cdot giveFloor_c\langle p_2 \rangle \cdot \xi(c, discuss_{p_2} \cdot cfv_c)$. In other words, after $askFloor_{p_2} \cdot giveFloor_c\langle p_2 \rangle$ occurs, c must have control over $discuss_{p_2} \cdot cfv_c$. If there were a commitment of kind $\mathsf{C}(p_2, c, \top, askFloor_{p_2} \cdot giveFloor_c\langle p_2 \rangle \cdot discuss_{p_2})$ at Step 5, $C_2(p_2)$ would be safe because after $askFloor_{p_2} \cdot giveFloor_c\langle p_2 \rangle$ it would have residuated to $\mathsf{C}(p_2, c, \top, discuss_{p_2})$. The fact that no similar commitment is adopted by all agents can lead to the violation described above.

Let us suppose that the commitment adopted after *openDebate* were instead: $\mathsf{C}(c, p_i, askFloor_{p_i} \cdot giveFloor_c\langle p_i \rangle \cdot discuss_{p_i}, askFloor_{p_i} \cdot giveFloor_c\langle p_i \rangle \cdot discuss_{p_i} \cdot cfv_c)$. Can the chair cause the event *openDebate* without worrying about the above commitment? Again, the answer depends on whether the commitment is safe for c. From the definition, it is easy to see that when the antecedent is satisfied and, thus, $askFloor_{p_i} \cdot giveFloor_c\langle p_i \rangle \cdot discuss_{p_i}$ occurs, safety depends on the fact that $\xi(c, cfv_c)$ which is trivially true. The effect of *openDebate* in Table 1 is safe.

The notion of control enables other kinds of reasoning. Let us consider the antecedent of the second commitment that is created as an effect of the action *openAssembly*: $discuss_{p_i} \wedge \overline{giveFloor_c\langle p_i \rangle} \vee discuss_{p_i} \cdot giveFloor_c\langle p_i \rangle$ The effect of the action will be a punishment applied to p_i. Does p_i have control over the set of runs respecting these dependencies, so as to avoid activating the commitment of the chair to punish it? The application of the M_{12} rule allows for destructuring the expression into (a) $discuss_{p_i} \wedge \overline{giveFloor_c\langle p_i \rangle}$ and (b) $discuss_{p_i} \cdot giveFloor_c\langle p_i \rangle$. Condition (a) is a conjunction where $discuss_{p_i}$ is a physical action which is controlled by p_i. The conjunction can be made false by avoiding contributing to a discussion when $\overline{giveFloor_c\langle p_i \rangle}$. Condition (b) is a sequence that is started by a physical action, which is controlled by p_i: this agent can avoid that the condition becomes true by avoiding to contribute to the discussion until c gives it the floor.

5. DISCUSSION

The RONR case study validates some important claims about REGULA. First, it shows that commitments with regulations better help a group of agents coordinate their interactions than traditional propositional commitments would.

In general, because of the autonomy of the agents, no agent x may legitimately expect that another agent y would satisfy a particular regulation. However, the existence of a commitment whose debtor is y and creditor x, and whose consequent is the given regulation precisely specifies and legitimizes such an expectation. The placement of regulations within the antecedents and consequents of commitments helps make the regulations explicit within the system of interacting agents and thereby facilitates their coordination. In particular, autonomous agents can potentially trade off the regulations that they respectively prefer with such trade offs expressed in the commitments they make to one another. Lastly, using events makes regulations, control, safety computationally precise while preserving flexibility of commitments: no event trace is explicitly dictated.

Safety is a means for deciding whether it is reasonable for an agent to adopt a commitment in a given state. We showed that when safety holds for the current state, then there is a possible evolution to another safe state. Further, it is possible to exploit the notion of control in other kinds of reasoning. For instance, as the above example shows, if an agent wants to prevent another agent's commitment from being activated, it can check whether it controls the antecedent of such a commitment.

5.1 Relevant Literature

We now review some previous efforts at enriching commitments with time. Following Searle [15], we can classify norms as either constitutive or regulative. Clearly *openAssembly*, *openDebate*, and so on are *constitutive:* their meaning is defined in terms of commitments using the *means* construct, which amounts to a *counts-as* relation. However, the commitments also have a naturally regulative flavor, which we enhance thanks to the coordination requirements arising from the temporal nature of their content. Thus, in effect, both kinds of norms are grounded in communication in our framework. By contrast, Boella and van der Torre [4] define both kinds of norms in terms of agents' mental states.

Aldewereld et al. [2] use the counts-as relation to operationalize norms. By contrast, we use the counts-as relation to understand physical events in terms of the normative relations they create. Our framework includes significant operational elements. The notion of commitment progression is an operational one. Further, the notion of control may be also viewed as an operational tool for reasoning about norm compliance.

Alberti et al. [1] use events and expectations to model interaction protocols. Expectations help define a temporal relation between events. For example, one can state that if an event occurs at

a certain point in time, another event must occur afterward. Expectations, however, are not scoped by a debtor or creditor nor they are used inside commitments. By contrast, we include temporal regulations inside commitments. This enables precisely identifying who is responsible for each regulation and potentially liable for a violation. Moreover, we define control and safety properties both over regulations and commitments. Using these, an agent can determine whether it would be able to satisfy a (temporal) engagement and to reason about the opportunity of adopting specific commitments.

Baldoni et al. [3] define commitment-based protocols wherein the constitutive and the regulative specifications are decoupled. In particular, they assert regulations as temporal constraints but place them separate from commitments, not within the antecedents and consequents of commitments, as we have done here. By contrast, by including regulations inside commitment, we identify a debtor and a creditor with duties and rights, and propose a notion of control and of safety.

Commitment life cycles, that is progressions, have been variously formalized, especially by Fornara and Colombetti [10], Mallya et al. [13], and El-Menshawy et al. [9]. However, in general, these works neither provide a symbolic characterization of progression as we did above nor do they consider the interplay between control and commitment progression. And, the previous semantic approach work on commitments [17] considers only whether a commitment is active or not, and does not discuss the full life cycle.

Cranefield and Winikoff [7] formalize expectation progression in a linear temporal logic. However, unlike commitments, the expectation modality is not a relation between agents. Such a modality would not be able to support the notions of control and safety as we have formalized here.

Verification of protocols is an important theme. Giordano and Martelli [11] perform two kinds of verification: one, whether an agent's execution is compliant with the protocol, and two, whether the protocol specification itself satisfies some temporal property. Our notion of safety is a third category in that it helps an agent determine whether it has adequate control in order to be able to fulfill its commitments. Safety suggests that the protocol in question is well-designed and the agent's behavior complies with the protocol. We establish compliance at runtime through the notion of the progression of a commitment (Theorem 1).

van der Hoek and Wooldridge [18] reason about the abilities of a coalition of agents given each agent's control over certain variables. Moreover, control may be transferred via what they term a "delegate" operation. Our work embodies similar intuitions: commitments allow control to be passed among agents. Additionally, through the use of commitments, we can support cancel and release as ways to return control and delegate and assign as ways to propagate control.

5.2 Future Directions

The notions of control and of safety that we proposed concern single agents. Along the lines of van der Hoek and Wooldridge [18], a key future direction is to explore notions of *teamwork* and to extend the definitions of control and safety accordingly. It would be worth investigating a richer formal model and language in which we include both states and events as transitions between states. Another interesting question is: given a *specification* in terms of a set of temporal regulations, and knowledge of what events are performed by what agent, can we determine the safe commitments that the agents should adopt so that the resulting computation satisfies the original specification? Such set of commitments could be used to implement agents, interacting by means of commitment-based *protocols* [3, 19].

Acknowledgments

We thank the reviewers for their helpful comments. The Torino team was partially funded by Regione Piemonte, ICT4LAW project. Chopra was supported by a Marie Curie Trentino Fellowship.

6. REFERENCES

[1] M. Alberti, F. Chesani, D. Daolio, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Specification and Verification of Agent Interaction Protocols in a Logic-based System. *Scalable Computing: Pract. & Exp.*, 8(1):1–13, 2007.

[2] H. Aldewereld, S. Álvarez-Napagao, F. Dignum, and J. Vázquez-Salceda. Making norms concrete. *AAMAS*, pp. 807–814, 2010.

[3] M. Baldoni, C. Baroglio, and E. Marengo. Behavior-oriented commitment-based protocols. In *ECAI*, pp. 137–142, 2010.

[4] G. Boella and L. W. N. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *KR Conf.*, pp. 255–266, 2004.

[5] F. Chesani, P. Mello, M. Montali, and P. Torroni. Commitment tracking via the reactive event calculus. In *IJCAI*, pp. 91–96, 2009.

[6] A. K. Chopra and M. P. Singh. Contextualizing commitment protocol. In *AAMAS*, pp. 1345–1352, 2006.

[7] S. Cranefield and M. Winikoff. Verifying social expectations by model checking truncated paths. In *COIN*, *LNCS* 5428, pp. 204–219. Springer, 2009.

[8] N. Desai and M. P. Singh. On the enactability of business protocols. In *AAAI*, pages 1126–1131, July 2008.

[9] M. El-Menshawy, J. Bentahar, and R. Dssouli. Verifiable semantic model for agent interactions using social commitments. In *Proc. Intl. WS Languages, Methodologies, and Development Tools for Multi-Agent Sys.*, *LNCS* 6039, pages 128–152. Springer, 2010.

[10] N. Fornara and M. Colombetti. Operational specification of a commitment-based agent communication language. In *AAMAS*, pp. 535–542, 2002.

[11] L. Giordano, A. Martelli, and C. Schwind. Specifying and verifying interaction protocols in a temporal action logic. *Journal of Applied Logic*, 5(2):214–234, 2007.

[12] A. I. Goldman. *A Theory of Human Action*. Prentice-Hall, Englewood Cliffs, NJ, 1970.

[13] A. U. Mallya, P. Yolum, and M. P. Singh. Resolving commitments among autonomous agents. In *WS on Agent Communication*, *LNAI* 2922, pp. 166–182. Springer, 2003.

[14] H. M. I. Robert, W. J. Evans, D. H. Honemann, and T. J. Balch. *Robert's Rules of Order*, 10th Ed. Da Capo Press, 2000.

[15] J. R. Searle. *The Construction of Social Reality*. Free Press, New York, 1995.

[16] M. P. Singh. Distributed enactment of multiagent workflows: Temporal logic for service composition. In *AAMAS*, 2003.

[17] M. P. Singh. Semantical considerations on dialectical and practical commitments. In *AAAI*, pp. 176–181, 2008.

[18] W. van der Hoek and M. Wooldridge. On the dynamics of delegation, cooperation, and control: a logical account. In *AAMAS*, pp. 701–708, 2005.

[19] P. Yolum and M. P. Singh. Flexible protocol specification and execution: Applying event calculus planning using commitments. In *AAMAS*, pp. 527–534, 2002.

Specifying and Applying Commitment-Based Business Patterns

Amit K. Chopra
University of Trento
Via Sommarive, 14 I-38123 Povo, Italy
chopra@disi.unitn.it

Munindar P. Singh
North Carolina State University
Raleigh, NC 27695-8206
singh@ncsu.edu

ABSTRACT

Recent work in communications and business modeling emphasizes a commitment-based view of interaction. By abstracting away from implementation-level details, commitments can potentially enhance perspicuity during modeling and flexibility during enactment.

We address the problem of creating commitment-based specifications that directly capture business requirements, yet apply in distributed settings. We encode important business patterns in terms of commitments and group them into *methods* to better capture business requirements.

Our approach yields significant advantages over existing approaches: our patterns (1) respect agent autonomy; (2) capture business intuitions faithfully; and (3) can be enacted in real-life, distributed settings. We evaluate our contributions using the Extended Contract Net Protocol.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Design, Theory

Keywords

Protocols, Software engineering, Method engineering

1. INTRODUCTION

Commitment-based approaches to agent communication are finding broad traction in specifying interaction protocols. What makes commitments an appealing abstraction is that they naturally capture the business relationships that arise in our everyday life and business interactions, and offer flexibility in realizing them.

The expression C(debtor, creditor, antecedent, consequent) represents a commitment: it means that the debtor is committed to the creditor for ensuring the consequent if the antecedent holds. For example, C(buyer, seller, goods, paid) means that the buyer commits to the seller that if the seller provides the goods the buyer will ensure he is paid. Whereas it is easy enough to come up with commitments, it is not easy to specify the *right* commitments for particular applications. For instance, Desai et al. [4] show how a scenario dealing with foreign exchange transactions may be formalized in multiple ways using commitments, each with different ramifications on the outcomes. This leads us to the main question we address: *How can we guide software engineers in creating appropriate commitment-based specifications?*

Such guidance is often available for operational approaches such as state machines and Petri nets that describe interactions in terms of message order and occurrence. For instance, Figure 1 shows two common patterns expressed as (partial) state machines, which can aid software engineers in specifying operational interactions. Here, b and s are buyer and seller, respectively. (A) says that the seller may accept or reject an order; (B) says the buyer may confirm an order after the seller accepts it.

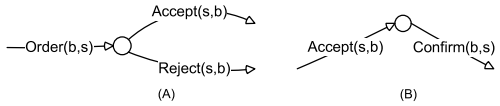


Figure 1: Example operational patterns.

By contrast, commitment protocols abstract away from operational details, focusing on the meanings of messages, not their flow. Clearly, operational patterns such as the above would not apply to the design of commitment protocols. What kinds of patterns would help in the design of commitment protocols? By and large, they would need to be business patterns—characterizing requirements, not operations—that emphasize meanings in terms of commitments. In contrast with Figure 1, business patterns—as we formalize them—describe what it *means* to make, accept, reject, or update an offer, not when to send messages.

We apply our patterns towards creating commitment-based specifications in a manner inspired by situational method engineering (SME) [10]. In SME, a method corresponds to a particular software engineering lifecycle and is composed of reusable fragments selected based on application and organizational requirements. For example, based on its requirements, a development organization may adopt goal-based or scenario-based requirements engineering or omit requirements engineering altogether. Analogously, for us, those developing commitment-based specifications would choose a commitment-based method that composes specific business patterns and that suits their requirements, including those relating to the organizational context [12] in which the sys-

Cite as: Specifying and Applying Commitment-Based Business Patterns, Amit K. Chopra and Munindar P. Singh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 475-482.

tem to be will be enacted. In this sense, a method describes a second-order business pattern.

Contributions. Our contributions are as follows. First, we identify business patterns as distinct from semantic and enactment patterns. Whereas semantic patterns encapsulate general commitment reasoning [2] and enactment patterns guide a commitment-based agent design, business patterns support specifying business protocols in cross-organizational settings. Second, we identify semantic antipatterns, which generally reflect a closed system way of thinking and are not suitable for open settings. Third, we identify several business patterns that accommodate common business situations. Fourth, we formulate engineering *methods* as sets of selected patterns and outline a simple approach based on organizational requirements for selecting among methods.

Like any set of patterns, the patterns in this paper reflect intuitions rooted in experience. Our patterns, however, are also motivated by the following requirements.

Autonomy-compatibility Autonomy broadly refers to the lack of control: no agent has control over another agent. To get things done, agents set up the appropriate commitments by interacting. Any expectation from an agent beyond what the agent has explicitly committed to is unreasonable.

Explicit meanings Our patterns make public the aspects of meaning that ought to have been public in the first place, but are often hidden within agent implementations. For example, updating a standing offer would mean replacing an existing commitment with a new one. An operational approach would simply allow for multiple *UpdateOffer* messages. If agents differently assume whether the latest message prevails, misalignment would ensue.

Distributed enactment Our business patterns build up systematically from a core set of communication primitives reflecting established ways to manipulate commitments in distributed settings.

Result. We evaluate our approach via a case study. The main result we obtain is that our approach highlights the critical design decisions and places them at a business level. First, a designer can see what is at stake in those decisions and can choose according to the needs of the business partners and the contextual setting in which they will interact. Second, through its focus on formalizing business meaning, our approach captures exactly what the business needs. In contrast, traditional approaches are guilty of over-specifying on some aspects (leading to rigid enactments) and under-specifying others (leading to potential ambiguity in realistic environments). Their only recourse against the former is to enumerate additional enactments and their only recourse against the latter is to insert additional ad hoc constraints, thus leaning toward over-specification. The overall outcome is excessive complexity.

Organization. The rest of the paper is organized as follows. Section 2 describes the necessary background for computing commitments in distributed settings. It also discusses semantic patterns. Section 3 introduces some business patterns, enactment patterns, and semantic antipatterns. Section 4 applies the patterns toward protocol specification via methods. Section 5 applies our approach to the Extended Contract Net Protocol [15]. Section 6 sums up our approach along with a discussion of the relevant literature.

2. BACKGROUND

We adopt Chopra and Singh's formal framework [2], including their language and reasoning postulates. Table 1 repeats their grammar for commitments and for messages that manipulate commitments. A sender can inform a receiver using a Declare. A commitment is detached when its antecedent becomes true ($\top$), meaning its debtor is unconditionally committed. A commitment is discharged when its consequent becomes true. Table 2 lists some important kinds of commitments that may arise in a fan-selling scenario.

Table 1: Syntax for commitments and messages.

Commitment $\longrightarrow$ C(Agent, Agent, DNF, CNF)
Content $\longrightarrow$ Atom | $\neg$ Atom | Stative(Agent, Agent, DNF, CNF)
Stative $\longrightarrow$ created | released | canceled | violated
DNF $\longrightarrow$ And | And $\vee$ DNF
CNF $\longrightarrow$ Or | Or $\wedge$ CNF
And $\longrightarrow$ Content | Content $\wedge$ And
Or $\longrightarrow$ Content | Content $\vee$ Or
Message $\longrightarrow$ Declare(Agent, Agent, News)
Message $\longrightarrow$ Op(Agent, Agent, DNF, CNF)
News $\longrightarrow$ Atom | Stative(Agent, Agent, DNF, CNF) | Atom $\wedge$ News | Stative(Agent, Agent, DNF, CNF) $\wedge$ News
Op $\longrightarrow$ Create | Cancel | Release | Delegate

Table 2: Commitments in the syntax of Table 1.

Name	**Commitment** (S is seller; B is buyer)
c_A	C(S, B, paid, fan): S commits to B that if payment is made, the fan will be delivered.
c_{UA}	C(S, B, $\top$, fan): The unconditional version of c_A. S commits to B that the fan will be delivered.
c_B	C(S, B, released(S, B, $\top$, fan), created(S, B, $\top$, discount)): S commits to B that if B releases S from the commitment to deliver the fan, S will give B a discount on its next purchase.
c_C	C(S, B, $\neg$fan$\wedge$released(S, B, $\top$, fan), created(S, B, $\top$, discount)): Similar to c_B except that it accounts for the case when S's delivery of the fan and B's release cross in transit—in such a case, S need not give the discount anymore.

The *statives* (except violated) record the history of commitment operations. For example, created(x, y, r, u) is added to an agent's KB when an agent has observed the message Create(x, y, r, u), and so on. We introduce violated to capture that an unconditional commitment has been violated, e.g., because the deadline for bringing about its consequent has passed. Table 3 lists each message along with its sender and receiver, and the effects of the messages (we omit the assignment operation for brevity).

Chopra and Singh's framework uses two kinds of postulates: update postulates (appropriately constrained by the

Table 3: Core messages pertaining to commitments.

Message	Sender	Receiver	Effect
$\mathsf{Create}(x, y, r, u)$	x	y	$\mathsf{C}(x, y, r, u)$
$\mathsf{Cancel}(x, y, r, u)$	x	y	$\neg\mathsf{C}(x, y, r, u)$
$\mathsf{Release}(x, y, r, u)$	y	x	$\neg\mathsf{C}(x, y, r, u)$
$\mathsf{Delegate}(x, y, z, r, u)$	x	z	$\mathsf{C}(z, y, r, u)$
$\mathsf{Declare}(x, y, p)$	x	y	p

conditions listed below) that capture the computation of an agent's state following its observation of a message, and commitment reasoning postulates such as (assuming the same debtor-creditor pair throughout a postulate) $u \rightarrow \neg\mathsf{C}(r, u)$ (captures discharge) and $\mathsf{C}(r \wedge s, u) \wedge s \rightarrow \mathsf{C}(r, u)$ (captures detach), and so on. These postulates encode *semantic patterns*, that is, the domain-independent rules of computing commitments. In Table 3, the effects are *nominal* because they hold only under the following conditions. (A commitment $\mathsf{C}(r, u)$ is stronger than $\mathsf{C}(r', u')$ iff $u \vdash u'$ and $v' \vdash v$.)

Novel Creation $\mathsf{Create}(r, u)$ is a noop if a stronger commitment $\mathsf{C}(s, v)$ holds or has held before (that is, if $\mathsf{created}(\mathsf{s}, \mathsf{v})$ holds).

Complete Erasure $\mathsf{Release}(r, u)$ or $\mathsf{Cancel}(r, u)$ removes all commitments weaker than $\mathsf{C}(r, u)$ provided no $\mathsf{C}(s, v)$ strictly stronger than $\mathsf{C}(r, u)$ holds; otherwise it is a noop.

Accommodation From $\mathsf{Release}(r, u)$ and $\mathsf{Cancel}(r, u)$, infer that each weaker $\mathsf{C}(s, v)$ has held before.

Notification Whenever a creditor learns of a condition that features in the antecedent, it notifies the debtor, and whenever a debtor learns of a condition that features in the consequent, it notifies the creditor.

Priority If two agents may take conflicting actions, the protocol specifies ahead of time whose action has priority.

The principal result that follows from the above conditions is that even when agents communicate asynchronously, they would remain aligned with respect to their commitments (assuming reliable in-order message delivery for every pair of agents—easily supported by common infrastructure such as reliable message queues).

3. COMMITMENT PATTERNS

We discuss three kinds of patterns. *Business patterns* capture the meanings of business communications in terms of commitments, *enactment patterns* specify *when* an agent may enact a particular business communication, and *semantic antipatterns* capture inappropriate patterns. All of our examples are from the fan-selling domain (Table 2).

3.1 Business Patterns

Business patterns encode the common ways in which businesses engage each other. By representing business patterns using Chopra and Singh's framework, we can guarantee alignment even for asynchronous enactments.

The messages of Table 3 correspond to elementary business patterns. Here, $\mathsf{Offer}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ means $\mathsf{Create}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ (the Generic Offer or GO pattern); $\mathsf{CancelOffer}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ means $\mathsf{Cancel}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ (the Cancel Offer or CO pattern); and $\mathsf{RejectOffer}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ means $\mathsf{Release}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ (the Release Offer or RO pattern). However, we can build upon the basic primitives to build more complex business patterns such as for updating, compensation, mutual commitment, and so on. Below, we list some recurring business patterns.

• Basic Offer (BO)
Intent To set up a basic business transaction.
Motivation Captures a basic way of doing business.
Implementation $\mathsf{BasicOffer}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ means $\mathsf{Create}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ where r and u are formulas over atoms (they contain no statives).
Example $\mathsf{BasicOffer}(\mathsf{S}, \mathsf{B}, \mathsf{paid}, \mathsf{fan})$
Consequences For progress, the creditor should be ready to bring about the antecedent.

• Nested Offer (NO)
Intent The debtor wants a commitment from the creditor for something in return for something else.
Motivation To set up a richer (both parties are committed) and more flexible engagement.
Implementation $\mathsf{NestedOffer}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ means $\mathsf{Create}(\mathsf{x}, \mathsf{y}, \mathsf{created}(\mathsf{y}, \mathsf{x}, \top, \mathsf{r}), \mathsf{u})$.
Example $\mathsf{NestedOffer}(\mathsf{S}, \mathsf{B}, \mathsf{paid}, \mathsf{fan})$
Consequences When the antecedent holds, both x and y are unconditionally committed to u and r, respectively. When that happens, each would gain some measure of safety in acting first and discharging its commitment, thus improving flexibility in enactment.

• Mutual Commitment Offer (MCO)
Intent Debtor should have the exact "reciprocal" commitment from the creditor: if the creditor commits to u for r, the debtor commits to r for u.
Motivation To set up a richer and more flexible engagement, wherein both parties are committed.
Implementation $\mathsf{MutualCommitmentOffer}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u})$ means $\mathsf{Create}(\mathsf{x}, \mathsf{y}, \mathsf{created}(\mathsf{y}, \mathsf{x}, \mathsf{u}, \mathsf{r}), \mathsf{created}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u}))$
Example $\mathsf{MutualCommitmentOffer}(\mathsf{S}, \mathsf{B}, \mathsf{paid}, \mathsf{fan})$
Consequences This pattern is less prone to violations than Nested Offer, as only one party could possibly violate its commitment.

• Business Transaction Identifiers (BTI)
Intent To enable an agent to distinguish distinct offers and to relate commitments that coherently fall into the same business transaction.
Motivation It is important (1) not to conflate distinct business transactions, so that commitments from different transactions do not interfere with each other, and (2) preserve logical structure so the reasoning is sound.
Implementation Introduce identifiers in the antecedent, propagating them as needed to the consequent.
Example Writing the identifier as the first parameter of a proposition, $\mathsf{C}(\mathsf{S}, \mathsf{B}, \mathsf{paid}(0), \mathsf{fan}(0))$ occurs in a different transaction from $\mathsf{C}(\mathsf{S}, \mathsf{B}, \mathsf{paid}(1), \mathsf{fan}(1))$.
Consequences We need a clear information model to make sure the commitments pertaining to one transaction do not involve the identifiers of another.

• Compensation (COM)
Intent To compensate the creditor in case of commitment cancellation or violation by the debtor.
Motivation It is not known in advance whether a party will fulfill its commitments; compensation commitments provides some assurance to the creditor in case of violations.
Implementation $\mathsf{Compensate}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u}, \mathsf{p})$ means $\mathsf{Create}(\mathsf{x}, \mathsf{y}, \mathsf{violated}(\mathsf{x}, \mathsf{y}, \mathsf{r}, \mathsf{u}), \mathsf{p})$.

Example Compensate(S, B, paid, fan, discount)
Consequences A commitment (even a compensation commitment) should ideally be supported by compensation; however, at some level, the only recourse is escalation to the surrounding business *context*—for example, the local jurisdiction [12].

• Update (UP)
Intent To update a previously made offer.
Motivation Changing business environments may require debtors to update their commitments.
Implementation Update(x, y, r, u, s, v) means Cancel(x, y, r, u) and Create(x, y, s, v).
Example Update(S, B, paid$12, fan, paid$15, fan)
Consequences One must be careful in applying updates since the creditor may not find the new commitment an acceptable substitute for the old commitment.

• Release Incentive (RI)
Intent To enable the debtor to offer an incentive to the creditor for releasing it from a commitment.
Motivation Due to changing business environments, it may be more profitable for the debtor to offer an incentive to the creditor for releasing it from an existing commitment.
Implementation ReleaseIncentive(x, y, r, u, p) means Create(x, y, ¬u ∧ released(x, y, r, u), p) where p represents the incentive. The conjunction in the antecedent is necessary to handle the case where Declare(x, y, u) may cross with Release(x, y, r, u): once u occurs, the debtor is off the hook.
Example ReleaseIncentive(S, B, ⊤, fan, discount)
Consequences The creditor may not take up the incentive offer; the debtor may then consider canceling the commitment unilaterally.

• Delegation Acceptance (DA)
Intent To set up the proper relationship between a delegator and delegatee for effective delegations.
Motivation The debtor may delegate (viewed as a request) a commitment to another party if it sees value in it; however, the delegatee is not bound to accept the delegation.
Implementation DelegationAcceptance(x, y, z, r, u) means Create(z, x, delegated(x, y, z, r, u), created(z, y, r, u)); delegated(x, y, z, r, u) captures the performance of the delegation request.
Example DelegationAcceptance(S, B, S_2, paid, fan)
Consequences The parties should set up additional notifications, for example, when the delegatee has discharged the commitment, for greater confidence.

• Redundancy (RED)
Intent To mitigate risk by assuring the creditor of service by a backup agent in case things go awry.
Motivation A debtor can reduce the risk of violating its commitments by introducing a backup.
Implementation Redundancy(x, y, z, r, u) means Create(x, y, risk(x, y, r, u), created(z, y, r, u)) (x is promising backup service by z to y). Her, risk(x, y, r, u) is a domain-specific predicate that holds when a commitment is at risk of being violated.
Consequences This pattern presumes the backup agent commits to *accepting* delegations from the debtor, for example via Delegate Acceptance.

In the end, all of the above patterns are specializations of the Generic Offer pattern, except Update which is a composite pattern, and yet we are able to capture a rich set of business patterns by appropriately changing the content of the commitments.

3.2 Enactment Patterns

Whereas a business pattern describes the meaning of communication, an enactment patterns describe the conditions under which an agent should enact a business pattern, that is, *when* to undertake the corresponding communication. In general, enactment is agent-specific. Nonetheless, some behaviors are commonly observed in practice, for example, in negotiation. A locus of such enactments may serve as the basic agent skeleton. We highlight two enactment patterns that are built upon the offer business patterns presented earlier.

• Improved Offer
Intent To make improved offers via stronger commitments.
Motivation The creditor has not taken up an earlier offer.
When x makes the offer C(x, y, r, u); y has not taken up the offer, that is, r does not hold. Then, x makes a *stronger* offer C(x, y, r', u') (recall strength from Section 2) in order to entice y into the deal.
Consequences The debtor is committed more strongly; ideally, it must make sure the stronger commitment has at least some positive utility, even if diminished. This pattern represents a concession.

• Counter Offer
Intent One party makes an offer to another, who responds with a modified offer of its own.
Motivation Essential for negotiation.
When Let C(x, y, r, u) be the commitment corresponding to the original offer. Making a counteroffer would amount to creating the commitment C(y, x, u', r') such that $u' \vdash u$ and $r \vdash r'$, in other words, if the consequent is strengthened and the antecedent is weakened. An alternative implementation includes doing Release(x, y, r, u) in addition.
Consequences When $u \equiv u'$ and $r \equiv r'$, the counter offer amounts to a mutual commitment.

3.3 Semantic Antipatterns

Below, we enhance Chopra and Singh's framework with *semantic antipatterns*—forms of representation and reasoning to be avoided because they conflict with the autonomy of the participants or with a logical basis for commitments.

• Commit Another as Debtor
Intent An agent creates a commitment in which the debtor is another agent.
Motivation To capture delegation, especially in situations where the delegator is in a position of power of over the delegatee.
Implementation The sender of Create(y, z, p, q) is x, thus contravening Table 3.
Example Consider two sellers S and S_2. S sends Create(S_2, B, paid, fan) to B.
Consequences A commitment represents a public undertaking by the debtor. A special case is when $x = z$. That is, x unilaterally makes itself the creditor.
Criteria Failed S_2's autonomy is not respected.

Alternative Apply delegation to achieve the desired business relationship, based on prior commitments. In the above example, S_2 could have a standing commitment with S to accept delegations. S can then send a delegate instruction to S_2 upon which S_2 commits to B. See the DELEGATION ACCEPTANCE and REDUNDANCY business patterns in Section 3.1.

• ACKED COMMIT

Intent A commitment may hold only when the creditor has acknowledged its creation to the debtor. That is, the creditor should accept the commitment [9].

Motivation A commitment should be set up only upon the agreement of both parties. This is often based on a misunderstanding of commitments: that the creditor is committed to the antecedent.

Implementation Creditor acknowledges a create message.

Example The seller S enacts $\mathsf{BasicOffer}(\mathsf{S}, \mathsf{B}, \mathsf{paid}, \mathsf{fan})$; however, the offer does not hold until the buyer B acknowledges the offer.

Consequences It rules out unilateral commitment by the debtor such as in a business offer or advertisement for services.

Criteria Failed Autonomy (a debtor shouldn't need a creditor's approval to create a commitment) and generality (as it is unable to capture common scenarios).

Alternative MUTUAL COMMITMENT OFFER.

• COMMITMENT IDENTIFIERS

Intent Gives a unique identifier to every commitment.

Motivation To distinguish transactions and to simplify reasoning about commitments in concurrent settings, e.g., in [5, 11].

Implementation Every commitment has an ID, as in $\mathsf{C}(id, \mathsf{debtor}, \mathsf{creditor}, \mathsf{antecedent}, \mathsf{consequent})$.

Example $\mathsf{C}(id_0, \mathsf{S}, \mathsf{B}, \mathsf{paid}, \mathsf{fan})$ and $\mathsf{C}(id_1, \mathsf{S}, \mathsf{B}, \mathsf{paid}, \mathsf{fan})$

Consequences Reasoning about commitments breaks down. For example, from $\mathsf{C}(x, y, r, u) \wedge \mathsf{C}(x, y, r, v)$, one infers $\mathsf{C}(x, y, r, u \wedge v)$. However, one cannot apply such an inference to $\mathsf{C}(id_0, x, y, r, u) \wedge \mathsf{C}(id_1, x, y, r, v)$. Further, commitment operations must now explicitly refer to the identifiers in addition to the logical content.

Criteria failed Generality, since general reasoning about commitments breaks down.

Alternative BUSINESS TRANSACTION IDENTIFIER.

4. PROTOCOL SPECIFICATION

We explain how the business patterns specified above may be used by protocol designers.

Business protocols are often specified around a central exchange of goods, services, or monies. Although a simple pattern such as BASIC OFFER is usually enough to capture the exchange, typically participants want the protocols to be robust in the following ways. Table 4 summarizes how our business patterns support the robustness requirements.

Creditor Confidence (CC) Inspire confidence in the creditor about the outcome of the interaction: e.g., COMPENSATION and REDUNDANCY.

Debtor Confidence (DC) Inspire confidence in the debtor by requiring commitments from other parties: e.g., NESTED OFFER, MUTUAL COMMITMENT OFFER, and DELEGATION ACCEPTANCE.

Progress (P) Ensure liveness by requiring the involved parties to act or risk being out of compliance, e.g., NESTED OFFER and COMPENSATION (once a violation happens).

Mitigation (M) Mitigate risk for the debtor of a commitment by helping it avoid noncompliance, e.g., RELEASE INCENTIVE and DELEGATION ACCEPTANCE.

Table 4: Business patterns and robustness.

	CC	DC	P	M
NO	–	Yes	Yes	–
MCO	–	Yes	Yes	–
COM	Yes	–	Yes	–
UP	–	–	–	Yes
RI	Yes	–	–	Yes
DA	Yes	Yes	–	–
RED	Yes	–	–	Yes

A bundle of business patterns is a reusable *method* for addressing certain requirements. For example, the method $\langle \text{MCO}, \text{COM} \rangle$ addresses the requirements of creditor and debtor confidence; $\langle \text{MCO}, \text{COM}, \text{DA} \rangle$ does the same job better; $\langle \text{MCO}, \text{COM}, \text{DA}, \text{RED} \rangle$ fares even better. Alternatively, a protocol designer could choose the method $\langle \text{NO}, \text{RI} \rangle$ in order to support progress as well as mitigation. In essence, the patterns can be grouped according to the required level of robustness.

However, in selecting a method, a protocol designer would take into account not only the robustness requirements, but also the intended organizational setting. The resources of the organization and its policies would affect the method selected. For example, a fan seller ModernFans might not want to use delegation as a mitigation strategy for competitive reasons. It might also want to makes offers which its customers may take advantage of directly by making payments; in such a case, ModernFans would select a method that includes BO instead of NO or MCO. Further, the more robust a method the more computational resources the agents would need to devote during enactments—another reason a less robust method may be selected. In general, a protocol designer must make judgments about robustness versus organizational policies and resource usage.

5. CASE STUDY

We now apply the patterns to the Extended Contract Net Protocol (xCNP) formalized by Vokřínek et al. [15]. xCNP involves two roles: contractor and contractee. Vokřínek et al.'s extensions enable the negotiation of penalties in case one of the parties is unable to fulfill its end of the bargain. The xCNP protocol has three distinct phases: contract formation (similar to the traditional CNP), contract decommitment (negotiation of penalties in case one of the parties wants out, that is, before the actual violation of the contract), and contract resolution (negotiation of penalties in case of an actual violation).

Vokřínek et al. formalize xCNP in a procedural manner via a state machine (Figure 2). Many enactments are possible. For example, a contract may be reached or a penalty may be successfully negotiated; the parties could negotiate back and forth many times before reaching an agreement; they could fail to arrive at an initial contract; one of them could propose decommitment and then take back the proposal, and so on.

Table 5: Commitments used to model an xCNP-like setting.

Label	D	C	Antecedent	Consequent
pr	cte	ctr	created(ctr, cte, built(0), paid(0))	created(cte, ctr, paid(0), built(0))
co	ctr	cte	created(cte, ctr, paid(0), built(0) ∧furnished(0))	created(ctr, cte, built(0) ∧ furnished(0), paid(0))
cv	cte	ctr	created(ctr, cte, built(0) ∧ furnished(0), paid(0)) ∧ created(ctr, cte, violated(ctr, cte, ⊤, paid(0)), penalty(0))	created(cte, ctr, paid(0), built(0) ∧ furnished(0))
cvs	cte	ctr	created(ctr, cte, built(0) ∧ furnished(0) ∧driveway(0), paid(0)) ∧ created(ctr, cte, violated(ctr, cte, ⊤, paid(0)), penalty(0))	created(cte, ctr, paid(0), built(0) ∧ furnished(0)∧ driveway(0))
py	ctr	cte	built(0) ∧ furnished(0) ∧ driveway(0)	paid(0)
vio	ctr	cte	violated(ctr, cte, ⊤, paid(0))	penalty(0)
ta	cte	ctr	paid(0)	built(0) ∧ furnished(0) ∧ driveway(0)
in	ctr	cte	¬paid(0) ∧ released(ctr, cte, ⊤, paid(0))	released(cte, ctr, ⊤, built(0) ∧ furnished(0)∧ driveway(0)) ∧ expensesPlusTen(0)
in_R	ctr	cte	⊤	expensesPlusTen(0)

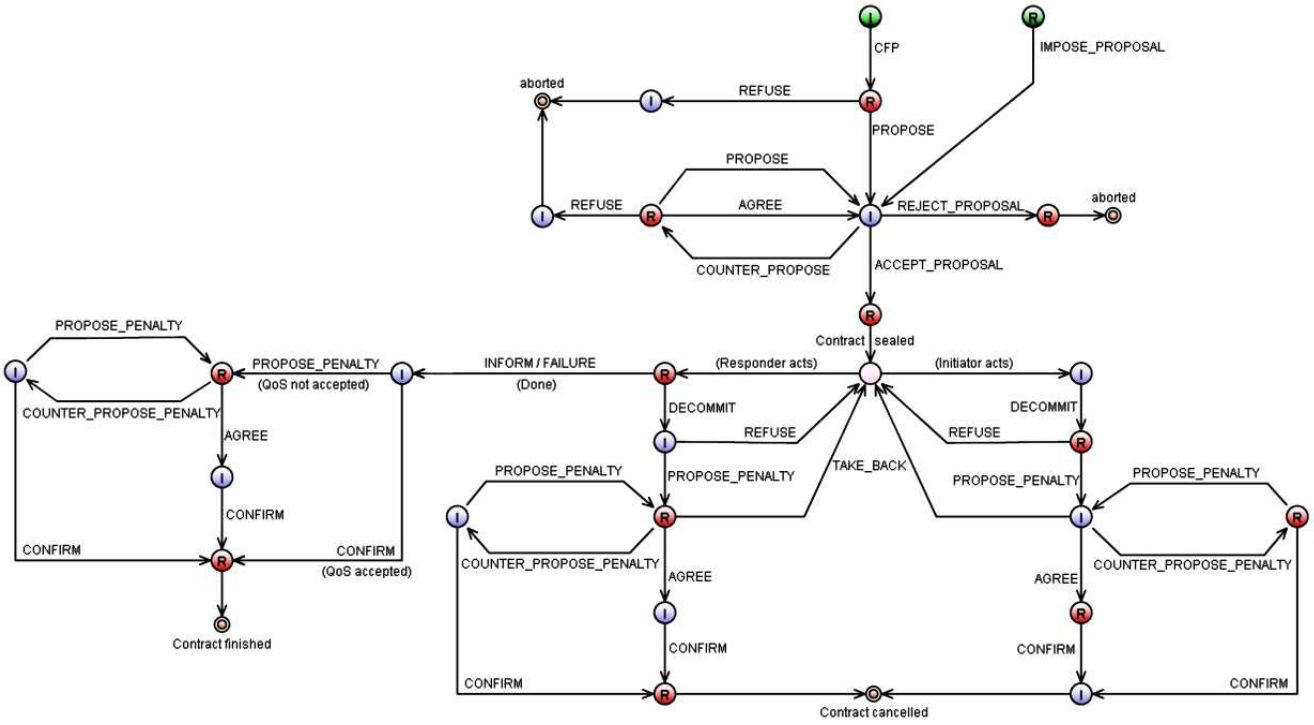


Figure 2: The xCNP protocol [15]. I and R refer to the contractor (ctr) and the contractee (cte), respectively.

5.1 Applying our Approach

We replace the operational model of xCNP with a model based on the appropriate business patterns.

- xCNP emphasizes the synchronizing agree-confirm operational pattern for arriving at any outcome: in contract formation (after one party agrees to a proposal, the other must confirm it), in penalty negotiation, and so on. We instead use Mutual Commitment Offer (MCO) or Nested Offer (NO).
- In order to enable parties to get out of their commitment, xCNP supports decommitment. In our framework, Release Incentive (RI) captures decommitment: the commitment is not yet violated, and the debtor is asking to be released by the creditor in return for a penalty (incentive from the creditor's point of view). An alternative set of patterns for implementing decommitment consists of Cancel Offer (CO) and Compensation (COM, as proposing a penalty for the cancellation).
- xCNP supports penalties for violation to capture contract resolution. We can instead use Compensation.

Thus, to capture a contract protocol, one can choose from the following business pattern methods.

Method 1. ⟨NO, RI, COM⟩
Method 2. ⟨MCO, RI, COM⟩
Method 3. ⟨NO, CO, COM⟩
Method 4. ⟨MCO, CO, COM⟩

As stated earlier, the choice of the method depends not

only upon the robustness criteria but also upon organizational requirements. For example, Method 2 is more robust than Method 4 because cancellation is in effect a violation. However, a business partner could still choose Method 4 if it did not care about violations as much as it cared about immediacy (in the sense that it does not have to *wait* to be released by the other party).

The four methods above are just samples; in general, designers could come up with more patterns and methods that meet various requirements.

5.2 Enactment

Table 5 lists the commitments used in xCNP; $name_U$ is the unconditional commitment resulting from *name*. Figure 3 shows an enactment of the contract formation stage using Method 2. The scenario is one where a contractor issues a CFP for an office block construction. Let's consider Figure 3 step by step.

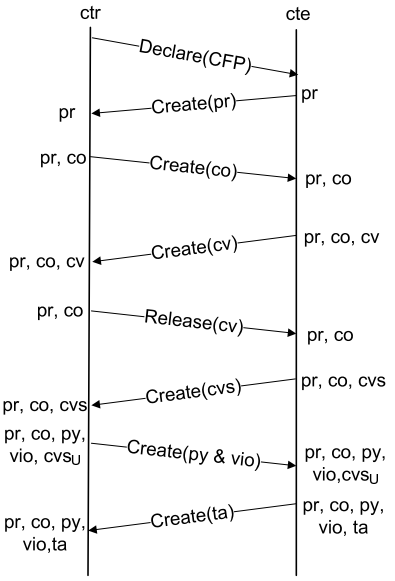


Figure 3: Method 2: Contract formation enactment.

1. Contractor ctr applies the DECLARE pattern in sending the CFP.
2. Contractee cte enacts the MCO pattern in response (to create *pr*): essentially the contractee will do built in return for paid.
3. ctr does a COUNTER OFFER in response (to create *co*): in addition to built, the contractor also wants furnished.
4. cte does a COUNTER OFFER in response (to create *cv*): the contractee is ready to do built and furnished for paid, but wants contractor ctr to commit to paying a penalty in case ctr cannot pay for the services rendered.
5. ctr applies REJECT OFFER in response.
6. cte then applies IMPROVED OFFER: it sweetens the offer by throwing in driveway.
7. ctr then creates the necessary commitments using the BASIC OFFER pattern; presumably the contractor is happy with the improved offer.
8. cte also creates the necessary commitments.

Figure 4 shows an enactment of the contract decommitment stage using Method 2. The figure begins from where the interaction has progressed so the commitments *py* and *ta* hold. Let's look at Figure 4 step by step.

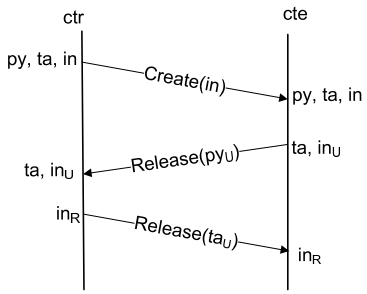


Figure 4: Method 2: Decommitment enactment.

- ctr applies the RELEASE INCENTIVE pattern (to create *in*): if cte releases it from py_U and payment has not yet happened, ctr will release cte from ta_U and reimburse cte to the extent of 110% of the expenses cte has already incurred.
- In response, cte releases it from py_U.
- In response, ctr releases cte from ta_U. At this point, in_R holds: ctr must still pay cte 110% of the expenses.

5.3 Observations and Conclusions

xCNP, as formalized by Vokřínek et al., does not consider the meanings of interactions. For example, it does not formalize what it means to decommit. By contrast, we formalized decommitment via two alternative patterns that have different ramifications for meeting organizational requirements. In addition, one of the alternatives turned out to be a composite pattern (CANCEL OFFER and COMPENSATION). Operational formalizations miss out on such nuances.

We showed four alternative methods that model the enhancements xCNP claims over the traditional CNP. We gave an example requirement of what might drive a protocol designer to choose one method over another. Since our methods are meaning-based, it is natural for a designer to select among them than from among the same number of alternative operational formalizations of xCNP.

All our methods draw from the patterns we introduced earlier, which themselves draw from the basic framework in [2]. Thus business-level interoperability is guaranteed even when agents enact the patterns asynchronously. By contrast, Vokřínek et al.'s xCNP formalization is both over-specified and under-specified. It is over-specified because it is highly synchronous and enforces rigid enactments, such as via the agree-confirm pattern for arriving at any outcome. It is under-specified in net effect because it cannot capture enactments that would be natural. For example, both the contractor and contractee (I and R, respectively in Figure 2) may act concurrently by sending CFP and IMPOSE_PROPOSAL, respectively. Although these transitions are allowed, the resulting state is not captured in the formalization. A similar situation ensues when, after sealing the contract, both parties act concurrently in order to decommit. In general, it is difficult to capture all possible executions paths via operational methods because of their lower level of abstraction.

One could try to repair Vokřínek et al.'s formalization by inserting additional enactment paths to address its rigidity and insert additional synchronizations to address messages crossing in transit (the latter would increase rigidity).

However, such a formalization would be overly complex, unwieldy to maintain, and difficult both for designers and end users to understand.

In conclusion, when the business meanings of interactions are made explicit, (1) designers gain in flexibility in selecting from a range of possible specifications, that is, the methods, and (2) agents gain in flexibility in enacting the specified system because they can reason about meanings and select among alternative courses of action.

6. DISCUSSION

Method engineering is an expanding area of SE. Traditionally, method engineering considers how to engineer and choose among methods in the large, such as Agile or Scrum [10], with selection based on the structure of the given software development organization. In contrast with existing work, we observe that a (modeling) method could be understood in terms of the families of interactions that we wish to support among agents, such as the partners in business processes. We too are concerned with organizations, but emphasize the organization of the business partners during enactment as well as the contextual organization in which the business process takes place. We envisage that suitable methods would be engineered based on features of such organizations as well as the flexibility supported by the business partners' agents. And designers who apply selected methods would create models of interaction that naturally meet those criteria.

Our approach to patterns is layered: method over business over semantic patterns. Lind and Goldkuhl [7] propose a layered approach to business modeling starting with business actions and building up to transactions; however, they overlook the meanings of the business actions themselves.

Conceptually any protocol, no matter how specified, is a reusable pattern of interaction. Existing approaches for protocol composition, e.g., [8, 14], focus on procedural aspects, which though valuable cannot substitute for business meanings. Singh et al. [12] motivate some commitment-based connector patterns, including multiparty ones. However, they do not consider the challenges of distributed enactment.

Traditionally, researchers have used action logics for commitment protocol specification, for example, as in [6]. Chopra and Singh [1] support the application of business patterns, such as for *Return and Refund*, to protocols specified in an action logic. However, these approaches assume synchronous communication and, further, freely mix meaning axioms along with operational constraints such as for message ordering. By contrast, our patterns are purely meaning-based, and they can be enacted asynchronously.

Wang et al. [16] annotate each commitment with types depending on the relative order in which its antecedent and consequent ought to be satisfied. For example, they annotate $\mathsf{C}(\mathsf{merchant}, \mathsf{customer}, \mathsf{payment}, \mathsf{refund})$ *strictly-ordered*: payment must be made before the refund can be made. However, *payment-before-refund* can be an enactment policy—a choice—on the part of the merchant; the ordering is not necessarily an issue of commitment specification. Annotating commitments as Wang et al. do unduly limits flexibility during enactment. In general, it is important to sort out the issues of agent specification from those of protocol specification [3].

Future directions include coming up a rich taxonomy of requirements that pertain to interactions, and providing tool-based support to designers for picking from among the methods in a repository.

Telang and Singh [13] propose a metamodel in which to express cross-organizational business models that includes a set of modeling patterns. They formalize commitments in a simplified temporal semantics assuming synchrony and show how to verify low-level protocols expressed in sequence diagrams with respect to the business models. It would be interesting to reconcile our approach with theirs.

Acknowledgments

We thank the reviewers for their helpful comments. Chopra was supported by a Marie Curie Fellowship.

7. REFERENCES

[1] A. K. Chopra, M. P. Singh. Contextualizing commitment protocols. *AAMAS*, pp. 1345–1352, 2006.

[2] A. K. Chopra, M. P. Singh. Multiagent commitment alignment. *AAMAS*, pp. 937–944, 2009.

[3] A. K. Chopra and F. Dalpiaz and P. Giorgini and J. Mylopoulos. Reasoning about agents and protocols via goals and commitments. *AAMAS*, pp. 457–464, 2010.

[4] N. Desai, A. K. Chopra, M. Arrott, B. Specht, M. P. Singh. Engineering foreign exchange processes via commitment protocols. *IEEE SCC*, pp. 514–521, 2007.

[5] N. Fornara, M. Colombetti. Operational specification of a commitment-based agent communication language. *AAMAS*, pp. 535–542, 2002.

[6] L. Giordano, A. Martelli, C. Schwind. Specifying and verifying interaction protocols in a temporal action logic. *J. Applied Logic*, 5(2):214–234, 2007.

[7] M. Lind, G. Goldkuhl. The constituents of business interaction—generic layered patterns. *Data & Knowledge Engineering*, 47(3):327–348, 2003.

[8] H. Mazouzi, A. E. F. Seghrouchni, S. Haddad. Open protocol design for complex interactions in multi-agent systems. *AAMAS*, pp. 517–526, 2002.

[9] P. McBurney, S. Parsons. Posit spaces: A performative model of e-commerce. *AAMAS*, pp. 624–631, 2003.

[10] A. Qumer, B. Henderson-Sellers. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4):280–295, 2008.

[11] M. Rovatsos. Dynamic semantics for agent communication languages. *AAMAS*, pp. 1–8, 2007.

[12] M. P. Singh, A. K. Chopra, N. Desai. Commitment-based service-oriented architecture. *IEEE Computer*, 42(11):72–79, 2009.

[13] P. R. Telang and M. P. Singh. Specifying and verifying cross-organizational business models. *IEEE Trans. Services Comput.*, 4, 2011.

[14] B. Vitteau, M.-P. Huget. Modularity in interaction protocols. *Proc. ACL*, *LNCS* 2922, pp. 291–309, 2004.

[15] J. Vokřínek, J. Bíba, J. Hodík, J. Vybíhal, M. Pěchouček. Competitive contract net protocol. *SOFSEM: Theory and Practice of Computer Science*, *LNCS* 4362, pp. 656–668, 2007.

[16] M. Wang, K. Ramamohanarao, J. Chen. Reasoning intra-dependency in commitments for robust scheduling. *AAMAS*, pp. 953–960, 2009.

On the Verification of Social Commitments and Time

Mohamed El Menshawy, Jamal Bentahar
Concordia University, Faculty of Engineering and Computer Science, Canada
m_elme@encs.concordia.ca, bentahar@ciise.concordia.ca

Hongyang Qu, Rachida Dssouli
Oxford University, Computing Laboratory, UK
Concordia University, Faculty of Engineering and Computer Science, Canada
Hongyang.Qu@comlab.ox.ac.uk, dssouli@ece.concordia.ca

ABSTRACT

Social commitments have been widely studied to represent business contracts among agents with different competing objectives in communicating multi-agent systems. However, their formal verification is still an open issue. This paper proposes a novel model-checking algorithm to address this problem. We define a new temporal logic, CTLC, which extends CTL with modalities for social commitments and their fulfillment and violation. The verification technique is based on symbolic model checking that uses ordered binary decision diagrams to give a compact representation of the system. We also prove that the problem of model checking CTLC is polynomial-time reducible to the problem of model checking CTLK, the combination of CTL with modalities for knowledge. We finally present the full implementation of the proposed algorithm by extending the MCMAS symbolic model checker and report on the experimental results obtained when verifying the NetBill protocol.

Categories and Subject Descriptors

D.2.4 [**Software/Program Verification**]: Model Checking

General Terms

Algorithms, Verification

Keywords

Social Commitments, Fulfillment, Violation

1. INTRODUCTION

Over the last two decades, a significant number of social approaches that aim to define a semantics for Agent Communication Languages (ACLs) have been proposed [1, 2, 9, 16, 19, 23]. These approaches particularly aim to overcome the shortcomings of ACLs semantics defined using mental (or cognitive) approaches where the mental semantics is expressed in terms of the agents' internal mental states such as believes, desires and intensions. Social commitments are employed in some of these social approaches that successfully provide a powerful basis to represent business contracts among autonomous and possibly heterogeneous agents with different competing objectives within multi-agent systems (MASs) [3, 7, 9, 19, 21]. Formally, social commitments are denoted by $\mathtt{C}(i, j, \varphi)$ meaning that i, the debtor, commits to j, the creditor, that φ holds [7, 8, 19].

Conventionally, the semantics of ACL messages in terms of social commitments satisfies some crucial criteria introduced in [19]: 1) formal (based on some temporal logics); 2) declarative (which focuses on what the message means not how the message is exchanged); 3) verifiable (we can check if the agents are acting according to the semantics); and 4) meaningful (the focus is on the content of messages, not on their representation as tokens). Recent research in agent communication using social commitments has highlighted their use in a variety of areas ranging from modeling business processes [8], developing artificial or virtual institutions [12], defining programming languages [22], developing web-based applications [21] to specifying multi-agent interaction protocols, called commitment protocols [2, 5, 7, 9, 16, 23]. In particular, these commitment protocols are more suitable for regulating and coordinating agent interactions than computer protocols formalized using *Finite State Machines* or *Petri Nets*, which only capture legal orderings of the exchanged messages without considering the meanings of those messages. Missing such meanings limits the ability to verify the compliance of agent behaviors with a given protocol.

Related Work. The motivation behind verifying that agents are acting according to a given commitment protocol was first investigated by Venkatraman and Singh [21]. They developed an approach for locally verifying whether the behavior of an agent complies with a given commitment protocol specified in *Computational Tree Logic* (CTL) [6]. Their verification method concentrates on the conditions under which an individual agent may check others' commitments toward itself. The ideas presented by Venkatraman and Singh were further complemented in two research works by Desai et al. [7] and Cheng [5]. They developed the idea of supporting the verification of properties geared toward the composition of commitment protocols. These properties are specified in *Linear Temporal Logic* (LTL) [6] and their approach depends on translating the protocol into PROMELA (the input language of the SPIN automata-based model checker) where commitments are represented as data structures [5] or processes [7]. Bentahar et al. [3] presented ACTL* logic (an extension of CTL*) to define semantics of social commitments and associated actions and specify multi-agent interaction protocols and some desirable properties. Their verification

Cite as: On the Verification of Social Commitments and Time, Mohamed El Menshawy, Jamal Benthar, Hongyang Qu and Rachida Dssouli, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 483-490.

technique is based on the translation of ACTL* formulae and protocol into a variant of alternating tree automata called *alternating Büchi tableau automata* (ABTA) in order to directly use the CWB−NC automata-based model checker where commitments are represented as variables and actions as atomic action propositions using CCS (the input language of CWB-NC). El-Menshawy et al. [9, 10] introduced CTL^{*sc} logic extending CTL* with commitments and associated commitment actions to drive a new specification language of multi-agent interaction protocols having social semantics. Their symbolic verification technique is based on reducing CTL^{*sc} logic into LTL^{sc} and CTL^{sc} sub-logics and then defining the participating agents in protocol as *SMV modules* using SMV and *agent sections* using ISPL (the input languages of the NuSMV and MCMAS symbolic model checkers respectively) where commitment states and commitment actions are defined as local state variables. Gerard and Singh [13] used CTL and MCMAS to verify the refinement of multi-agent interaction protocols having social semantics by developing a preprocessor tool that first reads protocols and specifications from files and then translates them into ISPL model in order to directly use MCMAS. However, the above frameworks are translation-based approaches, which have the following shortcomings: 1) they prevent verifying the real and concert semantics of commitments and related concepts as defined in the underlying logics and provide partial solution to the problem; 2) they may not be straightforward and prone to errors, particularly in the context of complex systems; and 3) they lack a full and dedicated model-checking algorithm.

The `motivation` of this paper is to address the above challenges by: 1) presenting a new semantics for social commitments and their fulfillment and violation using a new logic, CTLC, which extends CTL [6] with modalities for reasoning about social commitments and their fulfillment and violation (`Section 2`); 2) introducing a new model-checking algorithm to directly verify commitments and their fulfillment and violation (`Section 3`); and 3) presenting the full implementation of the proposed algorithm by extending the MCMAS symbolic model checker [15] (`Section 4`).

The introduction of a new logic is motivated by the fact that the needed modal connectives for social commitments and their fulfillment/violation cannot be expressed using only existing temporal logics, e.g. CTL. A dedicated logic and model checking for commitments play the same role as CTLK [17] and MCMAS do for knowledge. Furthermore, the election of CTL is motivated by our objective to balance between expressiveness and verification efficiency. Using more expressive languages such as *First Order Logic* (FOL) needs very complex and maybe intractable model checking. In fact, we prove that the problem of model checking CTLC is polynomial-time reducible to the problem of model checking CTLK (the combination of CTL with modalities for knowledge). For checking the effectiveness of our approach, we report on the experimental results obtained when verifying the NetBill protocol [20] taken from e-business domain. Our approach can complement the static verification method introduced in [23] to check the agent behaviors with given protocol specifications via an *event calculus planner*.

2. CTLC LOGIC

In this section, we briefly present the interpreted systems introduced in [11] to formalism MASs. The reason for using this formalism is the usefulness of ascribing autonomous and social behavior to the components of a system of agents. It also allows us to abstract from the details of the components and focus only on the interactions among the various agents. However, modeling complex and open systems such as MASs using the formalism of interpreted systems is typically conducted by using logic-based formalisms. Thus, we below present a new temporal logic called CTLC logic.

2.1 Interpreted Systems

An interpreted system as introduced by Fagin et al. [11] is a formalism that models the temporal evolution of a system of agents to reason about knowledge and temporal properties. In this formalism, the interpreted system is composed of a set of n agents $\mathtt{A} = \{1, \ldots, n\}$ and an environment e. This environment can be seen as a special agent that can capture any information, which may not pertain to a specific agent. For each agent $i \in \mathtt{A}$, we associate a set of local states L_i and a set of local states L_e is associated to the environment agent.

As in [11], we represent the instantaneous configuration of all agents in the MAS at a given time via the notion of global state. The set of all global states is denoted by S and a global state $s \in S$ is a tuple $s = (l_1, \ldots, l_n, l_e)$ where each component $l_i \in L_i$ represents a local state of agent i and l_e is an environment local state. Thus, the set of all global states $S \subseteq L_1 \times \ldots \times L_n \times L_e$ is a subset of the Cartesian product of all local states of n agents and local states of the environment in the system. We use the notation $l_i(s)$ to represent the local state of agent i in the global state s. $I \subseteq S$ is a set of initial global states for the system. To account for the temporal evolution of the system, the formalism of interpreted systems associates with each agent i the set Act_i of actions, and with environment the set Act_e of actions. It is assumed that $null \in Act_i$ for each agent i, where $null$ refers to the fact of doing nothing. Each agent $i \in \mathtt{A}$ has a local protocol $\mathcal{P}_i : L_i \rightarrow 2^{Act_i}$ to identify the set of the enabled actions that may be performed in a given local state. With the same meaning we can define $\mathcal{P}_e$.

As in [11], the interpreted system formalism is a synchronous model. So, we can define the global transition function as follows: $\tau : S \times ACT \rightarrow S$, where $ACT = Act_1 \times \ldots \times Act_n \times Act_e$ and each component $a \in ACT$ is a *joint action*, which is a tuple of actions (one for each agent). An evolution function t_i that determines the transitions for an individual agent i between its local states is defined as follows: $t_i : L_i \times ACT \rightarrow L_i$, where $t_i(l_i(s), null) = l_i(s)$. In a similar way, we have an evolution function for the environment's local states: $t_e : L_e \times ACT \rightarrow L_e$. Finally, given a set $\Phi_p = \{p, p_1, p_2, \ldots\}$ of atomic propositions and the valuation function V for those propositions $V : \Phi_p \rightarrow 2^S$, an interpreted system is a tuple:

$\mathcal{IS} = \langle (L_i, Act_i, \mathcal{P}_i, t_i)_{i \in \mathtt{A}}, (L_e, Act_e, \mathcal{P}_e, t_e), I, V \rangle$.

2.2 Syntax of CTLC

The proposed language CTLC is a multi-modal logic including branching time CTL [6] and modalities for social commitments and their fulfillment and violation.

Definition 1 (SYNTAX). *The syntax of CTLC logic is given by the following BNF grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{EX}\varphi \mid \mathtt{E}(\varphi \mathtt{U} \varphi) \mid \mathtt{EG}\varphi \mid \mathtt{C}(i, j, \varphi) \mid \mathtt{Fu}(\mathtt{C}(i, j, \varphi)) \mid \mathtt{Vi}(\mathtt{C}(i, j, \varphi))$$

In this definition, $p \in \Phi_p$ is an atomic proposition and $\mathtt{E}$ ("there exits a path") is the existential quantifier on paths. The formula $\mathtt{EX}\varphi$ stands for "φ holds in the next state in at least one path"; $\mathtt{E}(\varphi U \psi)$ stands for "there exists at least one path where ψ holds at some point in the future and φ holds in all states until then"; and $\mathtt{EG}\varphi$ stands for "there exists a path in which φ holds globally", i.e., φ holds in every future state in at least one path. Other temporal modalities, e.g., $\mathtt{F}$, and the universal path quantifier $\mathtt{A}$ ("for all paths") can be defined in terms of the above as usual, for examples, $\mathtt{AX}\varphi \triangleq \neg\mathtt{EX}\neg\varphi$ and $\mathtt{AG}\varphi \triangleq \neg\mathtt{EF}\neg\varphi$ where $\mathtt{EF}\varphi \triangleq \mathtt{E}(trueU\varphi)$. $\mathtt{A}(\varphi U \psi)$ has the obvious semantics. The modal connective $\mathtt{C}(i, j, \varphi)$ is read as "agent i commits towards agent j to bring about φ" or equivalently as "φ is committed to by i towards j". The modal connective $\mathtt{Fu}(\mathtt{C}(i, j, \varphi))$ is read as "$\mathtt{C}(i, j, \varphi)$ is fulfilled (discharged)" and the modal connective $\mathtt{Vi}(\mathtt{C}(i, j, \varphi))$ is read as "$\mathtt{C}(i, j, \varphi)$ is violated".

2.3 Semantics of CTLC

In order to define the semantics of CTLC formulae, a Kripke model $M = \langle W, I, R_t, R_c, V\rangle$ is associated to a given interpreted system $\mathcal{IS}$ as follows: the set of reachable[1] worlds W is the set S of global states for the system; $I \subseteq W$ is the set of initial states as defined in $\mathcal{IS}$; the temporal transition relation $R_t \subseteq W \times W$ for the system is defined using the local protocols and evolution functions and the two worlds w and w' are related by R_t (i.e., $(w, w') \in R_t$) iff there exists a joint action $(a_1, \ldots, a_n, a_e) \in ACT$ such that for all $i \in \mathtt{A}$, a_i and a_e are enabled by the protocols $\mathcal{P}_i$ and $\mathcal{P}_e$ respectively and $t_i(l_i(s), a_1, \ldots, a_n, a_e) = l_i(s')$; the relation $R_c : W \times A \times A \rightarrow 2^W$ is the social accessibility relation for social commitments. It is defined by $w' \in R_c(w, i, j)$ iff $\exists w'' \neq w$ such that: 1) $l_i(w) = l_i(w'') = l_i(w')$; and 2) $l_j(w'') = l_j(w')$; and V is an interpretation over the set of atomic propositions as defined in $\mathcal{IS}$.

The social accessibility relation R_c is transitive, symmetric, and Euclidean. Thus, the resulting logic of social commitments is $K4B5 \equiv KB5$. In this relation: 1) $l_i(w) = l_i(w'')$=$l_i(w')$ means that the local states of i in the global states w, w', and w'' are indistinguishable; and 2) $l_j(w'') = l_j(w')$ means that the local states of j in global states w'' and w' are indistinguishable where $w'' \neq w$. Intuitively, $w' \in R_c(w, i, j)$ means there is an intermediate state w'' so that there is no difference for the debtor i among being in w, w'' and w'; however, for the creditor j there is no difference between being in the intermediate state w'' and accessible state w'. This accessibility relation captures three fundamental issues: 1) the debtor's uncertainty about the current state ($w'' \neq w$); 2) the unchangeability of the debtor ($l_i(w)$=$l_i(w')$); and 3) the possible changeability of the creditor because of the intermediate state.

A path (or computation) $\pi = \langle w_i, w_{i+1}, w_{i+2}, \ldots\rangle$ such that for all $i \geq 0, (w_i, w_{i+1}) \in R_t$ is an infinite sequence of reachable global states in the system. $\pi(k)$ is the k^{th} global state of the path π. The set of all paths is denoted by Π, whilst Π^{w_i} is the set of all paths starting at the given state ($w_i \in W$). We define the set of states that are in the past of w ($\mathcal{P}as(w)$) as follows:

$$\mathcal{P}as(w) = \{w' \in W |\ (w', w) \in R_t \text{ or } \exists \pi \in \Pi \text{ such that } \pi = \langle w', \ldots, w, \ldots\rangle\} \cup \{w\}$$

We also define the set of states that are in the future of w ($\mathcal{F}ut(w)$) as follows:

$$\mathcal{F}ut(w) = \{w' \in W |\ (w, w') \in R_t \text{ or } \exists \pi \in \Pi \text{ such that } \pi = \langle w, \ldots, w', \ldots\rangle\} \cup \{w\}$$

Definition 2 (Satisfaction). *Satisfaction for a CTLC formula φ in the model M at a global state w, denoted as $\langle M, w\rangle \models \varphi$, is recursively defined as follows:*

- $\langle M, w\rangle \models p$ *iff* $w \in V(p)$;
- $\langle M, w\rangle \models \neg\varphi$ *iff* $M, \langle w\rangle \not\models \varphi$;
- $\langle M, w\rangle \models \varphi \vee \psi$ *iff* $\langle M, w\rangle \models \varphi$ *or* $\langle M, w\rangle \models \psi$;
- $\langle M, w\rangle \models \mathtt{EX}\varphi$ *iff there exists a path* π *starting at* w *such that* $\langle M, \pi(1)\rangle \models \varphi$;
- $\langle M, w\rangle \models \mathtt{E}(\varphi U \psi)$ *iff there exists a path* π *starting at* w *such that for some* $k \geq 0$, $\langle M, \pi(k)\rangle \models \psi$ *and* $\langle M, \pi(j)\rangle \models \varphi$ *for all* $0 \leq j < k$;
- $\langle M, w\rangle \models \mathtt{EG}\varphi$ *iff there exists a path* π *starting at* w *such that* $\langle M, \pi(k)\rangle \models \varphi$ *for all* $k \geq 0$;
- $\langle M, w\rangle \models \mathtt{C}(i, j, \varphi)$ *iff* $R_c(w, i, j) \neq \emptyset$ *and for all global states* $w' \in W$ *such that* $w' \in R_c(w, i, j)$ *we have* $\langle M, w'\rangle \models \varphi$;
- $\langle M, w\rangle \models \mathtt{Fu}(\mathtt{C}(i, j, \varphi))$ *iff there exists* w' *such that: 1)* $\langle M, w'\rangle \models \mathtt{C}(i, j, \varphi)$; *and 2)* $w \in \mathcal{F}ut(w')$; *and 3)* $w \in R_c(w', i, j)$;
- $\langle M, w\rangle \models \mathtt{Vi}(\mathtt{C}(i, j, \varphi))$ *iff there exists* w' *such that: 1)* $\langle M, w'\rangle \models \mathtt{C}(i, j, \varphi)$; *and 2)* $w \in \mathcal{F}ut(w')$; *and 3) for all* $w'' \in \mathcal{P}as(w) \cup \mathcal{F}ut(w)$ *we have* $w'' \notin R_c(w', i, j)$.

Excluding the commitment and its fulfillment (discharge) and violation, the semantics of CTLC state formulae is defined in the model M as usual (semantics of CTL)—see for example [6, 11]. The state formula $\mathtt{C}(i, j, \varphi)$ is satisfied in the model M at w iff the set of accessible states obtained by the social accessibility relation $R_c(w, i, j)$ is not empty and the content φ is true in every global state in this set. Note that this semantics requires checking whether or not $R_c(w, i, j) \neq \emptyset$ because the social accessibility relation is not necessarily reflexive[2] like for the epistemic accessibility relation $\sim_i$ for agent i in the logic of knowledge [11]. In this logic, the epistemic accessibility relation $\sim_i \subseteq W \times W$ represents that two global states are "indistinguishable" for this agent. Formally, $w \sim_i w'$ iff $l_i(w) = l_i(w')$ [11]. In fact, the emptiness checking is compatible with the uncertainty of agent i about the current state. The state formula $\mathtt{Fu}(\mathtt{C}(i, j, \varphi))$ is satisfied in the model M at w iff there exists a state w' satisfying the commitment (condition 1) and the current state (i.e., w) is both in the future of w' and accessible via the accessability relation $R_c(w', i, j)$ (conditions 2 and 3). The intuition behind $\mathtt{Fu}$'s semantics is to ensure that the current state w is reachable in terms of transitions and accessible in terms of the social accessibility relation from the state w' where the commitment holds, because to be fulfilled, the commitment should prior exist.

[1] W contains states in S that are reachable from I using R_t.

[2] This means that reflexivity is not always satisfied.

Conversely, the state formula $\texttt{Vi}(\texttt{C}(i,j,\varphi))$ is satisfied in the model M at w iff there exists a state w' satisfying the commitment and the current state (i.e., w) is in the future of w' such that every state both in the past and future of w is not accessible in terms of the accessability relation $R_c(w',i,j)$. The main motivation behind including Fu and Vi modal connectives is to ensure that an agent can detect if there exists a conflict among its commitment states. For example, from the semantics, we can easily check that when the commitment is violated, then there is no way to fulfill it in the future and it has not been fulfilled in the past and vice versa.

3. MODEL CHECKING CTLC FORMULAE

In a nutshell, given the model M representing a MAS w.r.t the formalism of interpreted system $\mathcal{IS}$ and a formula φ in CTLC describing a property, the problem of model checking can be defined as establishing whether or not $M \models \varphi$, i.e., $\forall w \in I : \langle M, w\rangle \models \varphi$. Symbolic approaches have been recently proven as an efficient technique to automatically verify MASs [15, 18]. This is because these approaches use less memory than automata-based approaches as their algorithms are applied to Boolean Functions (BFs) not to Kripke structures. In practice, space requirements for BFs that can be represented using ordered binary decision diagrams (OBDDs) [4] are exponentially smaller than for explicit representation. As a result, these approaches alleviate the "state explosion" problem, but cannot eliminate it totally as the space still increases when the model is getting larger.

In general, symbolic model checking techniques address the state explosion problem by computing the set of states satisfying φ in the model M (denoted by $[\![\varphi]\!]$), which is represented in OBDDs and then comparing it against the set of initial states I in M that is also represented in OBDD. If $I \subseteq [\![\varphi]\!]$, then the model M satisfies the formula; otherwise a counter example can be generated showing why the model does not satisfy the formula. This paper is only concerned with developing a new symbolic model-checking algorithm $SMC(\varphi, M)$ to compute the set $[\![\varphi]\!]$ of states satisfying a CTLC formula φ. This algorithm also provides a methodology to build the OBDD corresponding to $[\![\varphi]\!]$. For example, when the sets of states are encoded using BFs, all operations (e.g., intersection) on sets are translated into operations (e.g., conjunction) on BFs.

3.1 Symbolic Model-Checking Algorithm

The basic idea of our main $SMC(\varphi, M)$ algorithm is inspired by the standard symbolic procedure introduced in [14] for computing the set of states in M satisfying the formula φ in CTL (see Algorithm 1). In particular, we extend this algorithm by adding the procedures that deal with the new modalities of our logic. It starts by checking atomic formulae (line 1) and Boolean operators: negation and disjunction (lines 2 and 3). In lines 4 to 6, the algorithm calls the standard procedures $SMC_{\texttt{EX}}(\varphi_1, M)$, $SMC_{\texttt{EU}}(\varphi_1, \varphi_2, M)$ and $SMC_{\texttt{EG}}(\varphi_1, M)$ introduced in [14] to check the formulae having the forms $\texttt{EX}\varphi_1$, $\texttt{E}(\varphi_1 U \varphi_2)$ and $\texttt{EG}\varphi_1$ respectively. It then checks the commitment modality (line 7) by calling the procedure $SMC_{\texttt{c}}(i,j,\varphi_1,M)$ (see Algorithms 2 and 3). The algorithm proceeds to check the satisfiability of $\texttt{Fu}(\texttt{C}(i,j,\varphi_1))$ and $\texttt{Vi}(\texttt{C}(i,j,\varphi_1))$ by calling respectively the procedures $SMC_{\texttt{Fu}}(i,j,\varphi_1,M)$ (see Algorithm 4) and $SMC_{\texttt{Vi}}(i,j,\varphi_1,M)$ (see Algorithm 5) (lines 8 and 9).

Algorithm 1 $SMC(\varphi, M)$: the set $[\![\varphi]\!]$ satisfying the CTLC formula φ

1: φ is an atomic formula: return $V(\varphi)$
2: φ is $\neg\varphi_1$: return $W \backslash SMC(\varphi_1, M)$
3: φ is $\varphi_1 \vee \varphi_2$: return $SMC(\varphi_1, M) \cup SMC(\varphi_2, M)$
4: φ is $\texttt{EX}\varphi_1$: return $SMC_{\texttt{EX}}(\varphi_1, M)$
5: φ is $\texttt{E}(\varphi_1 U \varphi_2)$: return $SMC_{\texttt{EU}}(\varphi_1, \varphi_2, M)$
6: φ is $\texttt{EG}\varphi_1$: return $SMC_{\texttt{EG}}(\varphi_1, M)$
7: φ is $\texttt{C}(i,j,\varphi_1)$: return $SMC_{\texttt{c}}(i,j,\varphi_1,M)$
8: φ is $\texttt{Fu}(\texttt{C}(i,j,\varphi_1))$: return $SMC_{\texttt{Fu}}(i,j,\varphi_1,M)$
9: φ is $\texttt{Vi}(\texttt{C}(i,j,\varphi_1))$: return $SMC_{\texttt{Vi}}(i,j,\varphi_1,M)$

3.1.1 BDD-based Algorithm for Commitments

We use the social accessibility relation R_c to compute the set $[\![\texttt{C}(i,j,\varphi)]\!]$ of states in which the formula $\texttt{C}(i,j,\varphi)$ holds, as reported in the procedure of Algorithm 2. This procedure firstly computes the set X_1 of states in which the formula φ holds where φ is the commitment content. It then builds X_2, the set of states that have at least one accessible state via R_c and all the accessible states from each state in this set (i.e., X_2) are in X_1, which means they satisfy φ. The set $[\![\texttt{C}(i,j,\varphi)]\!]$ is finally computed by returning the set X_2.

Algorithm 2 $SMC_{\texttt{c}}(i,j,\varphi,M)$: the set $[\![\texttt{C}(i,j,\varphi)]\!]$

1: $X_1 \leftarrow SMC(\varphi, M)$
2: $X_2 \leftarrow \{w \in W |\ R_c(w,i,j) \neq \emptyset$ and $\forall w' \in R_c(w,i,j)$ we have $w' \in X_1\}$
3: return X_2

Example 1. *To clarify the computation of each set of states in each proposed BDD-based algorithm, we consider the following example. It consists of eight global states and the transitions between them along with the social accessibility relation R_c and the epistemic accessibility relations $\sim_i$ and $\sim_j$ such that w_1, w_2, w_3, w_4, w_5 and w_8 hold the formula φ and w_7 does not satisfy φ (see Figure 1).*

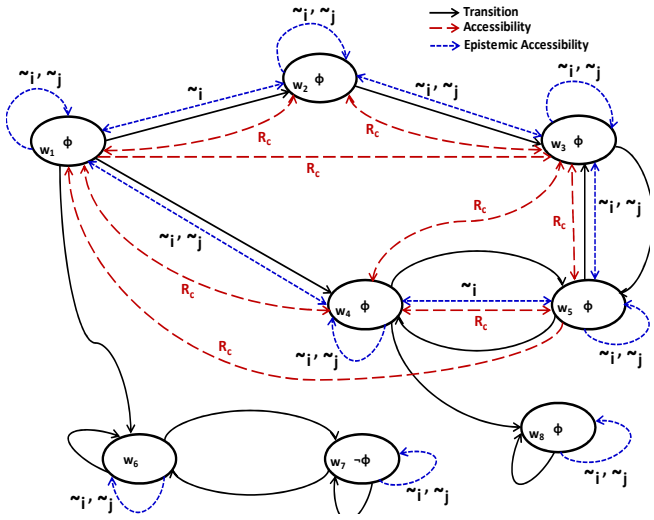


Figure 1: An example of R_c along with $\sim_i$ and $\sim_j$

Note that, $w' \in R_c(w,i,j)$ iff $\exists w'' \neq w$ such that $w \sim_i w''$ $\sim_i w'$ and $w'' \sim_j w'$. The reason behind using $\sim_i$ and $\sim_j$ in Figure 1 will be motivated later on. From example 1,

the computation of $SMC_{\mathtt{c}}$ algorithm (see Algorithm 2) is as follows: $X_1 = \{w_1, w_2, w_3, w_4, w_5, w_8\}$, $X_2 = \{w_1, w_2, w_3, w_4, w_5\}$. Finally, the algorithm returns X_2 (see Figure 1).

The procedure reported in Algorithm 2 represents a direct implementation of the proposed semantics of the social commitment modality. We can use an alternative procedure to implement it, which is more efficient by using the negation of the formula φ and the existential quantifier "$\exists$" instead of the universal one "$\forall$" in computing the sets X_1 and X_2 (see Algorithm 3). Note that, X_3 ensures that R_c is not empty and $\overline{X}_2$, in line 4, is the complement of X_2. From example 1, $X_1 = \{w_7\}$ contains the states satisfying $\neg\varphi$, $X_2 = \emptyset$, $X_3 = \{w_1, w_2, w_3, w_4, w_5\}$, $\overline{X}_2 = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8\}$ and $\overline{X}_2 \cap X_3 = \{w_1, w_2, w_3, w_4, w_5\}$, which is the same result obtained by Algorithm 2 (see Figure 1).

Algorithm 3 $SMC_{\mathtt{c}}(i, j, \varphi, M)$: the set $[\![\mathtt{C}(i, j, \varphi)]\!]$

1: $X_1 \leftarrow SMC(\neg\varphi, M)$
2: $X_2 \leftarrow \{w \in W |\ \exists w' \in X_1$ such that $w' \in R_c(w, i, j)\}$
3: $X_3 \leftarrow \{w \in W |\ R_c(w, i, j) \neq \emptyset\}$
4: `return` $\overline{X_2} \cap X_3$

3.1.2 BDD-based Algorithm for Fulfillment

The procedure $SMC_{\mathtt{Fu}}(i, j, \varphi, M)$ starts with computing the set X_1 of states satisfying the commitment $\mathtt{C}(i, j, \varphi)$ (see Algorithm 4). It then constructs the set X_2 of accessible states that can "see" by means of the social accessibility relation R_c a state in X_1. It then proceeds to compute the set X_3 of those states (i.e., X_2), which are reachable using transitions from the states in X_1 by calling the procedure $\mathcal{F}uture(X_1)$ (see Algorithm 7). From example 1, $X_1 = \{w_1, w_2, w_3, w_4, w_5\}$, $X_2 = \{w_1, w_2, w_3, w_4, w_5\}$, $\mathcal{F}uture(X_1) = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8\}$ and $X_3 = \mathcal{F}uture(X_1) \cap X_2 = \{w_1, w_2, w_3, w_4, w_5\}$. The algorithm finally returns X_3 (see Figure 1). It is clear that the main motivation of computing X_3 is to eliminate the states that are reachable but never accessible from the states of X_1 (e.g., w_8).

Algorithm 4 $SMC_{\mathtt{Fu}}(i, j, \varphi, M)$: the set $[\![\mathtt{Fu}(\mathtt{C}(i, j, \varphi))]\!]$

1: $X_1 \leftarrow SMC_{\mathtt{c}}(i, j, \varphi, M)$
2: $X_2 \leftarrow \{w \in W |\ \exists w' \in X_1$ and $w \in R_c(w', i, j)\}$
3: $X_3 \leftarrow \mathcal{F}uture(X_1) \cap X_2$
4: `return` X_3

3.1.3 BDD-based Algorithm for Violation

The procedure $SMC_{\mathtt{Vi}}(i, j, \varphi, M)$ starts with computing the set X_1 of states satisfying the commitment $\mathtt{C}(i, j, \varphi)$. It then computes the set X_2 of those states, which are accessible and reachable from each state w' in X_1 via the social accessibility relation and transitions. The procedure proceeds to compute the set X_4 of all global states in the system that are not reachable from and cannot reach the accessible states in X_2. Finally, the procedure returns those states (i.e., in X_4), which are in the future of states where the commitment holds (i.e., $X_3 \cap X_4$) (see Algorithm 5). From example 1, $X_1 = \{w_1, w_2, w_3, w_4, w_5\}$, $X_2 = \{w_1, w_2, w_3, w_4, w_5\}$, $X_3 = \mathcal{F}uture(X_1) = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8\}$ and $X_4 = W - \mathcal{P}ast(X_2) \cup \mathcal{F}uture(X)_2 = \{w_6, w_7\}$. Finally, the algorithm returns $X_3 \cap X_4 = \{w_6, w_7\}$. From Figure 1, it is obvious that w_6 and w_7 are the two states where the commitment $\mathtt{C}(i, j, \varphi)$ holding at w_1 is violated as they are reachable but not accessible from w_1.

Algorithm 5 $SMC_{\mathtt{Vi}}(i, j, \varphi, M)$: the set $[\![\mathtt{Vi}(\mathtt{C}(i, j, \varphi))]\!]$

1: $X_1 \leftarrow SMC_{\mathtt{c}}(i, j, \varphi, M)$
2: $X_2 \leftarrow \{w \in W |\ \exists w' \in X_1$ and $w \in R_c(w', i, j) \cap \mathcal{F}uture(\{w'\})\}$
3: $X_3 \leftarrow \mathcal{F}uture(X_1)$
4: $X_4 \leftarrow W - (\mathcal{P}ast(X_2) \cup \mathcal{F}uture(X_2))\}$
5: `return` $X_3 \cap X_4$

The above Algorithm 5 calls two procedures $\mathcal{P}ast(X)$ and $\mathcal{F}uture(X)$ that compute the set of past (resp. future) states of X (see Algorithms 6 and 7). Algorithm 6 reports the procedure $\mathcal{P}ast(X)$ by calling the standard procedure $\mathtt{pre}_\exists(X)$

Algorithm 6 $\mathcal{P}ast(X)$: the set of past states of X

1: $Y \leftarrow \mathtt{pre}_\exists(X) \cup X$
2: $Z \leftarrow \emptyset$
3: `While` $Z \neq Y$ `do`
4: $\quad Z' \leftarrow Z$
5: $\quad Z \leftarrow Y$
6: $\quad Y \leftarrow Y \cup \mathtt{pre}_\exists(Y - Z')$
7: `end While`
8: `return` Y

introduced in [14]. The main idea of $\mathcal{P}ast(X)$ procedure is to iterate using `while...do` construct over the set of past states captured by $\mathtt{pre}_\exists(X)$ until reaching the fix-point. Note that, line 1 reflects the idea that each state is the past of itself. The procedure $\mathtt{pre}_\exists(X)$ takes a set $X \subseteq W$ as input and computes the set of states $Y \subseteq W$ such that a transition is enabled to a state in X. Formally:

$Y = \mathtt{pre}_\exists(X) \leftarrow \{w \in W |\ \exists w'\ s.t.\ w' \in X\ and\ (w, w') \in R_t\}$

Similarly, the procedure $\mathcal{F}uture(X)$ depends on the procedure $\mathtt{next}_\exists(X)$, which is computationally the dual of the procedure $\mathtt{pre}_\exists(X)$ (i.e., it computes the next states enabled by the transition from the current state). The $\mathtt{next}_\exists(X)$ procedure is formally defined as follows:

$Y = \mathtt{next}_\exists(X) \leftarrow \{w \in W |\ \exists w'\ s.t.\ w' \in X\ and\ (w', w) \in R_t\}$

Algorithm 7 $\mathcal{F}uture(X)$: the set of future states of X

1: $Y \leftarrow \mathtt{next}_\exists(X) \cup X$
2: $Z \leftarrow \emptyset$
3: `While` $Z \neq Y$ `do`
4: $\quad Z' \leftarrow Z$
5: $\quad Z \leftarrow Y$
6: $\quad Y \leftarrow Y \cup \mathtt{next}_\exists(Y - Z')$
7: `end While`
8: `return` Y

This section is concluded by the following theorem:

THEOREM 1. *Model checking CTLC is polynomial-time reducible to the problem of model checking CTLK, the combination of CTL with the logic of knowledge (i.e., CTLC $\leq_p$ CTLK).*

PROOF. In order to prove this theorem, we present the semantics of the epistemic modality $\mathtt{K}_i\varphi$, which means "agent

i knows φ" [17]. Given a formula φ of CTLK, $\langle M, w\rangle \models \mathtt{K}_i\varphi$ iff for all $w' \in W$ such that $w \sim_i w'$ we have $\langle M, w'\rangle \models \varphi$.

Let ψ be a formula in CTLC, based on the structure of the formula ψ, three cases should be analyzed:

`Case 1`: $\psi = \mathtt{C}(i, j, \varphi)$.

From Section 2.3, $w' \in R_c(w, i, j)$ iff $\exists w'' \neq w$ such that $w \sim_i w'' \sim_i w'$ and $w'' \sim_j w'$. Therefore:

1) if $w \neq w'$ then $w' \in R_c(w, i, j)$ iff $w \sim_i w'$ as $\sim_i$ and $\sim_j$ are reflexive. Because the comparison of w and w' can be done in a polynomial time, the reduction in this case can be done in a polynomial amount of time.

2) if $w = w'$, to check if $w' \in R_c(w, i, j)$, we use the algorithm:

```
for all w'' such that w ∼_i w''
    if w'' ∼_j w return true
    return false
```

Since this algorithm is linear with the size of the model, the reduction of Case 1 can be done in a polynomial amount of time.

`Case 2`: $\psi = \mathtt{Fu}(\mathtt{C}(i, j, \varphi))$.

In this case, three steps are needed: 1) $\langle M, w'\rangle \models \mathtt{C}(i, j, \varphi)$; 2) $w \in \mathcal{F}ut(w')$; and 3) $w \in R_c(w', i, j)$. Step 1 is reducible in a polynomial time (Case 1). Step 2 is reducible to the future in CTLK in a polynomial time. Step 3 can be done in a polynomial time (see Case 1). Thus, the reduction of Case 2 can be also done in a polynomial amount of time.

`Case 3`: $\psi = \mathtt{Vi}(\mathtt{C}(i, j, \varphi))$.

In this case, three steps are also needed: 1) $\langle M, w'\rangle \models \mathtt{C}(i, j, \varphi)$; 2) $w \in \mathcal{F}ut(w')$; and 3) for all $w'' \in \mathcal{P}as(w) \cup \mathcal{F}ut(w)$ we have $w'' \notin R_c(w', i, j)$. Steps 1 and 2 are likewise steps 1 and 2 in Case 2. In step 3, checking the membership is linear with the size of the model and since the union of 2 sets can be done in a polynomial time, then the reduction of Case 3 can also be done in a polynomial amount of time, which completes the proof. □

It is obvious that model checking CTL is also polynomial-time reducible to the problem of model checking CTLC. We can conclude that $CTL \leq_p CTLC \leq_p CTLK$).

4. IMPLEMENTATION

This section includes a description of the extensions made on top of MCMAS to implement our BDD-based algorithms presented in Section 3.1. MCMAS [15] is developed particularly to verify MASs formalized using the interpreted systems. It also implements BDD-based algorithms to verify CTL modal connectives, epistemic logic, alternating time logic and deontic operators. MCMAS is developed in $C++$ and uses the efficient CUDD library that provides BDD data structure and performs OBDD operations and asynchronous variable reordering. It also provides fairness, counter-examples, witness generation and interactive execution.

4.1 BDD-based Algorithm of Commitments

As we mentioned, the needed BDD-based algorithms of CTL modal connectives are implemented in MCMAS. In order to fully implement the BDD-based algorithm $SMC_{\mathtt{c}}$ of social commitments on top of MCMAS, we need to perform the following two steps: 1) extend the method `check_formulae` in the `modal_formulae` class in the `parser` directory to handel the new commitment modality `C`; and 2) add the new BDD-based algorithm of commitment modality `C` along with other related methods into `utilities.cc` in the `utilities` directory. The motivation behind step 1 is to enforce the MCMAS's syntax [15] to accept the proposed new grammar specified in Definition 2.2. To achieve step 2, the BDD-based algorithm of social commitments (see Algorithm 3) is rewritten using the epistemic accessibility relations (i.e., $\sim_i$ and $\sim_j$) that define our social accessibility relation R_c. The set

Algorithm 8 $SMC_{\mathtt{c}}(i, j, \varphi, M)$: the set $[\![\mathtt{C}(i, j, \varphi)]\!]$

1: $X_1 \leftarrow SMC(\neg\varphi, M)$
2: $X_2' \leftarrow \{w \in W \mid \exists w' \in X_1$ such that $w \sim_i w'$ and $w \sim_j w'\}$
3: $X_2'' \leftarrow \{w \in W \mid \exists w' \in X_2'$ such that $w \sim_i w'$ and $w \neq w'\}$
4: $X_3 \leftarrow \{w \in W \mid \exists w' \in W$ such that $w \sim_i w'$ and $w \neq w'\}$
5: `return` $(W - X_2'') \cap X_3$

X_2 in the original BDD-based algorithm of social commitments is refined into two sets X_2' and X_2'' w.r.t. $\sim_i, \sim_j$ and $\sim_i$ respectively. Also, the set X_3 that checks the emptiness of R_c, in Algorithm 3, is rewritten w.r.t. $\sim_i$ (see Algorithm 8). The set $[\![\mathtt{C}(i, j, \varphi)]\!]$ is finally computed by returning the set of all global states W, which differs from the states accessible from states satisfying $\neg\varphi$ (i.e., X_2'') and accessible from all states in W w.r.t $\sim_i$ (i.e., in X_3)

In a similar way, we can easily perform the above two steps to implement the BDD-based algorithms $SMC_{\mathtt{Fu}}$ and $SMC_{\mathtt{Vi}}$ of `Fu` and `Vi` modal connectives respectively.

4.2 A Motivating Case Study

In this section, we provide a description of our motivating case study, called the NetBill protocol [20], which we used to evaluate the effectiveness of the proposed model-checking algorithm. The NetBill protocol is a security and transaction protocol optimized for the selling and delivery of low-priced information goods over the Internet. The original wording from [20] is as follows:

"*The NetBill payment protocol is eight steps (see Figure 2). The first message requests a quote based on the customer's identity, to allow for customized per-user pricing, such volume discounts or support for subscriptions. If the quote (step two) is accepted (step three), the merchant sends*

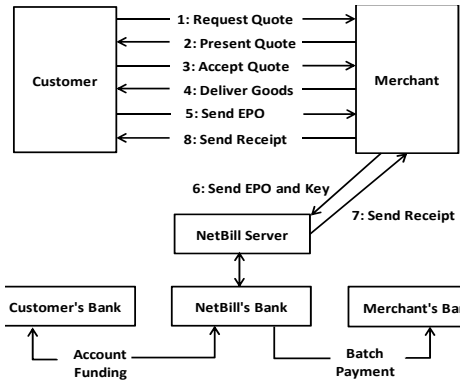


Figure 2: The NetBill payment protocol

the digital information to the customer (step four) but encrypts and withholds the key. The customer software constructs an electronic payment order (EPO) describing the transaction and including cryptographic checksum of the goods received. The order is signed with the customer's private key and sent to the merchant, who verifies its contents, appends the key for decrypting the goods, endorses the EPO

with a digital signature and sends it on to the NetBill server. The NetBill server verifies funds in the customer's NetBill account, debiting the customer and crediting the merchant, and digitally signed receipt, including the key to decrypt the goods, is sent first to the merchant and then on to the customer. The customer software can now decrypt the purchased information and present it to the customer".

Modeling NetBill Protocol

We used our formal model $M = \langle W, I, R_t, R_c, V\rangle$ associated to the interpreted system $\mathcal{IS}$ to model the NetBill Protocol. As in [9, 23], we omit the banking procedures by assuming that if a merchant gets an EPO, he can take care of it successfully. In this setting, the protocol rules interactions among two agents: the merchant (Mer) and customer (Cus). Each agent has a set of local states, a set of local actions, local protocol, local evolution function and local initial state. Because of space limit, we omit the details of the modeling process. As in [9, 23], the following abbreviations capture the commitments that exist in this protocol:

- `acceptQuote` abbreviates $goods \rightarrow \mathtt{C}(Cus, Mer, pay)$, which means that the customer commits to pay the agreed amount if he receives the goods.
- `promiseGoods` abbreviates `acceptQuote` $\rightarrow \mathtt{C}(Mer, Cus, goods)$, which means that the merchant commits to sending the requested goods if the customer commits to paying the agreed amount.
- `promiseReceipt` abbreviates $pay \rightarrow \mathtt{C}(Mer, Cus, receipt)$, which means that the merchant commits to sending the receipt if the customer pays the agreed amount.
- `offer` abbreviates `promiseGoods` $\wedge$ `promiseReceipt`.

The above commitments are established by exchanging messages among agents. These messages can also bring about certain propositions. For example, by exchanging "`send Goods`" message, we can realize the proposition "`goods`".

Specifications

To verify the NetBill protocol, various protocol properties are formalized using CTLC logic w.r.t the model M.

`Reachability property`. Given a particular state, is there a valid computation sequences to reach that state from an initial state. The following lists the formulae that can be used to check the reachable states in the NetBill protocol:

$$\varphi_1 = \mathtt{E}(\neg goods\ U\ (goods \wedge \mathtt{C}(Cus, Mer, pay)))$$
$$\varphi_2 = \mathtt{E}(\neg \mathtt{acceptQuote}\ U\ (\mathtt{acceptQuote} \wedge \mathtt{C}(Mer, Cus, goods)))$$
$$\varphi_3 = \mathtt{E}(\neg pay\ U\ (pay \wedge \mathtt{C}(Mer, Cus, receipt)))$$

For example, the formula φ_1 means that there exists a path where the customer will not commit to send payment to the merchant until he receives the requested goods.

`Safety property`. This property means "something bad never happens". For example, a bad situation is: the customer sends payment, but the merchant never commits to send the receipt to him:

$$\varphi_4 = \mathtt{AG}\ \neg(pay \wedge \neg \mathtt{C}(Mer, Cus, receipt))$$

`Liveness property`. This property means "something good will eventually happen". For example, in all paths globally if the customer requests a price quote, then in all paths in the future the merchant will commit to deliver the goods:

$$\varphi_5 = \mathtt{AG}(\mathtt{reqQuote} \rightarrow \mathtt{AF}\ (\mathtt{C}(Mer, Cus, goods))$$

`Fulfillment Commitment`. While verifying the behavior of agents for commitment fulfillment, it is crucial to verify some conditions under which the commitment fulfillment can occur. For example, when the customer sends the payment to the merchant, the commitment is successfully fulfilled:

$$\varphi_6 = \mathtt{EF}\ \mathtt{Fu}(\mathtt{C}(Cus, Mer, pay))$$

`Violation Commitment`. In a similar way, when the customer fails to send the agreed amount of payment to the merchant, the commitment is violated as the customer violates the protocol specification:

$$\varphi_7 = \mathtt{EF}\ \mathtt{Vi}(\mathtt{C}(Cus, Mer, pay))$$

4.3 Experimental Results

We encoded the NetBill protocol and the above properties in the ISPL model and verified them using the proposed algorithm implemented on top of MCMAS. In order to provide a thorough assessment, we tested our implementation on 10 experiments (see Table 1). These experiments are ranged from 1 customer requests goods from 1 merchant to 10 customers request goods from 10 merchants. The experiments were meant to check the effectiveness of the proposed algorithm in terms of execution time and memory in use. They are performed on an AMD Phenom(tm) 9600B Quad-Core Processor with 8GB memory running Fedora 12 x86_64 Linux. In fact, from experiment 2 we rewrite the defined properties in a parameterized form, for example in experiment 10:

$$\varphi_1' = \mathtt{E}(\bigwedge_{i=1}^{10} \neg goods_i\ U\ \bigwedge_{i=1}^{10} goods_i \bigwedge_{i=1}^{10} \mathtt{C}(Cus_i, Mer_i, pay_i))$$

which means that there exists a path where the ten customers will not commit to send payment to the ten merchants until they receive the requested goods.

Table 1 reports the number of reachable states, the execution time (in seconds) and BDD memory in use (in MBs) obtained in the verification of the NetBill protocol against the above properties, as a function of the number of customer and merchant agents (first and second columns). We found

Table 1: Verification Results

#Cus	#Mer	#States	Memory	Time
1	1	10	8.6 MB	< 0.01s
2	2	43	8.971 MB	< 0.01s
3	3	239	9.958 MB	< 0.01s
4	4	1597	12.056 MB	< 0.01s
5	5	11545	16.856 MB	1s
6	6	88055	36.134 MB	2s
7	7	708461	45.592 MB	8s
8	8	6.01734e+06	56.28 MB	29s
9	9	5.25729e+07	94.36 MB	426s
10	10	4.59517e+08	153.008 MB	1128s

that: 1) all the defined properties hold in the 10 experiments; and 2) the execution time and number of reachable states increase exponentially when the number of agents increases because the number of Boolean variables required to encode agents increases. However, the memory consumption does not increase exponentially because OBDDs encoding may

change from one model to another based on some internal optimization techniques. Furthermore, we did not compare our approach with others because unlike our proposal, they are based upon the translation process and do not use a dedicated model checker.

5. CONCLUSION AND FUTURE WORK

To have a full and dedicated model checking for social commitments and related concepts such as fulfillment and violation, a new temporal logic, called CTLC, is presented in this paper. Without such a logic, these concepts can only be encoded and abstracted as simple variables, processes or data structures in existing model checkers. Our CTLC logic extends CTL with modalities for social commitments and their fulfilment and violation. We developed a new model-checking algorithm that extended MCMAS to be able to verify commitments. We proved that the problem of model checking CTLC is polynomial-time reducible to the problem of model checking CTLK. In our implementation, we conducted 10 experiments, which demonstrate the effectiveness of our algorithm in terms of execution time and memory consumption. As future work, we plan to extend the proposed logic and its model checking to consider conditional commitments and commitment actions such as cancel, release, assign and delegate.

Acknowledgements

We thank the anonymous reviewers. Jamal Bentahar and Rachida Dssouli would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) and Fond Québecois de la recherche sur la société et la culture (FQRSC) for their financial support. Hongyang Qu would like to thank the European FP 7 project CONNECT (IST 231167).

6. REFERENCES

[1] M. Alberti, D. Daolio, P. Torroni, M. Gavanelli, E. Lamma, and P. Mello. Specification and Verification of Agent Interaction Protocols in a Logic-based System. In H. Haddad, A. Omicini, R. L. Wainwright, and L. M. Liebrock, editors, *SAC*, pages 72–78. ACM, 2004.

[2] M. Baldoni, C. Baroglio, and E. Marengo. Behavior Oriented Commitment-based Protocols. In H. Coelho, R. Studer, and M. Wooldridge, editors, *ECAI*, volume 215, pages 137–142. IOS Press, 2010.

[3] J. Bentahar, J.-J. C. Meyer, and W. Wan. Model Checking Agent Communication. In M. Dastani, K. V. Hindriks, and J.-J. C. Meyer, editors, *Specification and Verification of Multi-Agent Systems*, pages 67–102. Springer, First edition, 2010.

[4] R. E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. on Computers*, 35(8):677–691, 1986.

[5] Z. Cheng. *Verifying Commitment based Business Protocols and their Compositions: Model Checking using Promela and Spin*. PhD thesis, North Carolina State University, 2006.

[6] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, Cambridge, 1999.

[7] N. Desai, Z. Cheng, A. K. Chopra, , and M. P. Singh. Toward Verification of Commitment Protocols and their Compositions. In *Proc. of the 6th Int. Joint Conf. on AAMS*, pages 144–146. ACM, 2007.

[8] N. Desai, A. K. Chopra, and M. P. Singh. Amoeba: A Methodology for Modeling and Evolution of Cross-Organizational Business Processes. *ACM Trans. on Software Eng. and Methodology*, 19(2):1–40, 2009.

[9] M. El-Menshawy, J. Bentahar, and R. Dssouli. Verifiable Semantic Model for Agent Interactions using Social Commitments. In M. Dastani, A. E. Fallah-Seghrouchni, J. Leite, and P. Torroni, editors, *LADS*, volume 6039 of *LNCS*, pages 128–152, 2010.

[10] M. El-Menshawy, W. Wan, J. Bentahar, and R. Dssouli. Symbolic Model Checking for Agent Interactions (Extended Abstract). In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen, editors, *AAMAS*, pages 1555–1556. ACM, 2010.

[11] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. The MIT Press, Cambridge, 1995.

[12] N. Fornara, F. Viganò, M. Verdicchio, and M. Colombetti. Artificial Institutions: A Model of Institutional Reality for Open Multi-Agent Systems. *AI and Law*, 16(1):89–105, 2008.

[13] S. N. Gerard and M. P. Singh. Protocol Refinement: Formalization and Verification. In A. Artikis, J. Bentahar, A. K. Chopra, and F. Dignum, editors, *AAMAS Workshop on Agent Communication (AC)*, pages 19–36, 2010.

[14] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about System*. Cambridge University Press, Second edition, 2004.

[15] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. In A. Bouajjani and O. Maler, editors, *CAV*, volume 5643 of *LNCS*, pages 682–688. Springer, 2009.

[16] A. U. Mallya and M. P. Singh. An Algebra for Commitment Protocols. *Autonomous Agents and Multi-Agent Systems*, 14(2):143–163, 2007.

[17] W. Penczek and A. Lomuscio. Verifying Epistemic Properties of Multi-Agent Systems via Bounded Model Checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.

[18] F. Raimondi. *Model Checking Multi-Agent Systems*. PhD thesis, University College London, 2006.

[19] M. P. Singh. A Social Semantics for Agent Communication Languages. In F. Dignum and M. Greaves, editors, *Issues in Agent Communication*, volume 1916 of *LNCS*, pages 31–45. Springer, 2000.

[20] M. A. Sirbu. Credits and Debits on the Internet. *IEEE Spectrum*, 34(2):23–29, 1997.

[21] M. Venkatraman and M. P. Singh. Verifying Compliance with Commitment Protocols: Enabling Open Web-based Multiagent Systems. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, 1999.

[22] M. Winikoff. Implementing Commitment-based Interactions. In E. Durfee, M. Yokoo, M. Huhns, and O. Shehory, editors, *AAMAS*, pages 873–880, 2007.

[23] P. Yolum and M. P. Singh. Reasoning about Commitments in the Event Calculus: An Approach for Sepcifying and Executing Protocols. *Annals of Math. and AI*, 42(1–3):227–253, 2004.

Information-Driven Interaction-Oriented Programming: BSPL, the Blindingly Simple Protocol Language

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
singh@ncsu.edu

ABSTRACT

We present a novel approach to interaction-oriented programming based on declaratively representing communication protocols. Our approach exhibits the following distinguishing features. First, it treats a protocol as an engineering abstraction in its own right. Second, it models a protocol in terms of the information that the protocol needs to proceed (so agents enact it properly) and the information the protocol would produce (when it is enacted). Third, it naturally maps traditional operational constraints to the information needs of protocols, thereby obtaining the desired interactions without additional effort or reasoning. Fourth, our approach naturally supports *shared nothing* enactments: everything of relevance is included in the communications and no separate global state need be maintained. Fifth, our approach accommodates, but does not require, formal representations of the meanings of the protocols. We evaluate this approach via examples from the literature.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; D.2.1 [**Software Engineering**]: Requirements Specifications—*Methodologies*; H.1.0 [**Information Systems**]: Models and Principles—*General*

General Terms

Theory, Design

Keywords

Business process modeling, business protocols

1. INTRODUCTION

Interaction-oriented programming or IOP is concerned with the engineering of systems comprising two or more autonomous and heterogeneous components or *agents*. Such systems arise commonly in IT applications such as cross-organizational business processes and scientific collaboration. The key idea of IOP is that treating interactions as first-class concepts helps create systems whose participating agents can be independently designed and operated, with correctness judged largely based on their effective participation in the specified interactions. We model interactions as communications (messages sent by one agent to another), even though in some cases they may involve physical actions such as delivering a package or controlling an ocean glider. We specify interactions abstractly as arising between *roles*, each role a discrete conceptual party instantiated by one or more agents.

A *protocol* is a specification of (presumably, conceptually cohesive and suitably structured) communications among two or more agents that neglects the internal reasoning of the agents involved [3]. Two aspects of a protocol are relevant: (i) *operations*, to do with message occurrence and order; and, (ii) *meaning*, to do with the (business) import of the messages. Traditional approaches capture the operations procedurally but disregard the meaning. Procedural approaches are generally over-specified, rigid, and difficult to maintain, but yield obvious directives for agents and, hence, can be easy to realize. The declarative operational approaches support asserting operational constraints in logic, and offer increased clarity and flexibility. However, neither kind of approach handles the challenges of distributed computing well, especially to determine correct local enactments.

In contrast, a *business protocol* is one where we primarily or exclusively state the meanings of its messages and only indirectly any operational constraints on them [12]. Emphasizing meaning improves flexibility and maintainability. However, the meaning-based approaches rely upon a suitable characterization of the operations to unambiguously assign meanings to communications. Thus, existing approaches, whether procedural or declarative, end up specifying interactions in terms of the allowed orderings of messages.

Contributions. Our main motivation is to provide a simple declarative foundation for the operational underpinnings of IOP. We propose a novel declarative approach that (i) simplifies the operational details and (ii) cleanly separates operations from meanings, yet supports specifying meanings overlaid on operations. We claim that our approach is extremely simple: it has only two main constructs: (i) defining a message schema and (ii) composing existing protocols. Accordingly, we have dubbed it *BSPL*, the *Blindingly Simple Protocol Language*. BSPL states no constraints on the ordering or occurrence of messages, deriving any such constraints from the information specifications of message schemas. It treats interaction as first class and supports protocol compo-

Cite as: Information-Driven Interaction-Oriented Programming: BSPL, the Blindingly Simple Protocol Language, Munindar P. Singh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 491-498.

sition at its core. BSPL forces specifiers to be clear about the essential constraints, thereby helping them exclude spurious constraints that plague traditional approaches. It supports clear well-formedness conditions under which the specified protocols can be shown to be enactable. We show how BSPL captures a variety of common and subtle protocols.

Organization. Section 2 identifies some of the key principles that guide our approach. Section 3 introduces BSPL, including its motivation, syntax, intended semantics and several examples illustrating specification and composition of protocols. Section 4 suggests how a suitable semantics of BSPL may be formalized. Section 5 compares BSPL with two existing approaches based on protocols from the financial domain, demonstrating its advantages. Section 6 discusses related work and some directions for future research.

2. PRINCIPLES OF BSPL

The following principles distinguish our approach to IOP.

Information orientation. In information systems, agents interact with each other because they wish to obtain or convey information. Thus we specify each protocol as involving not only two or more roles but also one or more *parameters*, which stand in for the actual information items to be exchanged among the agents playing the roles during enactment.

Explicit causality. The flow of causality is reflected in the flow of information: there are no hidden flows of causality because there are no hidden flows of information. Indeed, if there were any hidden flows, then the very idea of protocols as a basis for IOP would be called into question.

No state. We need no global repository of state. All the relevant information affecting the notional *social state* of an interaction is explicit within the protocol—in the values of the parameters of the messages exchanged. No agent's private business logic is relevant, which is a key motivation for IOP.

Separating structure from meaning. The structure of a protocol is completely characterized by the names of the entities in its specification; the meaning relies upon the values exchanged during enactment. A protocol ought not to constrain the values directly since realizing such constraints would depend upon the implementations of agents. Instead, we would place any relevant constraints in the meaning layer, usually in terms of the commitments of the participants [12].

Putting the above observations together leads us to an extensive treatment of parameters. Parameters on messages are obvious, but we expand them to apply on protocols generally. Crucially, we adorn each parameter of a protocol with a specification of whether the parameter must be an input or an output of the protocol. Importantly, the adornments of the parameters are interpreted with respect to the protocol itself, not with respect to any role of the protocol. Thus, the adornment ⌜in⌝ means that the associated parameter must be instantiated so as to enact the protocol, i.e., "exogenously" or "externally" to the protocol. And, ⌜out⌝ means that the associated parameter must be instantiated in the course of enacting the protocol—i.e., it is instantiated "endogenously" or "internally" to the protocol. Consider a quote message as part of a price discovery protocol that includes an item description and a price, and may be sent in response to a request for quotes for a particular item. Clearly, for the quote message to be sent, its sender must instantiate all of its parameters. However, from the standpoint of the protocol, the item description is provided from outside the protocol and the price is provided by the protocol to the outside. Thus we would adorn the item description with ⌜in⌝ and the price with ⌜out⌝.

As indicated above, parameters and their adornments apply not only to individual messages but to entire protocols. For example, we might model *Shipping* as involving a parameter address with adornment ⌜in⌝. This indicates that *Shipping* does not determine the address, but must be "told" it. Conversely, the *Window Shopping* protocol (involving roles SHOPPER and STOREFRONT) would have parameters what and howmuch, each with an adornment ⌜out⌝. This indicates that *Window Shopping* would yield information about what the STOREFRONT is selling for how much.

Sometimes, the interaction explicitly demands a flow of information. For example, it is not possible for a bank to transfer funds without knowing how much and to what account. Therefore, a transfer message must follow a request that specifies the amount and the account. At other times, we may impose an ordering for "conventional" reasons. For example, although payment and delivery may occur concurrently once the item and price are determined, we may impose an ordering arbitrarily. For example, in a fast-food restaurant the customer pays first and in a traditional restaurant the customer pays at the end. We capture such conventions in our causal ordering by explicitly introducing suitable parameters, e.g., we may model the *payment* message as having a token adorned ⌜out⌝ that is adorned ⌜in⌝ on *delivery*. Another example of a convention is where a buyer must show a proof of age before buying an alcoholic drink. We can introduce a parameter ageProof to handle this case.

Our information orientation leads us to make three crucial assumptions, all based on treating a protocol as a conceptual entity or relation [5] whose instances or tuples are its enactments. Protocols are meant to be instantiated multiple times. For example, many agents would use *Purchase* to buy and sell many items, yielding a distinct instance (tuple) for each enactment. (In passing, we note that although we talk of relations here, practical BSPL settings would often use XML and parameters might be bound to XML documents.)

Uniqueness. We introduce the notion that some or all of a protocol's parameters define a *key*, which characterizes the expected uniqueness of the enactment: at most one enactment instance may occur one per key binding.

Integrity. Analogous to NOT NULL constraints on relations, each required public parameter in a complete protocol enactment must be bound. Otherwise, that means the enactment viewed as a tuple is incomplete.

Immutability. Each enactment viewed as an entity instance is immutable. That is, the parameters can be bound multiply often, but the different bindings arise in different enactments. Immutability provides robustness against asynchrony, because it ensures bindings are never ambiguous or out of date.

Together, the above assumptions delimit an enactment: it must proceed sufficiently to generate at least one tuple of parameter bindings; any additional duplication of parameter values is superfluous. This provides a general, principled basis for termination that contrasts both with procedural approaches (explicit enumeration of terminal states) and meaning-based approaches (termination as achieving a requisite meaning state, such as having no pending unconditional commitments). The latter treats termination the same as never beginning, which is operationally false. Further, it rejects protocols whose main purpose is to create unconditional commitments: thus it would formulate a negotiation protocol as producing conditional commitments, but then all or nearly all of its states would be terminal.

Protocols may be composed [3]. For example, we may define *Purchase* as a composition of *Order*, *Payment*, and *Shipping*. A fundamental tenet of our approach is that we make no distinction between individual or composite protocols. A composite protocol is expressed, assigned a formal semantics, and enacted the same way as any other protocol. The only difference might be that the constituent protocols generally exist before the design episode and the composite protocols are created during the design episode. As we remarked above, a message is the unit of interaction. Thus a single message is an atomic protocol.

The main benefit of composing protocols is to facilitate the reuse of designs and implementations by supporting multiple ways to compose the same protocols. Further, protocol composition offers a principled basis for adapting cross-organizational process models to capture evolving requirements, as in Desai et al.'s [2] approach.

Each protocol name is unique within our universe of discourse. Each protocol defines a scope (a unique namespace) within which its roles, parameters, and messages are also uniquely named. The roles and parameters of a protocol identify its public interface. In logical terms, a role acts as a kind of parameter, but semantically, they are quite different: a role corresponds to an executing agent whereas a parameter corresponds to a data item.

3. BSPL: LANGUAGE AND PATTERNS

Based on the foregoing, we define a protocol as consisting of exactly one message schema (template) or of the composition of two or more protocols. Although this is mathematically satisfactory, for practical convenience, we define the syntax of BSPL in a somewhat more conventional manner. The following is the syntax along with brief explanations. Superscripts of $+$ and $*$ indicate one or more and zero or more repetitions, respectively. Below, $\lfloor$ and $\rfloor$ delimit expressions, considered optional if without a superscript. For simplicity, we state cardinality restrictions informally.

For readability, we include some syntactic sugar tokens. In listings, we write reserved keywords in small sans serif, capitalize role names, and write parameters in camel case. In the text, we write message and protocol names *Slanted*, roles in SMALL CAPS, and parameters in sans serif. We insert ⌜ and ⌝ as delimiters, as in ⌜Self ↦ Other: hello[ID, name]⌝.

L_1. A protocol declaration consists of exactly one name, two or more roles, one or more parameters, and one or more references to constituent protocols or messages. A nilable parameter, ignored here for brevity, can remain unbound. All the parameters marked key (cannot be nilable) together form the key of this declaration.

Protocol ⟶ *Name* { role *Role*$^+$
parameter ⌊*Parameter*⌊key|⌊nilable⌋⌋$^+$ *Reference** }

L_2. A reference consists of the name of a protocol appended by the same number of roles and parameters as in its declaration. At least one of the parameters of the reference must be a key parameter of the declaration in which it occurs: this ensures the enactments of the reference relate to those of the current declaration.

Reference ⟶ *Name* (*Role*$^+$ *Parameter*$^+$)

L_3. Alternatively, a reference is a message schema, and consists of exactly one name along with exactly two roles, one or more parameters (at least one a key parameter of the declaration), and optionally a meaning.

Reference ⟶ *Role* ↦ *Role* : *Name* [*Parameter*$^+$]
⌊means *Expression*⌋

L_4. Each parameter consists of a name and an optional adornment.

Parameter ⟶ ⌊*Adornment*⌋ *Name*

L_5. An adornment is generally either ⌜in⌝ or ⌜out⌝. It may be ⌜nil⌝ in a reference to indicate that the adorned parameter is unknown, which can be crucial in some cases.

Adornment ⟶ in | out | nil

Now, we introduce a series of examples of BSPL as a way to informally describe its semantics. We omit the means clauses in our development but revisit them in Section 6.

3.1 Simple Protocol Declarations

Listing 1 demonstrates BSPL to define the simple *Pay* protocol. *Pay* consists of two roles and one message consisting of two parameters, ID and amount. We adorn both parameters ⌜in⌝ to indicate that they must be known (supplied) to enact *Pay*. In other words, a multiagent system must enact *Pay* in combination with some one or more other protocols that determine ID and amount. Because ID is the key, at most one payment may be made for a given ID value.

Listing 1: The *Pay* protocol.

```
Pay {
 role Payer, Payee
 parameter in ID key, in amount

 Payer ↦ Payee: payM[in ID, in amount]
}
```

Listing 2 shows *Offer*, which serves as a way to generate a price offer. Notice that both item and price are adorned ⌜out⌝, indicating that *Offer* would compute these parameters endogenously. Here, the BUYER generates item and the SELLER generates price, since these parameters are adorned ⌜out⌝ in messages to be sent by these roles. Thus, *Offer* can generate the amount (if identified with price) needed to enact *Pay*. Conversely, ID is adorned ⌜in⌝, meaning that *Offer* can only be used in combination with another protocol in which ID (suitably renamed if necessary) is adorned ⌜out⌝.

Listing 2: The *Offer* protocol.

```
Offer {
 role Buyer, Seller
 parameter in ID key, out item, out price
```

```
 Buyer ↦ Seller: rfq[in ID, out item]
 Seller ↦ Buyer: quote[in ID, in item, out
     price]
}
```

Listing 3 shows *Order*, which as formalized here repeats the entire *Offer*. Section 3.2 shows how to avoid such redundancy through composition. Notice that Listing 3 includes a parameter rID, which is adorned ⌜out⌝ in the protocol's interface as well as in the *accept* and *reject* messages. Each of these placements has an important ramification. First, if we were to omit rID from *Order*'s interface, its enactments would complete as soon as *quote* was sent, because all its parameters would then be bound. In other words, the enactment would complete prematurely. BSPL does not support separately requiring *accept* or *reject*, because doing so would violate the principle of explicit causality. Second, the presence of rID in both *accept* and *reject* indicates mutual exclusion of those two messages: simply because a parameter cannot be bound more than once in any enactment.

Listing 3: The *Order* protocol.

```
Order {
 role B, S
 parameter in ID key, out item, out price, out rID

 B ↦ S: rfq[in ID, out item]
 S ↦ B: quote[in ID, in item, out price]

 B ↦ S: accept[in ID, in item, in price, out rID]
 B ↦ S: reject[in ID, in item, in price, out rID]
}
```

3.2 Composing a Protocol

The power of protocols in modeling arises from the fact that they can be readily composed [3]. BSPL makes no distinction between a protocol that happens to be composed and one that is not. Indeed, each message can be viewed as a protocol in its own right. Listing 4 expresses the message ⌜From ↦ To: aMessage[in one, out two]⌝ as a protocol.

Listing 4: A message viewed as a protocol.

```
Message-as-Protocol {
 role From, To
 parameter in one key, out two key

 From ↦ To: aMessage[in one, out two]
}
```

Taking the same idea further, Listing 5 expresses *Order* as a composition of *Offer* and two protocols corresponding to the other messages defined in Listing 3. Even if we changed the role and parameter names, Listings 3 and 5 would remain semantically identical.

Listing 5: *Order* expressed as a composition.

```
Order {
 role B, S
 parameter in ID key, out item, out price, out rID

 Offer(S, B, in ID, out item out price)
 acceptProt(B, S, in ID, in item, in price, out
     rID)
 rejectProt(B, S, in ID, in item, in price, out
     rID)
}
```

3.3 More on Parameters in Protocols

Parameters are crucial to BSPL. We adorn the parameters not only in a declaration but also in each reference, including the individual messages. In general, such adornments are essential for capturing any constraints on what parameter bindings to propagate in what direction.

Listing 2 demonstrates two important well-formedness requirements on protocols. One, a parameter that is adorned ⌜in⌝ in a declaration must be ⌜in⌝ throughout its body. For brevity, we may sometimes omit such ⌜in⌝ adornments, but such parameters can take no adornment other than ⌜in⌝. Two, a parameter that is adorned ⌜out⌝ in the declaration must be ⌜out⌝ in at least one reference. At run time, at most one reference with an ⌜out⌝ adornment for a parameter may be enacted: thus such references are mutually exclusive.

If a parameter is adorned in a protocol declaration P, then any reference to P must apply the same adornment to that parameter. Otherwise, it would not be clear what propagation was appropriate. But we can leave some or all of P's parameters unadorned in a declaration or reference, thus signifying that propagation in both directions is permissible for that declaration or reference. We can think of this as the in–out adornment. When we refer to P from another protocol declaration, we may choose adornments for any of such unadorned parameters of P as a simple way to disambiguate the direction of information propagation.

Importantly, a top-level protocol declaration—one that stands alone and is ready to be enacted—must adorn all its parameters ⌜out⌝. Another way to think of this is as follows. For enactment, every parameter adorned ⌜in⌝ must have its value supplied through some other protocol, such as a message to one of the enacting agents, which would indicate that the given protocol omits relevant communications and therefore is not enactable in itself.

3.4 Common Specification Patterns

We now present some examples demonstrating the main concepts and typical usage of BSPL.

3.4.1 Duplicating a Parameter

Sometimes a role that needs to obtain a parameter binding might not be receiving it. Listing 6 shows how the originator of the binding can send a duplicate copy to another role. Because *Duplicating* has an ⌜in⌝ parameter, it is not enactable by itself. Here, we presume the ORIGINATOR produces a prior message in which aParameter is adorned ⌜out⌝.

Listing 6: *Duplicating* a parameter.

```
Duplicate-Parameter {
 role Originator, Consumer
 parameter in aParameter key

 Originator ↦ Consumer: share[in aParameter]
}
```

3.4.2 Generating an Identifier

A consequence of the information basis of BSPL is that the correctness of a protocol depends upon its keys. To facilitate composition in multiple contexts, it is convenient to define protocols that adorn an identifying parameter as ⌜in⌝, which means that such protocols cannot be enacted standalone. Listing 7 shows a simple protocol that generates an identifier, which can thus drive other protocols.

Listing 7: Generating an identifier.

```
Generate-Identifier {
 role Authority, Subject
 parameter out ID key

 Authority ↦ Subject: announce[out ID]
}
```

3.4.3 Local Parameters

A *local* parameter occurs within a protocol declaration but is not exposed in its public interface. Often, such a parameter may be essential for carrying out the desired interaction and thus would be included in underlying messages, but may not feature as an essential public interface of a protocol. Listing 8 shows a variant of *Purchase* in which the destination address is computed and used, but not deemed relevant for exposing from the overall interaction. Hence, if we were to refer to this *Purchase* variant from another declaration, we would not be able to refer to address.

Listing 8: Variant of *Purchase* with hidden address.

```
Purchase {
 role B, S, Shipper
 parameter in ID key, out what, out howmuch

 Order(B, S, in ID, out what, out howmuch)
 Decide-Address(B, S, in ID, out address)
 Ship(S, Shipper, in ID, in what, in address)
}
```

A local parameter must be adorned ⌜out⌝ in exactly one reference and ⌜in⌝ in all the rest. Hiding a parameter has consequences on the semantics. Because only the public parameters become part of the public interface, uniqueness applies only to tuples constructed from the public parameters. Thus, although the local parameters may take on multiple values, they would have no direct effect on the outcome as defined by the protocol. Consequently, a designer should hide only the parameters that are irrelevant to the intended outcome of the interaction, and should expose all the others.

3.4.4 Standing Offer

This is a common business situation where we need to generate multiple messages tied to the same standing offer. Desai et al. [2] describe such a situation in the insurance domain but, lacking a proper treatment of parameters, cannot formalize it. Once an insurance policy is created, it forms a standing offer: the insurance VENDOR would process howsoever many claims the SUBSCRIBER makes. Listing 9 shows how BSPL can naturally accommodate such a protocol.

Listing 9: The *Insurance Claims* protocol (from [2]).

```
Insurance-Claims {
 role Vendor, Subscriber
 parameter out policyNO key, out reqForClaim key,
     out claimResponse

 Vendor ↦ Subscriber: createPolicy[out
     policyNO, out details]
 Subscriber ↦ Vendor: serviceReq[in policyNO,
     out reqForClaim]
 Vendor ↦ Subscriber: claimService[in
     policyNO, in reqForClaim, out
     claimResponse]
}
```

Each claim refers to a unique policy and has a unique response; one policy may lead to multiple claims. Hence, we make policyNO and reqForClaim jointly the key. If necessary, we can include additional parameters to describe the policy, including its termination, in greater detail. The remaining protocols given by Desai et al. [2] involve simpler structures, such those demonstrated in the preceding sections.

3.5 Subtle Specification Patterns

The following patterns demonstrate the power of BSPL.

3.5.1 Flexible Sourcing of out Parameters

Listing 10 shows *Buyer or Seller Offer*, in which either the BUYER or the SELLER may generate the price. This protocol illustrates the distinction between a parameter being endogenous to a protocol versus being generated by one or another of the agents playing its roles. *Buyer or Seller Offer* involves two variants of *rfq* and *quote*, with differences in adornments of their parameters. We overload the message names since informally the names relate to meaning: the same commitment would be associated with a *quote* whether the price was ⌜in⌝ or ⌜out⌝ in it. We could equally well use different names. Notice that the interface of *Buyer or Seller Offer* is the same as that of *Offer* (Listing 2) since it has the same roles and parameters (with the same adornments).

Listing 10: The *Buyer or Seller Offer* protocol.

```
Buyer-or-Seller-Offer {
 role Buyer, Seller
 parameter in ID key, out item, out price

 Buyer ↦ Seller: rfq[in ID, out item, nil price]
 Buyer ↦ Seller: rfq[in ID, out item, out price]

 Seller ↦ Buyer: quote[in ID, in item, out
     price]
 Seller ↦ Buyer: quote[in ID, in item, in price]
}
```

In Listing 10, both *quote* variants rely upon the BUYER having provided item. As a result, the BUYER speaks first. The BUYER may announce the price or not, by choosing the appropriate variant of *rfq*. The two variants of *rfq* are mutually exclusive because they have incompatible adornments for price: thus at most one of them can be sent. Likewise, the two variants of *quote* are mutually exclusive. In essence, the choice is the BUYER's and the SELLER follows along. This is the reason we introduce the ⌜nil⌝ adornment on price in *rfq*. Upon receiving a ⌜nil⌝ price, the SELLER would not be able to send *quote* without generating the price locally.

3.5.2 in-out Polymorphism

Let us consider *Flexible Offer*, which can apply both where the price is exogenous (supplied) and where it is endogenous (computed by the protocol). We do so by omitting the adornment on price. Then, as Listing 11 shows, we need to provide alternatives so that each of the possible adornments of price is enactable.

If a reference to *Flexible Offer* adorns price ⌜in⌝, the only possible enactment is when B sends an *rfq* specifying the price to S, who responds with a *quote*. Alternatively, if a reference adorns price ⌜out⌝, B must send an *rfq* without specifying the price to S, who responds with a *quote* that specifies the price. That is, price is determined either from the reference (when it is referenced as ⌜in⌝) or by S (when it is referenced as ⌜out⌝). And, qID helps ensure that the enactment remains incomplete until *quote* occurs.

Listing 11: *Flexible Offer*: price as in or out.

```
Flexible-Offer {
 role B, S
 parameter in ID key, out item, price, out qID

 B ↦ S: rfq[ID, out item, nil price]
 B ↦ S: rfq[ID, out item, in price]

 S ↦ B: quote[ID, in item, out price, out qID]
 S ↦ B: quote[ID, in item, in price, out qID]
}
```

Listing 12 defines *Offer* as a simple variant of *Flexible Offer*. It illustrates a way to restrict the adornments of parameters without changing the logical structure of protocols. Even if we do not adorn the parameters of *Flexible Offer* as referenced from within *Offer*, there is no ambiguity, because those parameters are adorned in the declaration of *Offer*. Thus, when *Flexible Offer* is enacted, the parameters would be treated as in the declaration of *Offer* (qID is not used).

Listing 12: *Offer* as a restriction on *Flexible Offer*.

```
Offer {
 role Buyer, Seller
 parameter in ID key, out what, out howmuch

 Flexible-Offer(Buyer, Seller, in ID, out what,
     out howmuch, out qID)
}
```

3.6 Specification Patterns Hinting at Meaning

These patterns demonstrate connections with meaning.

3.6.1 Forwarding a Copy

Listing 13 shows a simple protocol that can be used to forward a parameter from one role to another. This protocol is often needed to help make a protocol enactable where a necessary parameter binding would not otherwise be known to a specified role. Notice that the functioning of this protocol relies upon meaning, namely, to ensure that the value of copy equals the value of original.

Listing 13: Forwarding a parameter value.

```
Forward {
 role From, To
 parameter in original key, out copy

 From ↦ To: forward[in original, out copy]
}
```

3.6.2 Mixed Initiative

Listing 14 shows a protocol that supports either role taking the initiative. This protocol is inspired by the formalization of the Enhanced NetBill by Yolum and Singh [12]. In this protocol, the BUYER and the SELLER can exchange as many messages as they like with the SELLER repeatedly sending *quote* messages and the BUYER *accept* messages. Each of them has the initiative and can work independently of the other. If necessary, we can combine mixed initiative with polymorphism, as introduced in Section 3.5.2.

Listing 14: The *Mixed Initiative Offer* protocol.

```
Mixed-Initiative-Offer {
 role B, S
 parameter in ID key, out qID key, out aID key
     out qItem, out qPrice, out aItem, out aPrice

 S ↦ B: quote[in ID, out qID, out qItem, out
     qPrice]
 B ↦ S: accept[in ID, out aID, out aItem, out
     aPrice]
}
```

The meaning layer would capture that the quoted and accepted items and prices are equal. Each message would correspond to the creation of a suitable commitment to provide a specified item if paid a specified amount and to pay a specified amount if provided a specified item. Assuming each party needs a commitment from the other in order to proceed, progress will occur only when they produce their respective commitments for the same item and price. This example shows that though BSPL captures the operational aspects in a declarative manner, it seeks neither to obstruct appropriate meaning nor to substitute for meaning.

3.6.3 Digressions

Yolum and Singh [12] introduced the idea of a digression where an agent may interact differently from a protocol for some steps but later resynchronize with it. Digression applies primarily to enactments rather than to protocols, although it facilitates the refinement of protocols. BSPL naturally supports digression. As long as the parameter adornments are satisfied, a digression has no impact on the enactment of a protocol. Digressions in the sense of Yolum and Singh depend upon a notion of meaning.

4. SKETCH OF A SEMANTICS FOR BSPL

We give an account of how BSPL protocols may be enacted and how to determine their distributed enactability using some mathematical concepts but, for brevity, without any mathematical notation.

A protocol describes an interaction by specifying messages to be exchanged between specific roles, and by imposing a partial order on the messages. An enactment of a protocol involves each of its roles being adopted by an agent, and the agents exchanging messages that the protocol specifies. Therefore, we capture the semantics of a protocol in terms of the enactments it allows. A message instantiates a message schema and is precisely described by its name, sender, receiver, and bindings for each of its parameters.

We define a *history* of a role as a sequence of messages, in each of which the role is either the sender or the receiver. Thus the history captures the local view of an agent who might adopt the role during the enactment of a protocol.

We define a *history vector* for a protocol as a vector each of whose elements is the history of a role mentioned in the protocol. A history vector is *quiescent* provided every message present in a sender's history is also present in its receiver's history. The fundamental causality constraint of distributed computing applies: a (receiving) role's history may contain a message reception only if the (sending) role's history contains the corresponding message emission [8]. However, we need a more sophisticated treatment of causality that captures the nature of parameters in BSPL.

The history of a role maps naturally to its local state. Notice we are interested in the local view of the public interactions, *not* in the internal state of an agent playing this role. Each message sent or received progresses the local state of the role, expressed in terms of the bindings of the parameters that the role knows. Specifically, a message emission is *viable* for a role if the role knows the bindings of all ⌜in⌝

parameters in that message and does not know the bindings of the ⌜out⌝ and ⌜nil⌝ parameters. In essence, it must produce the bindings for the ⌜out⌝ parameters, which it then knows (for future messages). A message reception is always viable and changes the state of the knowledge of the role, affecting the viability of future messages. A history vector is *viable* provided it arises from viable message emissions and receptions by the roles, i.e., by growing their local histories.

Informally, the *intension* of a protocol is given by the set of quiescent viable history vectors that enact it. The intension for a message is the set of quiescent viable history vectors in which it occurs. The intension of a composite protocol is the set of all viable interleavings of the history vectors in the intensions of its references. Only a protocol with an empty set of public ⌜in⌝ parameters may be enacted.

To understand when an enactment is correct, consider two references that occur within the same declaration and involve one or more common parameters, and consider their respective adornments of such a common parameter.

out–in indicates an ordering conflict: a message with ⌜out⌝ (even if nested in a reference) must precede, and the binding must propagate, to a message with ⌜in⌝.

nil–in **or** nil-out indicate a knowledge conflict and as such only apply to the same role: once a role sends or receives a message with ⌜out⌝ or ⌜in⌝, it cannot send a message with ⌜nil⌝.

out–out indicates an occurrence conflict: at most one of the references may occur anywhere in the system.

Our semantics addresses ordering conflicts through causality and knowledge conflicts through each role's view. For occurrence conflicts, there is no general solution, but we can analyze a BSPL specification to make sure that the same role controls which of the conflicting references occurs.

5. EVALUATION: CASE STUDY

We consider foreign exchange transactions, as formalized by Desai et al. [1]. *Bilateral Price Discovery* or *BPD* involves a TAKER sending a *priceRequest* to a MAKER, who responds with a *priceResponse*. Each message specifies a number of parameters, which for clarity we reduce to two parameters: query and result. Listing 15 shows the strikingly simple BSPL formalization of *BPD*.

Listing 15: The *Bilateral Price Discovery* protocol.

```
BPD {
 role Taker, Maker
 parameter out reqID key, out query, out result

 Taker ↦ Maker: priceRequest[out reqID, out
     query]
 Maker ↦ Taker: priceResponse[in reqID, in
     query, out result]
}
```

Desai et al. identify constraints under which a *priceResponse* message may not occur. These complicate Desai et al.'s specification, but Listing 15 captures them naturally: (1) "No priceRequest with a matching reqID has happened" (BSPL: adorn reqID (and query) as ⌜in⌝ on *priceResponse* and as ⌜out⌝ on *priceRequest*); (2) "A priceResponse with identical parameters has happened" (BSPL: automatic since repetitions are superfluous); (3) "A priceResponse with the same ID but a different result ... is happening simultaneously" (BSPL: mark reqID as a key); and (4) like #3 above but for other messages (BSPL: handle as above).

Consider Desai et al.'s [1] discussion of *multilateral price discovery (MPD)*, in which a TAKER interacts with a MAKER via an intermediary EXCHANGE. Intuitively, it makes sense that MPD is a composition of BPD with itself. Odell et al. [10] informally discuss the concept of *nesting* in AUML, wherein an agent playing a role in one protocol may participate in additional protocols in the middle. In Odell et al.'s terms, the EXCHANGE would be a MAKER in one copy of BPD and nest the second copy of BPD. There are two shortcomings with Odell et al.'s nesting. First, it draws a false hierarchy between two protocols, placing one as subservient to the other, whereas the interactions are conceptually peers. Two, and more fundamentally, nesting is a matter of how an agent is implemented. For all that anyone knows, even in the plain BPD, a MAKER might be shopping for deals in the background, possibly acting as a TAKER in another copy of BPD. But such internally driven behaviors are not public interactions and thus are not part of the given protocol.

Desai et al. [1] offer a better solution than nesting by explicitly composing BPD with itself to produce MPD. They assert data flow axioms whereby a query parameter in one copy of *priceRequest* is passed to the second copy, and likewise in the reverse direction for the result from *priceResponse*. However, Desai et al.'s approach violates encapsulation: it opens up each copy of BPD so as to enable stating constraints on the constituent messages of each copy in order to compose them as desired.

In BSPL, MPD can be expressed in a remarkably simple manner. Listing 16 uses in-out polymorphism (Section 3.5.2) to define a *Generalized BPD* or GBPD, in which query and res are not adorned. We can produce a specification of BPD equivalent to Listing 15 exactly as Listing 12 defines *Offer*.

Listing 16: *Generalized Bilateral Price Discovery*.

```
GBPD {
 role T, M
 parameter reqID key, query, res

 T ↦ M: priceRequest[out reqID, out query]
 T ↦ M: priceRequest[in reqID, in query]

 M ↦ T: priceResponse[in reqID, in query, out
     res]
 M ↦ T: priceResponse[in reqID, in query, in
     res]
}
```

Next, Listing 17 specifies MPD as an almost trivial composition of GBPD with itself. The adornments of the parameters in the two references to GBPD are different, and ensure that the composition is correct. Notice that the encapsulation is not broken (the GBPD declaration is not revealed here) and we are not specifying the internals of any role.

Listing 17: *Multilateral Price Discovery*.

```
MPD {
 role Taker, Exchange, Maker
 parameter out reqID key, out query, out res

 GBPD(Taker, Exchange, out reqID, out query, in
     res)
 GBPD(Exchange, Maker, in reqID, in query, out
     res)
}
```

6. DISCUSSION

What do we gain from an interaction-oriented approach wherein protocols are first-class entities? Although an agent-oriented approach, which focuses on the roles, is more familiar, it limits the modeling unnecessarily. By focusing on interactions, we can capture constraints from a public, as opposed to a role, perspective. In particular, when roles are introduced during composition, such new roles would automatically view any relevant constraint as satisfied.

Although we suppress the means clauses above, BSPL is geared toward providing the undergirding for any effective treatment of meanings. Listing 18 shows an example based on Listing 9 to give the reader a flavor of a means clause. Here C indicates a commitment [2, 12], and the expression states that the VENDOR commits to providing claim service to the SUBSCRIBER whenever the SUBSCRIBER sends a request under the specified claim. The benefit of BSPL here is that the operational basis for the meanings in terms of causality and information is taken care of automatically.

Listing 18: Meaning for *Insurance Claims*.

```
Insurance-Claims { ...

 Vendor ↦ Subscriber: createPolicy[out
     policyNO] means C(Vendor, Subscriber,
     serviceReq[policyNO, reqForClaim],
     claimService[policyNO, reqForClaim,
     claimResponse])
}
```

6.1 Literature

Increasing recognition of the importance of interaction has led to work on *choreographies* [11], which too capture the operational aspects of protocols as studied here. However, a choreography is typically specified procedurally, usually in a language such as message-sequence charts (MSCs) [6] or an analogous notation, such as WS-CDL [11].

AUML [10] is an important notation for protocols (many of its features were assimilated into UML 2.0). AUML's sequence diagram notation takes a strong procedural stance for describing interactions. Thus, it emphasizes explicit constraints on how messages are ordered. In contrast, our parameter adornments force clarification of the arrow of causality, making it correspond to the flow of information.

Recently, Miller and McGinnis [9] proposed $\mathcal{RASA}$, a language for protocols based on the proposition dynamic logic. Some of this language refers to agent reasoning and some to interaction. BSPL can capture the latter parts of it. In particular, in BSPL, iteration arises from the possible bindings of a protocol's parameters, and is limited only by the size of the cross-product of the domains of the key parameters. And, our semantics limits choice to guarded choice. $\mathcal{RASA}$ describes first-class protocols, i.e., those that an agent can inspect and reason about. BSPL, in addition, treats protocols as a first-class modeling concept for ready composition.

Desai and Singh [4] identify several challenges to the enactability of a protocol. BSPL avoids all the ordering problems they identify as varieties of *blindness*, because the only way to capture an ordering constraint in BSPL is to do so in a causally sound way: from a reference with an ⌜out⌝ adornment of a parameter to a reference with an ⌜in⌝ adornment of the same parameter. The problematic enactments cannot arise. The well-known problem of *nonlocal choice* [7] arises when correct behavior by a role depends on actions of another role. BSPL does not automatically avoid nonlocal choice. However, we can analyze a BSPL specification to determine that it is not at risk of nonlocal choice.

Traditional work on service composition primarily considers orchestrations where a conceptually central party controls two or more services. A strength of this work lies in its formalization of service behaviors and in its use of planning and constraint reasoning to construct appropriate service compositions. Although our present setting is quite different, we imagine that many of the techniques of service composition may be expanded and applied in our setting.

6.2 Future Work

A useful direction would be enhancing the treatment of the information model. For instance, it might be appropriate to entertain multiple keys for a protocol. Further, it would be useful to understand how important properties such as enactability may be verified in a compositional manner. Some natural extensions to BSPL that we will be considering include (1) a principled treatment of multicast, where multiple agents playing the same role receive a message and (2) accommodating discovery protocols, where the roles are bound late during enactment.

Acknowledgments

This work was partially supported by the OOI Cyberinfrastructure program, which is funded by NSF contract OCE-0418967 with the Consortium for Ocean Leadership via the Joint Oceanographic Institutions. Thanks to Matthew Arrott, Amit Chopra, and Kohei Honda for helpful discussions.

7. REFERENCES

[1] N. Desai, A. K. Chopra, M. Arrott, B. Specht, and M. P. Singh. Engineering foreign exchange processes via commitment protocols. In *SCC*, pp. 514–521, 2007.

[2] N. Desai, A. K. Chopra, and M. P. Singh. Amoeba: A methodology for modeling and evolution of cross-organizational business processes. *ACM TOSEM*, 19(2):6:1–6:45, Oct. 2009.

[3] N. Desai, A. U. Mallya, A. K. Chopra, and M. P. Singh. Interaction protocols as design abstractions for business processes. *IEEE TSE*, 31(12):1015–27, 2005.

[4] N. Desai and M. P. Singh. On the enactability of business protocols. In *AAAI*, pp. 1126–1131, 2008.

[5] R. Elmasri and S. Navathe. *Fundamental of Database Systems*. Benjamin Cummings, second edition, 1994.

[6] ITU. Formal description techniques (FDT)—Message Sequence Chart (MSC). Document Z.120, Apr. 2004.

[7] P. B. Ladkin and S. Leue. Interpreting message flow graphs. *Formal Aspects Comput.*, 7(5):473–509, 1995.

[8] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *CACM*, 21(7):558–565, 1978.

[9] T. Miller and J. McGinnis. Amongst first-class protocols. In *ESAW 2007*, *LNCS* 4995, pp. 208–223. Springer, 2008.

[10] J. Odell, H. V. D. Parunak, and B. Bauer. Representing agent interaction protocols in UML. In *AOSE 2000*, *LNCS* 1957, pp. 121–140. Springer, 2001.

[11] WS-CDL. Web Services Choreography Description Language. www.w3.org/TR/ws-cdl-10/, Nov. 2005.

[12] P. Yolum and M. P. Singh. Commitment machines. In *ATAL 2001*, *LNAI* 2333, pp. 235–247. Springer, 2002.

On Topic Selection Strategies in Multi-Agent Naming Game

Wojciech Lorkiewicz
Swinburne UT
Melbourne, Australia
wlorkiewicz@swin.edu.au

Ryszard Kowalczyk
Swinburne UT
Melbourne, Australia
rkowalczyk@swin.edu.au

Radoslaw Katarzyniak
Wroclaw UT
Wroclaw, Poland
radoslaw.katarzyniak@pwr.edu.pl

Quoc Bao Vo
Swinburne UT
Melbourne, Australia
bvo@swin.edu.au

ABSTRACT

Communication is a key capability of autonomous agents in a multi-agent system to exchange information about their environment. It requires a naming convention that typically involves a set of predefined names for all objects in the environment, which the agents share and understand. However, when the agents are heterogeneous, highly distributed, and situated in an unknown environment, it is very unrealistic to assume that all the objects can be foreseen in advance, and therefore their names cannot be defined beforehand. In such a case, each individual agent needs to be able to introduce new names for the objects it encounters and align them with the naming convention used by the other agents. A language game is a prospective mechanism for the agents to learn and align the naming conventions between them. In this paper we extend the language game model by proposing novel strategies for selecting topics, i.e. attracting agent's attention to different objects during the learning process. Using a simulated multi-agent system we evaluate the process of name alignment in the case of the least restrictive type of language game, the naming game without feedback. Utilising proposed strategies we study the dynamic character of formation of coherent naming conventions and compare it with the behaviour of commonly used random selection strategy. The experimental results demonstrate that the new strategies improve the overall convergence of the alignment process, limit agent's overall demand on memory, and scale with the increasing number of the interacting agents.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent systems, Languages and structures

General Terms

Experimentation

Keywords

cognitive agent, language emergence, naming game

Cite as: On Topic Selection Strategies in Multi-Agent Naming Game, W. Lorkiewicz, R. Kowalczyk, R. Katarzyniak, B. Vo, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 499-506.

1. INTRODUCTION

Language is an extensively used everyday tool, as it allows individuals to gain, share and utilise information in a social setting. It is also a key capability of autonomous agents that facilitates the exchange of information and enables collaboration in a multi-agent system. As such, language constitutes the collective adaptation to the changing circumstances of the environment and advances the performance of certain social tasks.

Conveying information about the state of the environment, i.e. communication, requires that agents share a set of predefined names for all of the perceivable objects. However, it is very unrealistic to assume that all of the objects can be foreseen in advance, and that all of the required names can be defined and shared beforehand. Therefore, all agents need to develop from scratch, and further sustain, their individual names for all of the perceived objects.

In principle, word learning is a rather simple task of mapping linguistic labels onto a set of pre-established concepts [2]. However, the problem is far more complex in a multi-agent setting, as any differences in individual mappings, i.e. naming conventions, result in miscommunication between interacting agents. As such, agents not only need to develop and sustain their individual names, but most importantly need to align them to form a coherent shared naming convention. In particular, each autonomous agent, through a series of consecutive interactions with other agents, needs to align its private linguistic mappings. As such, a multi-agent system comprised of communicating individuals can be considered as a 'complex adaptive system' [17] that collectively solves the problem of developing a shared communication system.

Despite several studies [4, 14, 18, 21, 22] the problem of language alignment is still an active area of research [13, 18, 20]. Moreover, language game model [7, 8, 18, 19] defines a prospective mechanism for agents to learn and align their naming conventions. In this paper we introduce a novel approach of agent's attention orienting, i.e. topic selection strategies (see section 4) in language game, and evaluate its impact on the process of name alignment. Using a simulated multi-agent system, formalised in section 3, we study the dynamics of the language alignment process in the case of no feedback naming game[1]. Incorporating the adaptive cross-situational learning scheme [8], in section 5 we study the dynamics of the emergent process against different topic selection strategies that are utilised by the speakers. We show how a proper modification of the topic selection strategy may improve the overall convergence of the alignment process, limit the overall demand on memory, and scale properly with the increasing number of agents.

[1]No feedback naming game [19] is a type of language game [18] (see section 2)

Finally, in section 6, we analyse the mechanism underlining the observed behaviour and conclude the paper in section 7.

2. BACKGROUND AND RELATED WORK

The general problem of language alignment is fundamental to the field of multi-agent systems, especially embodied multi-agent systems. For instance, incorporating a flexible semantic communication system into a smart sensors network [5, 10, 23] may lower system's energy utilisation and extend its operation time. However, the most significant, and most appealing is the incorporation of language alignment mechanism in robotic systems [13, 16, 18].

To focus the attention, let us assume a group of spatially distributed mobile agents operating in an unknown, highly dynamic, spacious and possibly adversarial territory, similarity to settings proposed in [12] or [6]. All autonomous agents are embodied, situated (physically bounded) and distributed, all are capable of basic manoeuvring and all share a common task of monitoring the environment. Each individual is wandering around the environment collecting valuable information and occasionally engaging in interaction with the nearby agents. Depending on it's current intentions the character of such interaction may differ, from basic mutual identification, through information exchange, to a complex coordinated action. Nevertheless, vast majority of these interactions involve linguistic communication, where language facilitates the interplay between cognitive agents [3]. In order to convey information about objects from agent's sight, for instance to align the attention of interacting agents and focus on a particular object from the surroundings, agents must exchange meaningful symbols.

Obviously, in order to focus attention on the exact same object the population must share and utilise a certain naming convention, i.e. shared form of language[2]. A simple solution would involve an arbitrary label-object mappings that are predefined in the agents. However when the environment is unknown at the design time, a coherent naming convention cannot be build-in and shared by all the agents beforehand. Moreover, due to natural restrictions following from the embodiment, i.e. limited range, interface errors and costly long range communication, neither any explicit central coordination approach, nor any global communication scheme are suitable. Additionally, as each individual is an equally valuable source of information, a single 'leader' agent cannot be directly imposed on the system. Thus, agents are required to develop their naming convention from scratch, and are restricted to a local ad-hoc communication, i.e. linguistic interaction involving only the nearby members of the population and concerning only a relatively local set of encountered objects.

As the coherent naming convention cannot be build-in at the design time, each agent needs to be equipped with an internal mechanism of name acquisition. In principle, allowing the agent to introduce new names for unknown objects. However, due to distributed character of the population, there is a high chance that a certain object is labelled differently by multiple agents. As such, introducing competing labels and increasing the miscommunication between agents. In order to reduce the number of conflicting words, the agents should be capable of altering their label-object mappings and form coherent associations within the entire population. Unfortunately due to local ad-hoc communication restriction, each agent has only limited knowledge about the naming conventions utilised by others. In principle, only the occasional interactions between agents provide valuable insights on the general population naming stance. Nevertheless, as the available information is very narrow and highly limited, a simple approach towards alignment cannot guarantee that multiple agents will eventually agree on a shared object-label mappings. For instance, as an individual hearing a new word may presumably assign it to an infinite number of objects in its sight, leading to indeterminacy of meaning [15].

In fact, developing a mechanism that would lead to a coherent formulation of names among interacting individuals is not a trivial task [11]. Several approaches have been proposed and investigated in the literature [4, 22, 21, 18, 14], ranging from associative types of memory [18], through genetic algorithm models [21], to neural network adaptation [14]. The most promising approach addressing the aforementioned problem is the language game model (LGM) [18], where a population of agents thrives to develop a shared set of associations between signs and meanings, using communicative acts. LGM offers a general framework for modelling the possible emergence of language and formulates basic settings for linguistic interaction between agents. It assumes that each agent has its own, strictly private and individually emerged, word-object associations (names) that are stored in an associative type of internal memory - lexicon. In particular, as the lexicons are private, they may differ between agents resulting in naming conflicts that occur during interaction.

The idea behind the language game is that through a series of routine pair-wise interactions, the agents can align their lexicons reaching a coherent state of the entire population. In naming game, type of language game specified by LGM, a single interaction is described as a simple interplay between two agents, one acting as a speaker, and the other as a hearer. The speaker agent selects a single object from its sight and names it, according to its internal naming convention. Whilst the hearer uses the heard utterance as a clue to identify which of the objects was intended by the speaker. Depending on the feedback the agents receive after the game, and assuming that agents are equipped with a pre-developed pointing mechanism, three basic types of naming games can be identified [19]. In the simplest case, both agents receive feedback, as the hearer points to the intended interpretation, and as the speaker points to the intended topic. In the case of limited feedback, only the speaker receives feedback, as only the hearer points to the intended interpretation. In the no-feedback case, neither the speaker nor the hearer receive additional feedback after the game leaving both agents clueless about the results of their interaction. It should be noted that in the simplest case, the hearer is able to precisely deduct speaker's intended mapping between the name and the object. Whereas, the absence of pointing procedure significantly increases the hearers uncertainty, as all objects in its sight are equally probable topics, resulting in indeterminacy of meaning.

Basic properties of the alignment process were studied in most favourable types of environments and population settings, focussing mainly on the simplest (feedback based) case of Naming Game [18]. Using a straightforward cross-situational learning (CSL) mechanism embodied agents were able to learn the naming conventions based solely on co-variances that occur across different situations. In [19] it is shown that in a multiple objects setting the CSL is hard to properly scale-up with the increasing number of agents, and it is hard to reach proper coherence among the agents. As such, the early procedures were extended to incorporate additional mechanism of synonymy reduction [7] and homonymy damping [8] leading to a substantial improvement in their performance. The former, introduced additional notion of word utilisation, as a word score resembling the frequency of its successful usages, whilst the latter approach, introduced an adaptive alignment mechanisms, i.e. intelligent cross-situational learning (ICSL). In addition to regular enforcement and inhibition rules that steer the population of interacting agents to coherent word-meaning mappings, ICSL preserves the relative differences between concurring words that allow it to outperform other existing approaches in zero feedback naming

[2]Throughout this paper, language is perceived as a complex adaptive system [17] that can be represented as a weighted complete bipartite graph (See section 3)

game settings [8].

The extensive literature studies,including most recent summaries in [18, 20], show that despite its popularity the LGM has been investigated only in a limited set of basic settings[3], where uniform world structures, random attention orienting strategies, one-step pair-wise interaction pattern are assumed. As such in this paper we investigate the effect of introducing non classical attention orienting strategies, i.e. topic selection strategies in the LGM. We argue that a rational strategy should reflect agent's internal character and it's individual intentions, and not just uniformly sample agent's current sight, as in the existing formulations.

3. GENERAL MODEL

We introduce the formal model of the investigated case, and begin by formalising the state of the multi-agent system as a 4-tuple $S(t)$, as follows

Definition 1. For each time point $t \in T = (t_1, ..., t_{K_T})$ a system state is a tuple:

$$\bar{S}(t) = \langle O(t), X_O(t), P(t), X_P(t) \rangle$$

- set of identifiable objects $O(t) = (o_1, ..., o_{K_O(t)})$
- context random process $X_O(t)$
- population $P(t) = (a_1, ..., a_{K_P(t)})$ of agents
- interaction random process $X_P(t)$.

The system state resembles a general state of the entire multi-agent system in a given point of time. It depicts currently identifiable objects O, currently operational agents P, and defines the externally imposed processes, i.e. the model of dynamic environment X_O (available through context) and the model of agent interaction X_P. As such, at each discrete time point t the random process X_O models the current state of the environment that is available to the system. Each agent $a \in P(t)$ perceives a certain part of its local environment - context $X^a_O(t)$ - as a set of objects in it's sight $\forall_{a \in P(t)}\ X^a_O(t) \subset O$. Analogous, the random process X_P for every time point t models the set of currently interacting agents, i.e. $X_P(t) \subset P^{K_I(t)}$, where $K_I(t)$ is the number of interacting agents.

In the assumed settings the context size is fixed $\forall_{t \in T, a \in P(t)}$ $\|X^a_O(t)\| = c$, and the interaction is limited to a single pair-wise $\forall_{t \in T} X_P(t) \in P(t) \times P(t)$ pattern. In the most general case, the set of identifiable objects and the set of all agents in the population can change during the system lifetime, however we assume a simpler case where both the set O and P are finite and static, i.e. $\forall_{t \in T} O(t) = O \wedge P(t) = P$.

3.1 Agent

An agent is the most fine-grained autonomous entity present in the system. It is embodied in the environment and is a part of the interacting population. In order to communicate, the agent needs to be equipped with an appropriate semantic infrastructure, that can be defined as the agent's state, as follows:

Definition 2. Agent's $a \in P(t)$ state in a given system state $\bar{S}(t) = \langle O(t), X_O(t), P(t), X_P(t) \rangle$ is a tuple:

$$\bar{A}(t) = \langle Ob^a(t), W^a(t), \mathcal{L}^a(t), \phi^a_P, \phi^a_I, \theta^a, \psi^a \rangle$$

- set of identified objects $Ob^a(t) = (o^a_1, ..., o^a_{K_{a,Ob}(t)}) \subseteq O$,
- set of words $W^a(t) = \{(w^a_1, s^a_1), ..., (w^a_{K_{a,W}(t)}, s^a_{K_{a,W}(t)})\}$,
- lexicon mapping $\mathcal{L}^a(t) : W^a(t) \times Ob^a(t) \rightarrow [0, 1]$
- interpretation function $\phi^a_I(t) : W^a(t) \times \mathcal{L}^a(t) \rightarrow Ob^a(t)$,
- production function $\phi^a_P(t) : Ob^a(t) \times \mathcal{L}^a(t) \rightarrow W^a(t)$,
- topic selection function $\theta^a(t) : 2^{Ob^a(t)} \rightarrow Ob^a(t)$,
- update function $\psi^a(t) : W^a(t) \times 2^{Ob^a(t)} \times \mathcal{L}^a(t) \rightarrow \mathcal{L}^a(t)$.

Each object represents a self contained invariant in the external environment that is available to agent's perception and that encapsulates the smallest indivisible entity available to its higher processes. As the precise formulation of agent's perception is outside of the scope of this paper we assume that for each agent an object is explicitly identified by a unique and strictly internal identifier ($i \sim o^a_i$). Research in [8] assumed a static and fixed set of objects, we extend their settings allowing the agent to gradually build up the set of known objects Ob^a as it encounters them in the environment.

Words, on the other hand, are external representations identified by the population as dedicated communication signs. Each signal $w^a_j \in W^a$ is associated with agent's subjective notion of usability $s^a_j \in [0, 1]$ denoting its individual estimate of strength of a word spread in the population. The set of words that the individual uses is iteratively build up by the agent, as new words are invented by the speaker whenever it lacks a proper word for a given topic, and are incorporated by the hearer whenever it hears an unknown word.

In terms of linguistic capabilities the most important part of the agent is its lexicon, i.e. the mapping $\mathcal{L}^a$ that represents actual correlation $\sigma^a(o, w) \in [0, 1]$ between objects $o \in Ob^a$ and words $w = (w^a_i, s^a_i) \in W^a$. The higher it is the more definite the agent is that a certain word is an adequate name for an object. As such, the lexicon encapsulates the current state of agent's language, that for convenience can be viewed as a weighted complete bipartite graph $L^a = (V^a, E^a, \sigma^a)$, where $V^a = W^a \cup Ob^a$ is the set of vertices, $E^a = W^a \times Ob^a$ is the set of edges, and $\sigma^a(w, o)$ is the weight of an edge (w, o). Each agent is then able to interpret external utterance w^a_i, i.e. select the most adequate object o based on its current state $L^a(t)$, and produce the external utterance w^a_i, i.e. the most adequate name for a given object o based on its current lexicon state (see section 3.2). As such, the actual graph structure modulates agent's interpretation and production scheme. In particular, agent's two ϕ^a_P and ϕ^a_I schemes reflect certain method of traversing the lexicon graph, i.e. proper selection of the edges according to the current distribution of weights.

We further assume the well established mechanism of interpretation and production [8]. The interpretation scheme is rather straight forward, as for a given word $w = (w^a(t), s^a(t)) \in W^a(t)$ the interpretation function ϕ^a_I selects the edge $(w, o) \in E^a$ with the maximum weight ($\phi^a_I(w, L^a(t)) = argmax_{o_i} \sigma^a(o_i, w)$), and thus interprets w as referring to o. On the other hand, the production scheme assumes that the speaker before uttering a name evaluates its subjective reflection of the population, by considering the usability s of each possible word w. As such, for a given object o the production function ϕ^a_P selects the edge $(w, o) \in E^a$ with word w having the highest usability from all the words that the agent is able to interpret as referring to the object o ($\phi^a_P(o, L^a(t)) = argmax_{w_i}\{w_i : o = \phi^a_I(w_i, L^a(t))\}$), and thus names o.

3.2 Interaction

Interaction between agents is the only opportunity for an individual to verify the appropriateness of its language, and it is the only way to gain additional information about the naming conventions utilised by others. In the assumed settings, the interaction is governed by the means of no feedback naming game routine, where at each time point $t \in T$ a random pair of agents $X^a_P(t) = (a_S(t), a_H(t))$ where $a_S(t) \neq a_H(t)$ (a_S - speaker, a_H - hearer) advances in a simple communication. The speaker selects a single object $o_T(t)$ as the topic of conversation, according to its topic selection strategy $o_T(t) = \theta^{a_S}(X^{a_S}_O(t))$ and current context $o_T(t) \in X^{a_S}_O(t)$. Further, the speaker names the intended topic $w_T(t) = \phi^{a_S}_P(o_T(t), L^{a_S}(t))$, based on its current lexicon state

[3]For the sake of completeness, we note the research in [1], where different population structures were investigated in a minimal naming game (single object environment).

and utilising its production function $\phi_P^{a_S}$. Next, the uttered word is transmitted to the hearer, that receives it along with the current context of perception $X_O^{a_H}(t)$. It is assumed that the topic of the utterance is shared among both contexts, i.e. $o_T(t) \in X_O^{a_S}(t) \cap X_O^{a_H}(t)$. Based on this information, i.e. the context and the associated uttered word, the hearer updates its lexicon $L^{a_H}(t) = \psi^{a_H}(w_T(t), X_O^{a_H}(t), L^{a_H}(t-1))$, and interprets the utterance $o_I(t) = \phi^{a_H}(w_T(t), L^{a_H}(t))$[4]. As agents do not receive feedback concerning the outcomes of the game, the interpreted meaning and the heard word pair $(w_T(t), o_I(t))$ is regarded as the most probable one. As such agent's subjective notion of usability $s_T(t)$ of the heard word $w_T(t)$ should be increased, whilst the usability of all concurring names $\{w_i : o_I(t) = \phi_I^{a_H}(w_i, L^{a_H}(t))\}$, i.e. all other names that can be interpreted as the identified object $o_I(t)$, should be decreased. Moreover, learning from co-occurrence between words and objects (cross-situational learning) implies that after each interaction the hearer updates its lexicon $L^a(t)$ by modifying the correlations $\sigma^{a_H}(o, w_i)$.The update function ψ^{a_H} dampens the correlation $\sigma^{a_H}(o, w_i)$ between the received word w_i and currently not perceived objects $o \notin X_O^{a_H}(t)$, and enforces the correlation between the received word w_i and currently perceived objects $o \in X_O^{a_H}(t)$, while the correlations with other words remain unchanged. In settings involving context with multiple objects, a single interaction is typically insufficient to determine the utilised naming convention, as presumably all objects from the context are equally probable intended meanings. We note, that an object can dominate the correlation between a certain word only if it occurred, with this word, more times then with any other object.

3.3 Measures

In order to formulate differences in the dynamics of the alignment process, we identify two major axes of comparison, i.e. coherence and word statistics, and focus on the evolution of language in the assumed multi-agent system. We study the behaviour of the system based on four measures: success rate, language coherence rate, average number of used and overall number of words.

The most obvious measure is the frequency of successful communications between agents. It resembles the observed ability of the system to transfer information from one agent to another, and as such it allows to reason about the utility of the communication system itself. In order to keep track of the effectiveness of the communication we calculate the success rate μ_{SR}, as follows:[5]

$$\mu_{SR(N)} = \sum_{t \in T|_N} \mathcal{I}_{\{o_T(t) = \phi_I^{a_H}(\phi_P^{a_S}(o_T(t), L^{a_S}(t)), L^{a_H}(t))\}} \quad (1)$$

In general, the success rate $\mu_{SR(N)}$ of order N is the frequency of successful communications in the last N interactions $(T|_N)$, i.e. successful in terms of that both agents focus on the same object (1). In isolation, despite its simplicity, this measure is not very useful, as it does not take into account all objects from the environment, and can be easily deformed. For instance, agents communicating only about a single object are able to reach highest possible success rates, as they might share a common name for the preferred object, despite having poor coherence between other names.

Due to the above restrictions, we need to formulate additional measure resembling the naming convention spread among the entire population and reflecting the coherence of names among all existing objects. As such, we introduce language coherence μ_{LC}, as the probability that two randomly selected agents assign the same name for a randomly selected object from the environment, as follows:

$$\mu_{LC} = E_{a,a' \in P, a \neq a', o \in O}[\phi_I^a(\phi_P^{a'}(o, L^{a'}), L^a) = o] \quad (2)$$

The lowest possible coherence, i.e. $\mu_{LC} = 0$, reflects a state of no language coherence in the system, as there are no two agents that use the same name for any of the objects. The highest possible coherence, i.e. $\mu_{LC} = 1$, represents the state of full coherence, where all agents share the same naming conventions. It should be noted that in the assumed settings a system is absorbed by the coherent state, as from this point all of the utterances are consistent with the observed context, and without any external disturbance all of the strongest associations remain strongest.

In order to analyse the characteristics of the emergent language we keep track of the number of used words μ_{UW}, and keep track of the total number of all invented words μ_{TW}, defined as follows:

$$\mu_{UW} = E_{a \in P}[\|\{w \in W^a : \exists_{o \in O} b^a \sigma^a(w, o) > 0\}\|] \quad (3)$$

$$\mu_{TW} = E_{a \in P}[\|W^a\|] \quad (4)$$

The former, is calculated as the average, over all agents, number of positively associated words, and it resembles the stability of current associations. As the optimal communication system has one-to-one mappings between words and objects, i.e. the same number of used words as the number of existing objects, and any deviation from this proportion reflects a potentially unstable situation, as miscommunication might occur. In the latter case, we calculate the overall number of existing words in the system, resembling again the stability of the communication system during its development. It should be stressed that new words may enter the lexicon, i.e. as agents are inventing new words, on regular basis. However, as it is not possible for a word w_j to leave the lexicon, the opposite mechanisms is a bit different, and a word can become disassociated through the dampening procedure, i.e. weight of associations $\sum_o \sigma(w_j, o)$ shared with w_j and/or usability s_j of w_j is close to 0. Nevertheless, the higher the number of different words in the system, the significantly higher is the number of all possible associations and possibly lower coherence. Moreover the higher the number of words, both used and invented, the more technically demanding the system is, as it needs more memory to store all associations and more processing power to cope with all possible association.

4. TOPIC SELECTION STRATEGIES

The most common strategy investigated in the literature is the purely random selection of topic, where a speaker uniformly samples its current context in order to select the intended meaning of its utterance. It is a rational approach in the presence of direct feedback, as the context degenerates into single object and different selection strategies do not affect the evolution of the system. However, in case of limited feedback and significant context sizes the topic selection strategy can significantly affect the overall evolution of the system (See section 5). We must underline the fact that, all of the following extensions relate only to the speaker. Moreover, as the hearer has no a priori knowledge about the strategy utilised by the speaker, we assume that it treats all utterances as a result of random sampling, and follows the behaviour described in section 3.

Different topic selection strategies can be analysed twofold, from the theoretical point of view and from a more pragmatic stance. The former approach assumes that selection is just a basic procedure of choosing a single object $o_T(t)$ from a set of objects $X_O(t)$. As such, the speaker agent a perceives current state of the environment as the context $X_O^a(t)$ of ongoing interaction, and in a predefined manner selects a single object $o_T(t)$ as the topic. From the more practical point of view, the topic selection strategy resembles the speaker's reaction to the recent state of the environment.

[4]If the intended meaning o_T is the same as the interpretation o_I then the game is considered successful, otherwise it is a failure.

[5]$\mathcal{I}$ is the identity function, i.e. $\mathcal{I}_{x=x} = 1$ and $\forall_{x \neq y} \mathcal{I}_{x=y} = 0$

As such, the context $X_O^a(t)$ is a form of short term memory that stores the most recent and most important objects, and depending on its internal perception the speaker agent a selects a single object $o_T(t)$ that it found valuable, interesting or significant. In this case topic selection strategy resembles the internal force that drives agent's attention, and orients its sensory receptors towards a particular object and away from other available stimuli [9]. Having this interpretation in mind, we formulate and introduce three basic topic selection strategies, and justify them as different points of attention that affect individual perception and cognition.

4.1 Random

As noted, the original model proposed in [8] assumes a purely random selection of topic, where at each time point the speaker is uniformly sampling its current context in order to select the topic of its utterance. In this case the topic selection function θ^{orig} (See equation 5) is a random variable with a uniform distribution over all objects in the context and can be defined as follows:

$$\forall t \in T \; Pr(\theta^{random}(X_O(t) = o)) = 1/\|X_O(t)\| \qquad (5)$$

This situation resembles the case where the attention of the agent is randomly focusing on different objects in the environment. As such all objects are equally valuable to the agent, and uttering the name of each one of them is of an equal importance to the speaker.

4.2 Min / Max

It is obvious that perception is usually not passive, and it is the individual that is actively looking or listening in order to see or hear [9]. Previous strategy assumes no direct force that is applied on agent's perception, i.e. the agent perceives the environment in purely passive manner. However, agent's focus should depend on both, agents past observations and the current state of the environment. As such, attention of a curious agent should be attracted by a new, or relatively unknown objects from the environment. Resulting in agent's significant tendency to speak about the least occurring, in past interactions, object. On the other hand, attention of a more stagnant agent should be attracted by already familiar objects, or relatively known, objects from the environment. Resulting in agent's tendency to select the most occurring, in its past interactions, object. In principle, both, i.e. min and max, approaches represent two similar forces that drive the attention of an individual. One is focusing on the least known elements of the environment (min), whilst the other on the most known elements of the environment (max).

We further assume that each agent $a \in P$ is able to store the frequencies $F^a(t) = \{f_1^a(t), ..., f_{K_{a,Ob}}^a(t)\}$ of the observed occurrences of the encountered objects in its past interactions. This basic statistics is further stored as agent's private knowledge about the environment, and is utilised in its future interactions to drive agents attention towards certain aspects of the environment. In this case the topic selection functions θ^{min} and θ^{max} are deterministic functions that for a given state of the environment select the most and least frequent object, respectively. In order to maintain the probabilistic notation we denote this deterministic functions as a random variable with a Dirac delta distribution as follows:

$$\forall t \in T \; Pr(\theta^{\max}(X_O(t) = o_i) = \begin{cases} 1, & f_i = \max_j f_j \\ 0, & \text{otherwise} \end{cases}$$

$$\forall t \in T \; Pr(\theta^{\min}(X_O(t) = o_i) = \begin{cases} 1, & f_i = \min_j f_j \\ 0, & \text{otherwise} \end{cases}$$

4.3 Preference

The point of attention of the system can also depend entirely on the internal structure of the agent, i.e. as the agent may have certain preferences over the objects, or as simply its perception may be attracted by certain objects. As such it is the embodied, i.e. physical properties of the perception apparatus, and the internal structure, i.e. pre-build preferences and biases, that has significant impact on agents orientation. For instance, being equipped with very sensitive microwave sensor the agent might have tendency to focus on objects that emit such wavelengths, and as such naturally tend to select them as the intended topics.

In this paper we assume that each agent has a predefined set of preferences $R^a(t) = \{r_i^a(t) : o_i \in Ob^a(t)\}$ over the objects. These preference values r can be understood as affordances, i.e. individual utility of an object as perceived by the agent. Without any loss of generality preferences can be treated as probabilities, where for every agent $a \in P$ following conditions hold $\sum_{r \in R} r = 1$ and $\forall_{r \in R} r \geq 0$. In such a case we can model the topic selection functions θ^{pref} (See equation 6) as a random variable with a discrete distribution over the objects defined by the preferences structure $R^a(t)$ as follows:

$$\forall t \in T \; Pr(\theta^{pref}(X_O(t) = o_i)) = r_i(t) \qquad (6)$$

5. EVALUATION

In order to evaluate different topic selection strategies we perform numerous simulations. All experiments share a common framework, and assume finite, static set of objects O and agents P, all incorporate a uniform interaction process[6] with a pair-wise communication model ($a_S(t) \neq a_H(t)$), and all are restricted to a shared context setting ($o_T(t) \in X_O^{a_S}(t) \cap X_O^{a_H}(t)$). Moreover, it is assumed that agent's behaviour is governed by a set of standard interpretation ϕ_I, production ϕ_P, and update ψ rules, as described in section 3. We investigate a number of simulation settings, including various population sizes, various object sizes and different context sizes using versatile measures, from basic success rate, to more complex synonymy and homonymy spread in the population. However, due to the space limitations we only focus on the general properties of the system, and present the obtained results as an exemplification of the observed system's behaviour.

Baseline parameters assume: ten agents, ten objects, fixed context size limited to two objects, and random selection strategy. All of the presented graphs are an average over fifty consecutive runs and as such are a good representation of the observed dynamic behaviour of the system. In order to compare the topic selection strategies it is important to guarantee the same experimental settings for each selection procedure by fixing the context path (sequence of randomly generated consecutive context) and interaction path (sequence of randomly generated consecutive agent pairs) before each run, and sharing it with all of the strategies.

5.1 Success rate and language coherence

Figure 1 depicts the typical character of language coherence dynamics. On the right column graphs, we can observe the slow phase shift dynamics of the coherence rate (See equation 2), reflecting three fundamental stages of system's evolution. Whilst, on the left column, we can observe the typical dynamics of the success rate (see equation 1).

Initial iterations form and maintain a plateau of low coherence, where the early invented words shape hooks that gradually begin to fill up agent's lexicons with words and cast fresh possible conventions (see section 5.2). Despite, the initial burst of new conventions and sudden increase of the overall usability of words s_i, the average strength of correlation σ is still relatively low. In the second phase the system undergoes a sudden increase of coherence. Due to a particular realisation of random processes X_O and X_P, some

[6]where each pair of agents is equally probable to interact

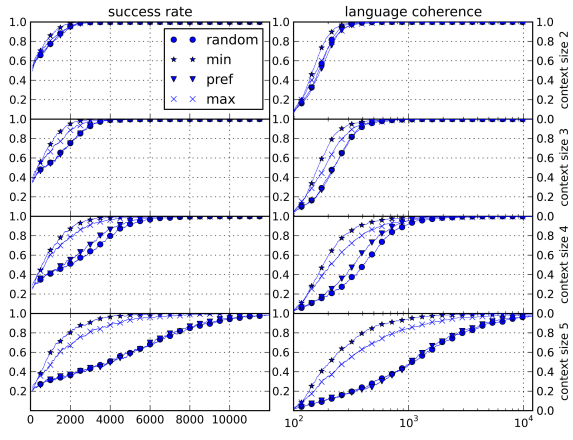


Figure 1: Success rate and language coherence in different topic selection strategies and four different context sizes.

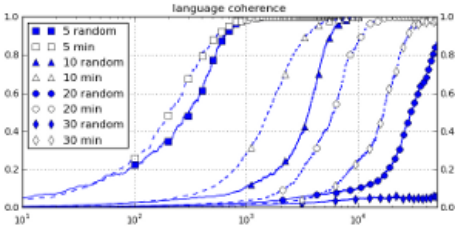


Figure 2: Language coherence in different topic selection strategies and three different population sizes (context size set to 4).

of the initial hooks are dampened, as such some of the words are no longer used (see figure 3), whilst the other ones are enforced, as such the overall strength of correlation of the used words increases. Resultant, the strongest words start to dominate the convention and can be easier shared among the individuals. The last stage resembles the significant slow down in alignment process. As most of the language is already shared by the agents, i.e. the coherence level is above $\mu_{LC} = 0.8$ (80% of the maximum coherence), and due to the random character of participant and context selection, the less probable, and still unaligned, cases must occur, i.e. the minority must adopt the dominant naming convention.

Three basic observations can be made from the obtained results (see section 6). At first higher levels of coherence are reached by the min(max) strategy, i.e. at each iteration it is higher compared to random strategy. Secondly, the more significant the context size is, the more significant is the observed disproportion. As observed in figure 2 analogous tendency is maintained with the increase of population size. Third, the min/max strategies seem to resemble very similar characteristics under the influence of changing context size.

5.2 Words statistics

Figure 3 depicts the typical dynamic character of the average number of words used by an individual (see equation 3) and the overall number of words present in the system (see equation 4). As already noted the system undergoes three distinguishable phases. Initially, as agents lexicons begun to fill up with words, and as agents still lack of precise information, a sudden increase in the number of used words is observed. Reaching its maximum at about the level of 20% of the maximum coherence (see section 5.1). Further, as some of the initial formed names, due to random character of the process, are more 'popular' they tend to dominate the population, and begin to systematically eliminate all other competing words from their usage. This early alignment results in the observed sudden decrease in the number of used words, and is correlated with the increase of language coherence (see section 5.1). In short, as agents begin to share more and more conventions, all of the most obviously incorrect ones, least 'popular', can be quickly dampen. Obviously, this decrease is less sudden then the initial burst, and it steadily diminishes with time. Again, the last stage resembles the significant slow down in the alignment process, as the minority must aligned to the dominant convention. Finally, the number of used words stabilises at the number of objects present in the environment, reaching as such the ideal one-to-one naming convention (see section 3.3).

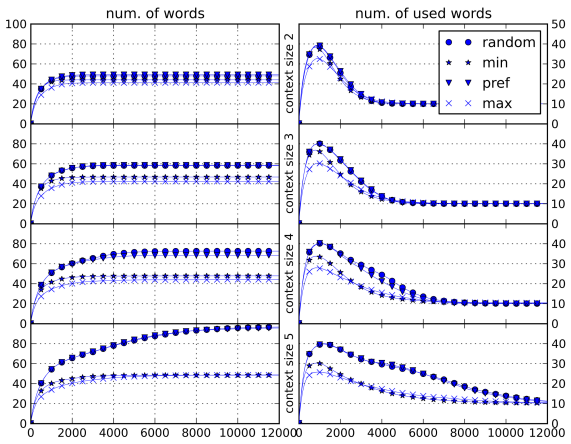


Figure 3: Number of words and used words in different topic selection strategies and four different context sizes.

Again three basic observations can be made from the obtained results. At first, the maximum number of words directly depends on the number of objects in the context and on the selection strategy. In case of random selection the increase of needed number of words, with the increase of context size, is significant, whereas the min/max strategies are more or less stable. Importantly, the min (max) strategy in all context sizes requires significantly less words, also less used words, then the random strategy. Secondly the more significant the context size is the less words are needed for the min/max strategies, and the more significant is the disproportion between min (max) strategy and the 'other' strategies. Third, the min/max strategies seem to resemble very consistent characteristics without any significant influence from the changing context size. The number of invented words is stable at around the same level (40) for both strategies, and for both strategies the maximum number of used words undergoes similar change, i.e. decreases with the increase of context size.

5.3 Dynamic context size

In all of the previous simulations a fixed context size settings were assumed, where $\forall_{t \in T} \|X_O(t)\| = c$. However, despite its analytical simplification it is still a significant limitation imposed on the system, as it requires that all interactions between agents involve a strictly predefined number of objects from the environment - $c \leq K_O$. Therefore, it is reasonable to ask how general is the observed behaviour, and whether it is not only restricted to a fixed context settings. In order to verify this notion, we introduce a modification to the previous settings and before interaction alter the number of objects present in the context. Introduced change is governed by a predefined probability distribution, i.e. $Pr(X_O(t) = c)$. In particular, as all objects are equally probable to appear in the

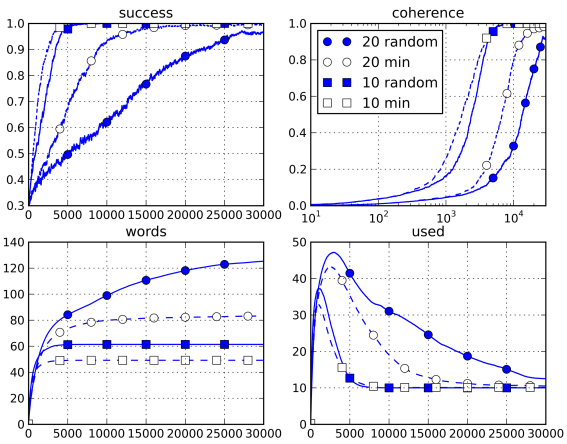


Figure 4: Varying context size settings.

context it seems rational to investigate an analogous type of distribution, where $Pr(X_O(t) = c) = \frac{1}{K_O}$. We should note that on average, as the number of interaction increases, each agent interacts equally often in each possible context size setting, e.g. equally often perceives a single object ($c = 1$), and equally often perceive the entire set of objects ($c = K_O$).

Figure 4 represents a typical behaviour of the alignment procedure in dynamic context settings. As in fixed context settings the min/max strategies result in higher coherence rates, however due to the aforementioned independent cases and the dynamic character of their change the increase is less significant. Nevertheless, still the min/max selection strategies require less words to reach a coherent naming convention, as such limit the required memory of the system and limit the number of used words in the system. Additionally, dynamic context sizes still maintain the scaling property, as with the increasing number of objects the difference between language coherence of random and modified selection is increasing. In short, it should be noted that the behaviour pattern of the alignment process in case of dynamic context sizes remains consistent with the observations for the case of fixed context structures.

6. ANALYSIS

We can recall that the considered learning mechanism is based solely on learning from co-occurrences between words and objects (cross-situational learning), and the more agents 'talk' about a certain object the more names are invented and more conventions tested. Moreover, the more different agents start to talk about the same set of objects the more possible naming conflicts might occur, i.e. more conflicts must be resolved and the names must concur with other numerous competitors.

At first, the noticeable differences between the random and min (max) strategy may be falsely attributed to the specificity of the assumed experimental settings. Seemingly, the limited number of agents increases the probability that multiple agents share similar statistics of the environment, i.e. similar private frequencies $F(t)$. As such, whenever speaker selects the least (or the most) occurring object the shared frequencies increases the chance that the hearer also perceives the topic as rare, and resultantly both agents tend to select similar objects. In particular, being in line with hypothetical specificity of the assumed settings requires that with the increasing number of agents the disproportion between min/max and random strategy should diminish. As this behaviour is not observed, i.e. the results presented in figure 2, it supports our justification that the presupposed similarity of perceptions between communicating individuals does not influence the tendency to dominate correlations.

The key to understand the significant difference between the random strategy and min/max strategies lies in the characteristics of the random process that each selection strategy represents. The fundamental probability that a given object o is present in the context is given as $p_o = Pr(o \in X_O(t)) = \frac{1}{K_O}$, and therefore the probability that a certain frequency increases is $Pr(f_i(t_2) > f_i(t_1)) = p_{o_i}$ (t_1 and t_2 indicate two consecutive time points when the agent interacted). Fixing a strategy, results in speaker's linguistic behaviour being governed by its selection θ process, that in case of min/max strategy is additionally modulated by the perceived statistics of the environment.

Let us consider an isolated (single speaker agent) process of random topic selection, at each time point a given number of objects c is drawn (without replacement) from a set containing K_O identifiable objects and put into a shared bin B, i.e. $Pr(o_i \in B) = \binom{K_O-1}{c-1} \cdot \binom{K_O}{c}^{-1} = \frac{c}{K_O}$. Further, a random object $i*$ is selected from the bin, i.e. $Pr(o_{i*} = \theta^{orig} | o_{i*} \in B) = \frac{1}{c}$. The resultant probability of object o_{i*} being selected is equal to $Pr(o_{i*} = \theta^{orig}) = Pr(o_{i*} = \theta^{orig} | o_{i*} \in B) \cdot Pr(o_{i*} \in B) = \frac{c}{K_O} \cdot \frac{1}{c} = \frac{1}{K_O}$. If the latter selection procedure is uniform, representing the random strategy, the initial distribution of objects is maintained, i.e. each object is equally probable to be selected as the topic. Based on this observation the expected number of times an object o was selected by the speaker $X_o^{a_S}(N)$ after N iterations is equal to $E[X_o^{a_S}] = N \cdot Pr(o = \theta^{orig})$, as the process X^{a_S} follows the multinomial distribution. As such, all objects are evenly selected by all agents, and significant number of naming conflicts occurs. This is in line with simulation results (See figure 2), where in early stage the number of concurring words increases drastically and the number of invented words is significant.

On the other hand in the case of min/max selection strategy the presented procedure must be extended to a case where for every drawn object o_i an identical one is added to a shared bin, whilst the original one is returned to the set. As such the number n_i of objects of type i in the shared bin constitutes the frequency of a certain object $f_i = n_i / \sum_j n_j$. Now if at each iteration the agent selects the bin with lowest / highest number of balls, then this process represents min / max strategy appropriately. Let us assume a simple case, where there are only two objects o_1 and o_2 present in the environment. At each iteration a single agent $a \in P$ along with one object o_i is randomly selected, increasing agent's a frequency of o_i occurrence f_i^a. Afterwards agent a selects a single object (o_1 or o_2) based on its current frequencies (f_1^a, f_2^a) and (min/max) strategy. After N iterations the probability that the frequency of occurrences is equal for both objects is $Pr(f_1^a(N) = f_2^a(N)) = \frac{1}{2}^N$, and it significantly decreases with the number of iterations. As $\theta^{min} = argmin_{o_j} f_j^a$ the probability that after N iterations the selection process is going to switch objects is equal to $Pr(f_1^a(N) = f_2^a(N)) Pr(\theta^{min}(N-1) \neq \theta^{min}(N+1)) = \frac{1}{2}^N \frac{1}{2}$. Obviously, with the increasing number of iterations the probability that the agent a used to select $o_1(o_2)$ will switch to $o_2(o_1)$ is decreasing exponentially, e.g. for $N = 10$ the probability of switching is .05‰, and is highly defined by the early realisation of the random selection. Resultantly, the agent has a strong preference over one of the objects (opposite to random selection). It should be noted that as agents do not share their private perceptions the frequencies differ between the individuals, and result in even distribution of preferences between agents, i.e. most likely the same number of agents will prefer o_1 as o_2. As such in case of min/max strategy, the population of interacting agents randomly transforms themselves into a population of individuals that tend to speak about

different parts of the environment, i.e. individuals that tend to have unique selection preferences. Whilst in the case of random strategy, the population of interacting agents resembles an opposite transformation into a group of individuals that tend to equally (in terms of frequency) speak about all parts of the environment. As such in case of min/max approach the agents need to invent less words (see section 5.2), i.e. on average less conflicts occur, and due to limited possibilities the higher coherence is easier to achieve (see section 5.1). Interestingly, as the context size increases the agent's preferences, selection strategy, tends to be more specialised and focusing on a single object. Therefore the observed decrease in number of needed words in increasing context sizes (see figure 3).

7. CONCLUSIONS

In principle, developing a mechanism that would lead to a coherent formulation of names among multiple interacting individuals is not a trivial task. Several approaches have been proposed and investigated in the literature, however, the language game model seems to be still the most significant framework for language emergence. Presented approach is in line with the ongoing research, as it extends the 'classical' LGM approach of random topic selection, and studies the dynamic character of the formation of coherent naming conventions. Using a simulated multi-agent system we give insights on the effects of different attention attracting procedures, i.e. topic selection strategies, in the case of the least restrictive type of naming game (without feedback).

The attention orienting strategies are an important aspect in the research on language emergence based on the language game model. In this paper we have introduced three general meta-models of different topic selection mechanisms, and studied their effects on the behaviour of no feedback naming game with significant contexts sizes. We justify that incorporation of different topic selection strategies influences the behaviour of the system, resulting in higher levels of language coherence and maintaining a the minimal memory requirements. Moreover, we show that the more significant the context size is the more significant is the observed disproportion between different strategies. In particular, we have shown that the 'classical' settings of random selection do not guarantee the best performance, and can be easily enriched through a more deterministic strategy. Higher levels of coherence can be reached by agents tending to select the best known objects (max strategy) or tending to select the least known objects (min strategy). Additionally, the more the agents in the population then again more significant is the observed disproportion between different strategies. As such, we show that min/max topic selection strategies scale significantly better then the extensively used random selection.

Our future research focuses on extending the proposed mechanism to a more flexible population structures and less restrictive environments. We further intend to introduce adaptation procedures that would allow to dynamically modulate agent's selection strategy, allowing to study more advanced and complex models of attention orienting.

8. REFERENCES

[1] A. Baronchelli, V. Loreto, L. DallAsta, and A. Barrat. Bootstrapping communication in language games: Strategy, topology and all that. In *Proceedings of the 6th International Conference on the Evolution of Language*, p. 11-18, 2006.

[2] P. Bloom. *How children learn the meanings of words*, volume 24. The MIT Press, 2002.

[3] A. Cangelosi. The grounding and sharing of symbols. *Cognition Distributed: How Cognitive Technology Extends Our Minds*, p. 83, 2008.

[4] A. Cangelosi and D. Parisi. *Simulating the evolution of language*. Springer-Verlag, NY, USA, 2002.

[5] D. Cook and S. Das. How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2):53-73, 2007.

[6] P. Corke, R. Peterson, and D. Rus. Localization and Navigation Assisted by Networked Cooperating Sensors and Robots. *The International Journal of Robotics Research*, 24(9):771-786, 2005.

[7] B. DeVylder and K. Tuyls. Towards a common lexicon in the naming game: The dynamics of synonymy reduction. *Workshop on Semiotic Dynamics of Language Games*, 2005.

[8] J. DeBeule, B. DeVylder, and T. Belpaeme. A cross-situational learning algorithm for damping homonymy in the guessing game. In *In proceedings of ALIFE X*, MIT Press., 2006.

[9] W. J. Freeman. The physiology of perception. 264:78-85, 1991.

[10] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, and S. Devlin. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing*, 5(3):236-252, 2009.

[11] W. Lorkiewicz and R. Katarzyniak. Issues on Aligning the Meaning of Symbols in Multiagent Systems. *New Challenges in Computational Collective Intelligence*, 217, Springer, 2009

[12] K. H. Low, J. M. Dolan, and P. Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, p. 23-30, 2008.

[13] M. Mirolli and S. Nolfi. Evolving communication in embodied agents: Theory, Methods, and Evaluation. *Evolution of Communication and Language in Embodied Agents*, p. 105-121, 2010.

[14] S. Nolfi. Emergence of communication in embodied agents: Co-adapting communicative and non-communicative behaviours. *Connection Science*, 17(3):231-248, 2005.

[15] W. Quine. *Word and object*. The MIT Press, 1960.

[16] I. Rekleitis. Distributed coverage with multi-robot system. *In Proceedings ICRA'06*, p. 2423-2429, 2006.

[17] L. Steels. Language as a complex adaptive system. In *Parallel Problem Solving from Nature PPSN VI*, page 17-26. Springer, 2000.

[18] L. Steels. Modeling The Formation of Language in Embodied Agents: Methods and Open Challenges. *Evolution of Communication and Language in Embodied Agents*, page 223-233, 2010.

[19] P. Vogt and H. Coumans. Investigating social interaction strategies for bootstrapping lexicon development. *Journal of Artificial Societies and Social Simulation*, 6(1):1, 2003.

[20] P. Vogt and B. De Boer. Editorial: Language Evolution: Computer Models for Empirical Data. *Adaptive Behavior*, 18(1):5, 2010.

[21] K. Wagner, J. a. Reggia, J. Uriagereka, and G. S. Wilkinson. Progress in the Simulation of Emergent Communication and Language. *Adaptive Behavior*, 11(1):37-69, 2003.

[22] *AAMAS'02*, p. 362-369, 2002. J. Wang and L. Gasser. Mutual online concept learning for multiple agents.

[23] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, and R. Handcock. Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas. *IEEE Pervasive Computing*, 8(1):30-36, 2009.

Game Theory and Learning

Reaching Correlated Equilibria Through Multi-agent Learning

Ludek Cigler
Ecole Polytechnique Fédérale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
ludek.cigler@epfl.ch

Boi Faltings
Ecole Polytechnique Fédérale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
boi.faltings@epfl.ch

ABSTRACT

Many games have undesirable Nash equilibria. For example consider a resource allocation game in which two players compete for an exclusive access to a single resource. It has three Nash equilibria. The two pure-strategy NE are efficient, but not fair. The one mixed-strategy NE is fair, but not efficient. Aumann's notion of correlated equilibrium fixes this problem: It assumes a correlation device which suggests each agent an action to take.

However, such a "smart" coordination device might not be available. We propose using a randomly chosen, "stupid" integer coordination signal. "Smart" agents learn which action they should use for each value of the coordination signal.

We present a multi-agent learning algorithm which converges in polynomial number of steps to a correlated equilibrium of a wireless channel allocation game, a variant of the resource allocation game. We show that the agents learn to play for each coordination signal value a randomly chosen pure-strategy Nash equilibrium of the game. Therefore, the outcome is an efficient correlated equilibrium. This CE becomes more fair as the number of the available coordination signal values increases.

We believe that a similar approach can be used to reach efficient and fair correlated equilibria in a wider set of games, such as potential games.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Economics

Keywords

Multiagent Learning, Coordination, Game Theory

1. INTRODUCTION

The concept of Nash equilibrium forms the basis of game theory. It allows us to predict the outcome of an interaction between rational agents playing a given game.

Cite as: Reaching Correlated Equilibria Through Multi-agent Learning, L. Cigler and B. Faltings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 509-516.

However, many games have undesirable equilibrium structure. Consider the following resource allocation game: Two agents are trying to access a single resource. Agents can choose between two actions: *yielding* (Y) or *accessing* (A). The resource may be accessed only by one agent at a time. If an agent accesses the resource alone, she receives a positive payoff. If an agent does not access the channel, her payoff is 0. If *both* agents try to access the channel at the same time, their attempts fail and they incur a cost c.

The payoff matrix of the game looks as follows:

	Y	A
Y	0, 0	0, 1
A	1, 0	$-c$, $-c$

Such a game has two pure-strategy Nash equilibria (NE), in which one player yields and the other one goes straight. It has also one mixed-strategy NE, where each player yields with probability $\frac{1}{c+1}$. The two pure-strategy NE are efficient, in that they maximize the social welfare, but they are not fair: Only one player gets the full payoff, even though the game is symmetric. The mixed-strategy NE is fair, but not efficient: The expected payoff of both players is 0.

In his seminal paper, Aumann ([1]) proposed the notion of *correlated equilibrium* which fixes this problem. A correlated equilibrium (CE) is a probability distribution over the joint strategy profiles in the game. A correlation device samples this distribution and recommends an action for each agent to play. The probability distribution is a CE if agents do not have an incentive to deviate from the recommended action.

In the simple game described above, there exists a CE which is both fair and socially efficient: just play the two pure-strategy NE with probability $\frac{1}{2}$. This corresponds to an authority which tells each player whether to yield or access the resource.

Correlated equilibria have several nice properties: They are easier to find (for a succinct representation of a game, in polynomial time, [11]) and every Nash equilibrium is a correlated equilibrium. Also, any convex combination of two correlated equilibria is a correlated equilibrium. However, a "smart" correlation device which randomizes over joint strategy profiles might not always be available.

It is possible to achieve a correlated equilibrium without the actual correlation device. Assume that the game is played repeatedly, and that agents can observe the history of actions taken by their opponents. They can learn to predict the future action (or a distribution of future actions) of the opponents. These predictions need to be *calibrated*, that is, the predicted probability that an agent i will play a cer-

tain action a_j should converge to the actual frequency with which agent i plays action a_j. Agents always play an action which is the best response to their predictions of opponents' actions. Forster and Vohra in [5] showed that in such a case, the play converges to a set of correlated equilibria.

However, in their paper, Foster and Vohra did not provide a specific learning rule to achieve a certain CE. Furthermore, their approach requires that every agent were able to observe actions of every other opponent. If this requirement is not met, convergence to a correlated equilibrium is not guaranteed anymore.

In this paper, we focus on a variant of the resource allocation game, a game of wireless channel allocation. In this game, there are N agents who always have some data to transmit, and there are C channels over which they can transmit. We assume that $N \geq C$. Access to a channel is slotted, that is, all agents are synchronized so that they start transmissions at the same time. Also, all transmissions must have the same length. If more than one agent attempts to transmit over a single channel, a collision occurs and none of the transmissions are successful. An unsuccessful transmission has a cost for the agent, since it has to consume some of its (possibly constrained) power for no benefit. Not transmitting does not cost anything.

We assume that agents only receive binary feedback. If they transmitted some data, they find out whether their transmission was successful. If they did not transmit, they can choose some channel to observe. They receive information whether the observed channel was free or not.

The game has several efficient (but unfair) pure-strategy Nash equilibria, in which a group of C agents gets assigned all the channels. The remaining $N - C$ agents get stranded. It has also a fair but inefficient mixed-strategy NE, in which agents choose the transmission channels at random. As in the resource allocation game, there exists a correlated equilibrium which is efficient and fair.

In this scenario, a global coordination device that would tell each agent which channel to transmit on is not available. Imagine that the agents are wireless devices belonging to different organizations. Setting up such a coordination device would require additional communication before the transmissions. Moreover, agents cannot observe all the actions of their opponents, since the feedback they receive is very limited. Therefore, they cannot learn the fair and efficient correlated equilibrium from the history of the play.

We propose a different approach to achieve an efficient and fair correlated equilibrium in such a game. We do not want to rely on a complex correlation device which needs to know everything about the game. Also, we do not want to rely on the history which may not be observable. Instead, we assume that agents can observe, before each round of the game, a randomly chosen integer from a set $\{0, 1, \ldots, K-1\}$. For each possible signal value, agents learn which action to take.

Our correlation signal does not need to know anything about the game. It does not have to tell agents which action to take. For example, the agents may just observe noise on some frequency. This is the principal difference from using the "smart" coordination device, which is assumed in the original definition of correlated equilibrium.

The main contributions of this work are the following:

- We propose a learning strategy for agents in the wireless channel allocation game which, using minimal information, converges in polynomial time to a randomly chosen efficient pure-strategy Nash equilibrium of the game.

- We show that when the agents observe a common integer correlation signal, they learn to play such an efficient pure-strategy NE for each signal value. The result is a correlated equilibrium which is increasingly fair as the number of available signals K increases.

The rest of the paper is organized as follows: In Section 2, we present the algorithm agents use to learn an action for each possible correlation signal value. In Section 3 we prove that such an algorithm converges to an efficient correlated equilibrium in polynomial time in the number of agents and channels. We show that the fairness of the resulting equilibria increases as the number of signals K increases in Section 4. Section 5 highlights experiments which show the actual convergence rate and fairness. In Section 6 we present some related work from game theory and cognitive radio literature, and Section 7 concludes.

2. LEARNING ALGORITHM

In this section, we describe the algorithm which the agents will use to learn a correlated equilibrium of the wireless channel allocation game.

Let us denote the space of available correlation signals $\mathcal{K} := \{0, 1, \ldots, K-1\}$, and the space of available channels $\mathcal{C} := \{1, 2, \ldots, C\}$. Assume that $C \leq N$, that is there are more agents than channels (the opposite case is easier). An agent i has a strategy $f_i : \mathcal{K} \to \{0\} \cap \mathcal{C}$ which it uses to decide which channel it will access at time t when it receives a correlation signal k_t. When $f_i(k_t) = 0$, the agent does not transmit at all for signal k_t. The agent stores its strategy simply as a table.

It adapts the strategy as follows:

1. In the beginning, for each $s \in \mathcal{K}$, $f_i(s)$ is initialized uniformly at random from $\mathcal{C}$.

2. At time t, if $f_i(k_t) > 0$, the agent tries to transmit over channel $f_i(k_t)$. If otherwise $f_i(k_t) = 0$, the agent chooses a random channel $m_i(t) \in \mathcal{C}$ which it will monitor for activity.

3. Subsequently, the agent observes the outcome of its choice: if the agent transmitted over some channel, she observes whether the transmission was successful. If it was, the agent will keep her strategy unchanged. If a collision occurred, the agent sets $f_i(k_t) := 0$ with probability p.

4. If the agent did not transmit, it observes whether there was a transmission on the channel $m_i(t)$ it monitored. If that channel was free, the agent sets $f_i(k_t) := m_i(t)$.

3. CONVERGENCE

An important property of the learning algorithm is if, and how fast it can converge to a pure-strategy Nash equilibrium of the channel allocation game for every signal value. The algorithm is randomized. Therefore, instead of analyzing its worst-case behavior (which may be arbitrarily bad), we will analyze its expected number of steps before convergence.

3.1 Convergence for $C = 1, K = 1$

We prove the following theorem:

THEOREM 1. *For N agents and $C = 1, K = 1, 0 < p < 1$, the expected number of steps before the allocation algorithm converges to a pure-strategy Nash equilibrium of the channel allocation game is $O\left(\frac{1}{p(1-p)} \log N\right)$.*

To prove the convergence of the algorithm, it is useful to describe its execution as a Markov chain.

When N agents compete for a single signal value (a "slot"), a state of the Markov chain is a vector from $\{0,1\}^N$ which denotes which agents are attempting to transmit. For the purpose of the convergence proof, it is only important how *many* agents are trying to transmit, not which agents. This is because the probability with which the agents back-off is the same for everyone. Therefore, we can describe the algorithm execution using the following chain:

Definition 1. A Markov chain describing the execution of the allocation algorithm for $C = 1, K = 1, 0 < p < 1$ is a chain whose state at time t is $X_t \in \{0, 1, \ldots, N\}$, where $X_t = j$ means that j agents are trying to transmit at time t.

The transition probabilities of this chain look as follows:

$$P(X_{t+1} = N | X_t = 0) = 1 \qquad \text{(restart)}$$

$$P(X_{t+1} = 1 | X_t = 1) = 1 \qquad \text{(absorbing)}$$

$$P(X_{t+1} = j | X_t = i) = \binom{i}{j} p^{i-j}(1-p)^j \qquad i > 1, j \leq i$$

All the other transition probabilities are 0.

We are interested in the number of steps it will take this Markov chain to first arrive at state $X_t = 1$ given that it started in state $X_0 = N$. This would mean that the agents converged to a setting where only one of them is transmitting, and the others are not. This quantity is known as the *hitting time*.

Definition 2. [10] Let $(X_t)_{t \geq 0}$ be a Markov chain with state space I. The *hitting time* of a subset $A \subset I$ is a random variable $H^A : \Omega \to \{0, 1, \ldots\} \cup \{\infty\}$ given by

$$H^A(\omega) = \inf\{t \geq 0 : X_t(\omega) \in A\}$$

Specifically, we are interested in the *expected* hitting time of a set of states A, given that the Markov chain starts in an initial state $X_0 = i$. We will denote this quantity

$$k_i^A = \mathbb{E}_i(H^A).$$

In general, the expected hitting time of a set of states A can be found by solving a system of linear equations. Solving them analytically for our Markov chain is however difficult. Fortunately, when the Markov chain has only one absorbing state $i = 0$, and it can only move from state i to j if $i \geq j$, we can use the following theorem to derive an upper bound on the hitting time (proved in [12]):

THEOREM 2. *Let $A = \{0\}$. If*

$$\forall i \geq 1 : E(X_{t+1} | X_t = i) < \frac{i}{\beta}$$

for some $\beta > 1$, then

$$k_i^A < \lceil \log_\beta i \rceil + \frac{\beta}{\beta - 1}$$

The Markov chain of our algorithm does not have the property required by this theorem. The problem is that the absorbing state is state 1, and from state 0 the chain goes back to N.

Nevertheless, we can use Theorem 2 to prove the following lemma:

LEMMA 1. *Let $A = \{0, 1\}$. The expected hitting time of the set of states A in the Markov chain described in Definition 1 is $O\left(\frac{1}{p} \log N\right)$.*

PROOF. We will first prove that the expected hitting time of a set $A' = \{0\}$ in a slightly modified Markov chain is $O\left(\frac{1}{p} \log N\right)$.

Let us define a new Markov chain $(Y_t)_{t \geq 0}$ with the following transition probabilities:

$$P(Y_{t+1} = 0 | Y_t = 0) = 1 \qquad \text{(absorbing)}$$

$$P(Y_{t+1} = j | Y_t = i) = \binom{i}{j} p^{i-j}(1-p)^j \qquad j \geq 0, i \geq 1$$

Note that the transition probabilities are the same as in the chain $(X_t)_{t \geq 0}$, except for states 0 and 1. From state 1 there is a positive probability of going into state 0, and state 0 is now absorbing. Clearly, the expected hitting time of the set $A' = \{0\}$ in the new chain is an upper bound on the expected hitting time of set $A = \{0, 1\}$ in the old chain. This is because any path that leads into state 0 in the new chain either does not go through state 1 (so it happened with the same probability in the old chain), or goes through state 1, so in the old chain it would stop in state 1 (but it would be one step shorter).

If the chain is in state $Y_t = i$, the next state Y_{t+1} is drawn from a binomial distribution with parameters $(i, 1-p)$. The expected next state is therefore

$$E(Y_{t+1} | Y_t = i) = i(1-p)$$

We can therefore use the Theorem 2 with $\beta := \frac{1}{1-p}$ to derive that for $A' = \{0\}$, the hitting time is:

$$k_i^{A'} < \left\lceil \log_{\frac{1}{1-p}} i \right\rceil + \frac{1}{p} \approx O(\frac{1}{p} \log i)$$

which is also an upper bound on k_i^A for $A = \{0, 1\}$ in the old chain. □

LEMMA 2. *The probability h_i that the Markov chain defined in Definition 1 enters state 1 before entering state 0, when started in any state $i > 1$, is greater than $1 - p$.*

PROOF. Calculating the probability that the chain X enters state 1 before state 0 is equal to calculating the *hitting probability*, i.e. the probability that the chain ever enters a given state, for a modified Markov chain where the probability of staying in state 0 is $P(X_{t+1} = 0 | X_t = 0) = 1$. For a set of states A, let us denote h_i^A the probability that the Markov chain starting in state i ever enters some state in A. To calculate this probability, we can use the following theorem (proved in [10]):

THEOREM 3. *Let A be a set of states. The vector of hitting probabilities $h^A = (h_i^A : i \in \{0, 1, \ldots, N\})$ is the minimal non-negative solution to the system of linear equations*

$$h_i^A = \begin{cases} 1 & \text{for } i \in A \\ \sum_{j \in \{0,1,\ldots,N\}} p_{ij} h_j^A & \text{for } i \notin A \end{cases}$$

For the modified Markov chain which cannot leave neither state 0 nor state 1, computing h_i^A for $A = 1$ is easy, since the matrix of the system of linear equations is lower triangular.

We'll show that $h_i \geq \gamma = 1 - p$ for $i > 1$ using induction. The first step is calculating h_i for $i \in \{0, 1, 2\}$.

$$\begin{aligned} h_0 &= 0 \\ h_1 &= 1 \\ h_2 &= (1-p)^2 h_2 + 2p(1-p)h_1 + p^2 h_0 \\ &= \frac{2p(1-p)}{1-(1-p)^2} = \frac{2(1-p)}{2-p} \geq 1-p. \end{aligned}$$

Now, in the induction step, derive a bound on h_i by assuming $h_j \geq \gamma = 1 - p$ for all $j < i, j \geq 2$.

$$\begin{aligned} h_i &= \sum_{j=0}^{i} \binom{i}{j} p^{i-j}(1-p)^j h_j \\ &\geq \sum_{j=0}^{i} \binom{i}{j} p^{i-j}(1-p)^j \gamma - ip^{i-1}(1-p)(\gamma - h_1) - p^i h_0 \\ &= \gamma - ip^{i-1}(1-p)(\gamma - 1) \geq \gamma = 1 - p. \end{aligned}$$

This means that no matter which state $i \geq 2$ the Markov chain starts in, it will enter into state 1 earlier than into state 0 with probability at least $1 - p$. □

From Lemma 2, we derive that in the original Markov chain (where stepping into state 0 meant going into state N), the chain takes on average $\frac{1}{1-p}$ passes through all its states before it converges into state 1. We know from Lemma 1 that one pass takes in expectation $O\left(\frac{1}{p}\log N\right)$ steps, so the expected number of steps before reaching state 1 is $O\left(\frac{1}{p(1-p)}\log N\right)$. This concludes the proof of Theorem 1.

3.2 Convergence for $C \geq 1, K = 1$

THEOREM 4. *For N agents and $C \geq 1, K = 1$, the expected number of steps before the learning algorithm converges to a pure-strategy Nash equilibrium of the channel allocation game is $O\left(C\frac{1}{1-p}\left[\frac{1}{p}\log N + C\right]\right)$.*

PROOF. In the beginning, in at least one channel, there can be at most N agents who want to transmit. It will take on average $O\left(\frac{1}{p}\log N\right)$ steps to get to a state when either 1 or 0 agents transmit (Lemma 1). We will call this period a *round*.

If all the agents backed off, it will take them on average at most C steps before some of them find an empty channel. We call this period a *break*.

The channels might oscillate between the "round" and "break" periods in parallel, but in the worst case, the whole system will oscillate between these two periods.

For a single channel, it takes on average $O\left(\frac{1}{1-p}\right)$ oscillations between these two periods before there is only one agent who transmits in that channel. For $C \geq 1$, it takes on average $O\left(C\frac{1}{1-p}\right)$ steps between "round" and "break" before all channels have only one agent transmitting. Therefore, it will take on average $O\left(C\frac{1}{1-p}\left[\frac{1}{p}\log N + C\right]\right)$ steps before the system converges. □

3.3 Convergence for $C \geq 1, K \geq 1$

To show what is the convergence time when $K > 1$, we will use a more general problem. Imagine that there are K identical instances of the same Markov chain. We know that the original Markov chain converges from any initial state to an absorbing state in expected time T. Now imagine a more complex Markov chain: In every step, it selects uniformly at random one of the K instances of the original Markov chain, and executes one step of that instance. What is the time T_{all} before all K instances converge to their absorbing states?

This is an extension of the well-known *Coupon collector's problem* ([4]). We will prove the following rough upper bound:

LEMMA 3. *Let there be K instances of the same Markov chain which is known to converge to an absorbing state in expectation in T steps. If we select randomly one Markov chain instance at a time and allow it to perform one step of the chain, it will take on average $E[T_{all}] = O(K^2 T)$ steps before all K instances converge to their absorbing states.*

PROOF. Let R_i be the number of steps of the joint Markov chain after which the instance i converges (by joint Markov chain we mean the chain that selects randomly an instance to perform one step). We are interested in

$$E[T_{all}] = E\left[\max_{i\in\{1,\ldots,K\}} R_i\right]$$

For this, it holds that

$$E\left[\max_{i\in\{1,\ldots,K\}} R_i\right] \leq E\left[\sum_{i=1}^{K} R_i\right] = \sum_{i=1}^{K} E[R_i]$$

For $\forall i$, $E[R_i] = KT$, because an instance i is selected in every step with probability $\frac{1}{K}$, and it takes it in expectation T steps to converge. Therefore, $E[T_{all}] \leq K^2 T$. □

For arbitrary $C \geq 1, K \geq 1$, the following theorem follows from Theorem 4 and Lemma 3:

THEOREM 5. *For N agents and $C \geq 1, K \geq 1, 0 < p < 1$, the expected number of steps before the learning algorithm converges to a pure-strategy Nash equilibrium of the channel allocation game for every $k \in \mathcal{K}$ is*

$$O\left(K^2 C\frac{1}{1-p}\left[C + \frac{1}{p}\log N\right]\right).$$

From [1] we know that any Nash equilibrium is a correlated equilibrium, and any convex combination of correlated equilibria is a correlated equilibrium. We also know that all the pure-strategy Nash equilibria that the algorithm converges to are efficient: there are no collisions, and in every channel for every signal value, some agent transmits. Therefore, we conclude the following:

THEOREM 6. *The learning algorithm defined in Section 2 converges in expected polynomial time (with respect to K, C, $\frac{1}{p}$, $\frac{1}{1-p}$ and $\log N$) to an efficient correlated equilibrium of the wireless channel allocation game.*

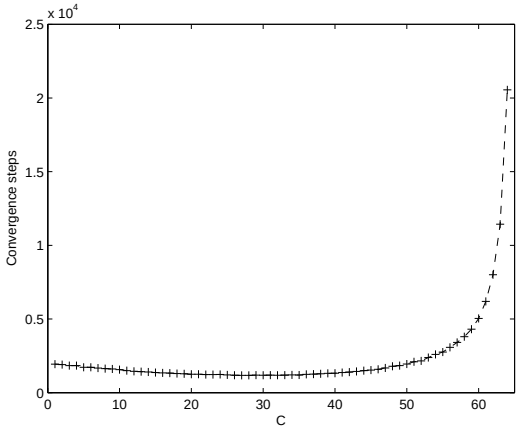


Figure 1: Average number of steps to convergence for $N = 64$, $K = N$ and $C \in \{1, 2, \ldots, N\}$.

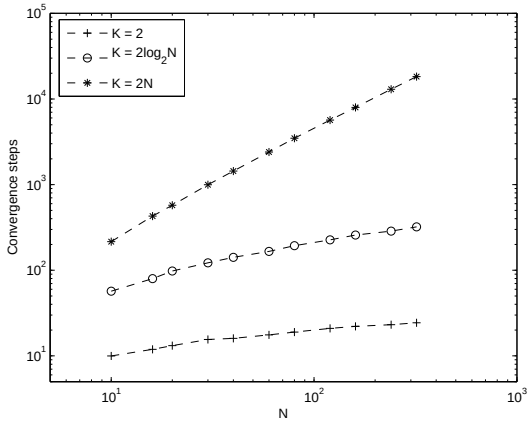


Figure 2: Average number of steps to convergence for $C = \frac{N}{2}$ and varying K.

4. FAIRNESS

Agents decide independently for each value of the coordination signal (a "slot"). Therefore, every agent has an equal chance that the game converges to an equilibrium which is favorable to her. If the agent can transmit in the resulting equilibrium for a given signal value, we say that the agent *wins* the slot. For C available channels and N agents, an agent wins a given slot with probability $\frac{C}{N}$ (since no agent can transmit in two channels at the same time).

We can describe the number of slots won by an agent i as a random variable X_i. This variable is distributed according to a binomial distribution with parameters $\left(K, \frac{C}{N}\right)$.

As a measure of fairness, we use the *Jain index* ([7]). For a random variable X, the Jain index is the following:

$$J(X) = \frac{(E[X])^2}{E[X^2]}$$

When X is distributed according to a binomial distribution with parameters $(K, \frac{C}{N})$, its first and second moments are

$$E[X] = K \cdot \frac{C}{N}$$
$$E\left[X^2\right] = \left(K \cdot \frac{C}{N}\right)^2 + K \cdot \frac{C}{N} \cdot \frac{N-C}{N},$$

so the Jain index is

$$J(X) = \frac{C \cdot K}{C \cdot K + (N - C)}.$$

For the Jain index it holds that $0 < J(X) \leq 1$. An allocation is considered fair if $J(X) = 1$.

THEOREM 7. *For any C, if $K = \omega\left(\frac{N}{C}\right)$, that is the limit $\lim_{N\to\infty} \frac{N}{C \cdot K} = 0$, then*

$$\lim_{N\to\infty} J(X) = 1,$$

so the allocation becomes fair as N goes to ∞.

PROOF. The theorem follows from the fact that

$$\lim_{N\to\infty} J(X) = \lim_{N\to\infty} \frac{C \cdot K}{C \cdot K + (N - C)}$$

For this limit to be equal to 1, we need

$$\lim_{N\to\infty} \frac{N - C}{C \cdot K} = 0$$

which holds exactly when $K = \omega\left(\frac{N}{C}\right)$ (note that we assume that $C \leq N$). □

5. EXPERIMENTAL RESULTS

5.1 Convergence

First, we are interested in the convergence of our allocation algorithm. From Section 3 we know that it is polynomial. How many steps does the algorithm need to converge in practice?

Figure 1 presents the average number of convergence steps for $N = 64$, $S = N$ and increasing number of available channels $C \in \{1, 2, \ldots, N\}$. Interestingly, the convergence takes the longest time when $C = N$. The lowest convergence time is for $C = \frac{N}{2}$, and for $C = 1$ it increases again.

What happens when we change the size of the signal space K? Figure 2 shows the number of convergence steps in that case, for increasing number of agents in the system. Note that this graph uses a double logarithmic scale, so a straight line denotes polynomial, rather than linear dependence of the number of convergence steps on N.

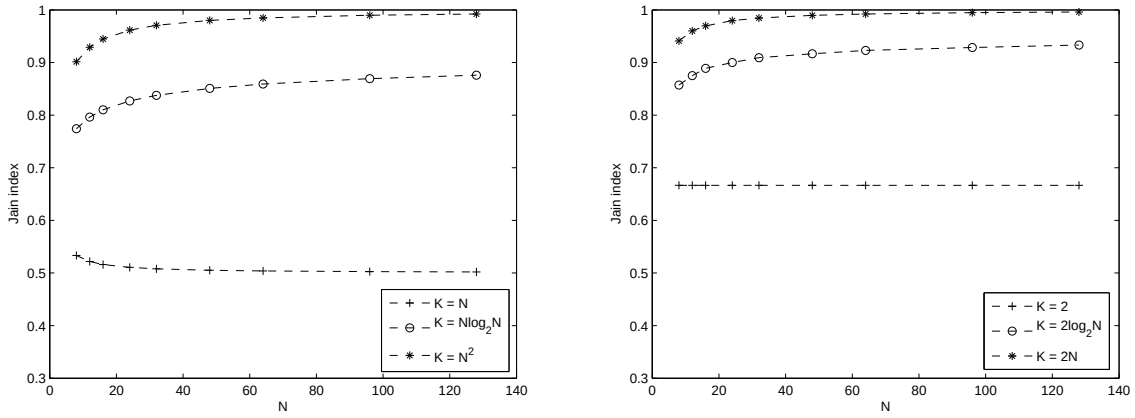
5.2 Fairness

From Section 4, we know that when $K = \omega\left(\frac{N}{C}\right)$, the Jain fairness index converges to 1 as N goes to infinity. But how fast is this convergence? How big do we need to choose K, depending on N and C, to achieve a reasonable bound on fairness?

Figure 3 shows the Jain index as N increases, for $C = 1$ and $C = \frac{N}{2}$ respectively, for various settings of K. Even though every time when $K = \omega\left(\frac{N}{C}\right)$ the Jain index increases, there is a marked difference between the various settings of K.

5.3 Optimizing Fairness

We saw how fair the outcome of the allocation algorithm is when agents consider the game for each slot independently.



(a) $C = 1$ (b) $C = \frac{N}{2}$

Figure 3: Jain fairness index for different settings of C and K, for increasing N.

However, is it the best we can do? Can we further improve the fairness, when each agent correlates her decisions for different signal values?

In a perfectly fair solution, every agent wins (and consequently can transmit) for the same number of slots. However, we assume that agents do not know how many other agents there are in the system. Therefore, the agents do not know what is their fair share of slots to transmit in. Nevertheless, they can still use the information in how many slots they already transmitted to decide whether they should back-off and stop transmitting when a collision occurs.

Definition 3. For a strategy f_i of an agent i, we define its *cardinality* as the number of signals for which this strategy tells the agent to transmit:

$$|f_i| = |\{k \in \mathcal{K} | f_i(k) > 0\}|$$

Intuitively, agents whose strategies have higher cardinality should back-off more often than those with a strategy with low cardinality.

We compare the following variations of the channel allocation scheme, which differ from the original one only in the probability with which agents back off on collisions:

Constant Our scheme; Every agent backs off with the same constant probability p.

Linear The back-off probability is $p = \frac{|f_i|}{K}$.

Exponential The back-off probability is $p = \gamma^{\left(1-\frac{|f_i|}{K}\right)}$ for some parameter $0 < \gamma < 1$.

Worst-agent-last In case of a collision, the agent who has the *lowest* $|f_i|$ does not back off. The others who collided, do back off. This is a greedy algorithm which requires more information than what we assume that the agents have.

To compare the fairness of the allocations in experiments, we need to define the Jain index of an actual allocation. For an allocation $\mathbb{X} = (X_1, X_2, \ldots, X_N)$, its Jain index is:

$$J(\mathbb{X}) = \frac{\left(\sum_{i=1}^{N} X_i\right)^2}{N \cdot \sum_{i=1}^{N} X_i^2}$$

Figure 4 shows the average Jain fairness index of an allocation for the back-off probability variations. The fairness is approaching 1 for the *worst-agent-last* algorithm. It is the worst if everyone is using the same back-off probability. As the ratio between the back-off probability of the lowest-cardinality agent and the highest-cardinality agent decreases, the fairness increases.

This shows that we can improve fairness by using different back-off probabilities. Nevertheless, the shape of the fairness curve is the same for all of them. Furthermore, the exponential back off probabilities lead to much longer convergence, as shown on Figure 5.

6. RELATED WORK

Broadly speaking, in this paper we are interested in games where the payoff an agent receives from a certain action is inversely proportional to the number of other agents who chose the same action. How can we achieve efficient and fair outcome in such games? Variants of this problem have been studied in several previous works.

The simplest such variant is the *Minority game* ([3]). In this game, N agents have to simultaneously choose between two actions. Agents who chose an action which was chosen by a minority of agents receive a payoff of 1, whereas agents whose action choice was in majority receive a payoff of 0. This game has many pure-strategy Nash equilibria, in which some group of $\left\lfloor \frac{N-1}{2} \right\rfloor$ agents chooses one action and the rest choose the other action. Such equilibria are efficient, since the largest possible number of agents achieve the maximum payoff. However, they are not fair: the payoff to the losing group of agents is always 0. This game has also one mixed-strategy NE which is fair: every agent chooses its action randomly. This equilibrium, on the other hand, is not efficient: the expected size of the minority group is lower than $\left\lfloor \frac{N-1}{2} \right\rfloor$ due to variance of the action selection.

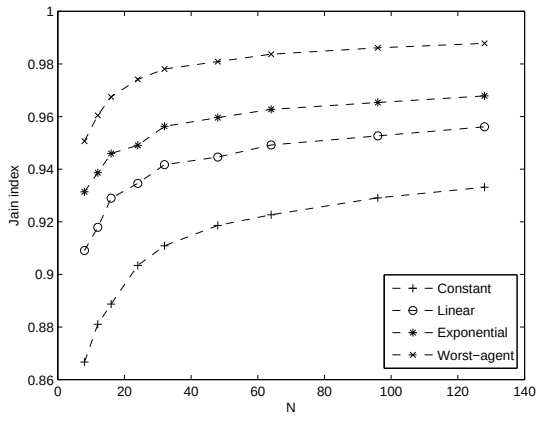


Figure 4: Jain fairness index of the channel allocation scheme for various back-off probabilities, $C = \frac{N}{2}$, $K = 2\log_2 N$

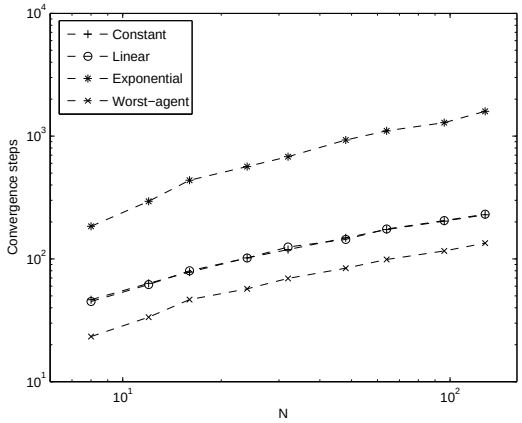


Figure 5: Convergence steps for various back-off probabilities.

Savit *et al.* ([13]) show that if the agents receive feedback on which action was in the minority, they can learn to coordinate better to achieve a more efficient outcome in a repeated minority game. They do this by basing the agents' decisions on the history of past iterations. Cavagna [2] shows that the same result can be achieved when agents base their decisions on the value of some random coordination signal instead of using the history. This is a direct inspiration for our work.

The ideas from the literature on Minority games have recently found their way into the cognitive radio literature. Mahonen and Petrova [8] present a channel allocation problem much like ours. The agents learn which channel they should use using a strategy similar to the strategies for minority games. The difference is that instead of preferring the action chosen by the minority, in the channel allocation problem, an agent prefers channels which were not chosen by anyone else. Using this approach, Mahonen and Petrova are able to achieve a stable throughput of about 50% even when the number of agents who try to transmit over a channel increases. However, each agent is essentially choosing one out of a fixed set of strategies, which they cannot adapt. Therefore, it is very difficult to achieve a perfectly efficient channel allocation.

Another, more general variant of our problem, called *dispersion game* was described by Grenager *et al.* in [6]. In a dispersion game, agents can choose from several actions, and they prefer the one which was chosen by the smallest number of agents. The authors define a *maximal dispersion outcome* as an outcome where no agent can move to an action with fewer agents. The set of maximal dispersion outcomes corresponds to the set of pure-strategy Nash equilibria of the game. They propose various strategies to converge to a maximal dispersion outcome, with different assumptions on the information available to the agents. On the contrary with our work, the individual agents in the dispersion games do not have any particular preference for the actions chosen or the equilibria which are achieved. Therefore, there are no issues with achieving a fair outcome.

Verbeeck *et al.* [14] use reinforcement learning, namely *linear reward-inaction automata*, to learn Nash equilibria in common and conflicting interest games. For the class of conflicting interest games (to which our wireless channel allocation game belongs), they propose an algorithm that allows the agents to circulate between various pure-strategy Nash equilibria, so that the outcome of the game is fair. In contrast with our work, their solution requires more communication between agents, and it requires the agents to *know* when the strategies converged. In addition, linear reward-inaction automata are not guaranteed to converge to a PSNE in conflicting interest games; they may only converge to pure strategies.

All the games discussed above, including the wireless channel allocation game, form part of the family of *potential games* introduced by Monderer and Shapley ([9]). A game is called a potential game if it admits a *potential function*. A potential function is defined for every strategy profile, and quantifies the difference in payoffs when an agent unilaterally deviates from a given strategy profile. There are different kinds of potential functions: exact (where the difference in payoffs to the deviating agent corresponds directly to the difference in potential function), ordinal (where just the sign of the potential difference is the same as the sign of the payoff difference) etc.

Potential games have several nice properties. The most important is that any pure-strategy Nash equilibrium is just a local maximum of the potential function. For finite potential games, players can reach these equilibria by unilaterally playing the best-response, no matter what initial strategy profile they start from.

The existence of a natural learning algorithm to reach Nash equilibria makes potential games an interesting candidate for our future research. We would like to see to which kind of correlated equilibria can the agents converge there, if they can use a simple correlation signal to coordinate.

7. CONCLUSIONS

In this paper, we proposed a new approach to reach desirable correlated equilibria in games. Instead of using a "smart" coordination device, as the original definition of CE assumes, we use "stupid" signal, a random integer k taken from a set $\mathcal{K} = \{0, 1, \ldots, K-1\}$, which has no a priori relation to the game. Agents then are "smart": they learn,

for each value of the coordination signal, which action they should take.

We showed a learning strategy which, for a variant of a wireless channel allocation game, converges in expected polynomial number of steps to an efficient correlated equilibrium. We also proved that this equilibrium becomes increasingly fair as K, the number of available synchronization signals, increases. We have confirmed both the fast convergence as well as increasing fairness with increasing K experimentally.

In the future work, we would like to see whether this approach ("stupid" coordination signal and "smart" learning agents) can help to reach desirable correlated equilibria of other games, such as potential games.

8. REFERENCES

[1] R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, March 1974.

[2] A. Cavagna. Irrelevance of memory in the minority game. *Physical Review E*, 59(4):R3783–R3786, April 1999.

[3] D. Challet, M. Marsili, and Y.-C. Zhang. *Minority Games: Interacting Agents in Financial Markets (Oxford Finance)*. Oxford University Press, New York, NY, USA, January 2005.

[4] W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley, 3 edition, January 1968.

[5] D. P. Foster and R. V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2):40–55, October 1997.

[6] T. Grenager, R. Powers, and Y. Shoham. Dispersion games: general definitions and some specific learning results. In *Proceedings of the Eighteenth national conference on Artificial intelligence (AAAI-02)*, pages 398–403, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[7] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, Digital Equipment Corporation, September 1984.

[8] P. Mahonen and M. Petrova. Minority game for cognitive radios: Cooperating without cooperation. *Physical Communication*, 1(2):94–102, June 2008.

[9] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, pages 124–143, May 1996.

[10] J. R. Norris. *Markov Chains (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, July 1998.

[11] C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):1–29, July 2008.

[12] V. Rego. Naive asymptotics for hitting time bounds in markov chains. *Acta Informatica*, 29(6):579–594, June 1992.

[13] R. Savit, R. Manuca, and R. Riolo. Adaptive competition, market efficiency, and phase transitions. *Physical Review Letters*, 82(10):2203–2206, March 1999.

[14] K. Verbeeck, A. Nowé, J. Parent, and K. Tuyls. Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Autonomous Agents and Multi-Agent Systems*, 14(3):239–269, June 2007.

Sequential targeted optimality as a new criterion for teaching and following in repeated games

Max Knobbout
Utrecht University
Dept. of Computer Science
The Netherlands
mknobbout@gmail.com

Gerard A.W. Vreeswijk
Utrecht University
Dept. of Computer Science
The Netherlands
gv@cs.uu.nl

ABSTRACT

In infinitely repeated games, the act of teaching an outcome to our adversaries can be beneficial to reach coordination, as well as allowing us to 'steer' adversaries to outcomes that are more beneficial to us. Teaching works well against followers, agents that are willing to go along with the proposal, but can lead to miscoordination otherwise. In the context of infinitely repeated games there is, as of yet, no clear formalism that tries to capture and combine these behaviours into a unified view in order to reach a solution of a game. In this paper, we propose such a formalism in the form of an algorithmic criterion, which uses the concept of targeted learning. As we will argue, this criterion can be a beneficial criterion to adopt in order to reach coordination. Afterwards we propose an algorithm that adheres to our criterion that is able to teach pure strategy Nash Equilibria to a broad class of opponents in a broad class of games and is able to follow otherwise, as well as able to perform well in self-play.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent system*

General Terms

Algorithms, Theory

Keywords

Game Theory, Implicit Cooperation, Coordination, Teaching

1. INTRODUCTION

In the area of multiagent learning, game theory is an important tool to model the interaction between agents that arises. In order to establish and sustain coordination in a repeated game, the agents need to achieve a mutual beneficial outcome. In a setting where the agents are not pre-coordinated and have no explicit way of communication (only by observing actions/outcomes) this quickly becomes a complex scenario. From this perspective, the act of proposing (or forcing) an outcome to our adversaries makes sense, which we will informally describe as 'teaching' behaviour. On the other hand we have 'following' behaviour, which can be understood as the act of going along with such a proposal. Teaching behaviour does not only make sense in order to reach coordination, but often adopting the role of a teacher allows us to 'steer' followers to outcomes that are more beneficial to us. However, it can lead to miscoordination if multiple agents try to teach different outcomes of the game. Without an external designation of these roles, it can be hard to decide whether to take on the role of a teacher or a follower. In the context of infinitely repeated games there is, as of yet, no clear formalism that tries to capture and combine these behaviours into a unified view in order to reach a solution of a game. A reason for this could be the fact that the distinction between teaching behaviour on one hand, and following behaviour on the other, is often not so clear-cut as one might presume. In order for the reader to place this observation into perspective, the next section discusses different lines of previous work related to this work in which either (1) intuitively (a combination of) teaching and following behaviour occurs, but the authors do not explicitly mention this or in which (2) the authors mention the existence of (one of) these behaviours but do not provide a formalism or a unified view.

Cite as: Sequential targeted optimality as a new criterion for teaching and following in repeated games, Max Knobbout and Gerard A.W. Vreeswijk, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011),* Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 517-524.

2. PREVIOUS WORK

In order to make the distinction between teaching and following behaviour, one can try and identify the nature of teaching behaviour. In [5], the authors mention the concept of teaching (or leading) in repeated games, which is introduced in the form of two strategies. These strategies, named Godfather and Bully, can be used to induce good performance from 'followers'. Bully assumes it has first mover advantage, and optimizes its payoff assuming that the other player is a follower. Godfather (a generalization of Tit for Tat) uses the threat of security level to maintain a mutually beneficial outcome. Intuitively, these strategies can indeed be understood as teacher strategies, but the authors do not explicitly mention why this is the case. In [3], the authors argue that the main difference between teaching strategies as opposed to following strategies is the fact that teacher strategies also take into account the payoff of the opponent. We believe that this notion is insufficient, since arbitrary mixed strategies can also be considered teaching strategies: they force the opponent into a way of play by reducing the setting to a Markov problem.

Another approach can be to try to identify follower strategies in order to make the distinction between the two. Fol-

lower strategies can be intuitively understood as strategies that condition on the way of play of the opponent. Typical examples are model-based learners, such as Fictitious Play and Rational Learning. However, Godfather also conditions on the way of play of the opponent, so it is a natural question to ask why it is not a following strategy. In [5], the authors argue that reinforcement learning algorithms like Q-learners can be considered followers. However, we believe that Q-learners capture a bit of both: with a high learning rate they are able to quickly adapt (follow), while a low learning rate ensures that the agent stays committed to a way of play (teach). Even more so, the difference between teaching and following quickly becomes a grey area when we consider strategies that use a multitude of strategies, like no-regret learners.

We also have related research in which intuitively the combination of teaching and following behaviour occurs, but often the authors never seem to mention it. One example in which the authors do mention this can be found in [3], in which the authors use a variant of godfather that uses both teacher and follower utility together with the notion of guilt to determine the length of the punishment phase. Here guilt is the extra reward the opponent has accumulated by deviating from the target solution. The problem with this approach is that guilt has little scientific basis and it is often very unclear if the opponent should remain guilty in particular cases. Moreover, the algorithm can lead to very unpredictable and complex behaviour, which is hard for the opponent to predict this.

In other research we have that authors never mention the existence of teaching and following behaviour, even though their approach does intuitively seem to exhibit it to some degree. In [1], the authors use the WoLF principle (Win or Learn Fast) to extend the basic gradient ascent IGA algorithm. The WoLF principle states that if the player is winning, the algorithm should use a lower learning rate in the case if it is losing. Adopting a low learning rate can be seen as unwillingness to change your strategy, hence teaching behaviour, while adopting a high learning rate can be seen as follower behaviour. Another example of such research can be found in [2], where the authors propose the AWESOME (adapt when everybody is stationary, otherwise move to equilibrium) algorithm which is able to play a best response against stationary opponents in n-action n-player games, and is also able to converge to a Nash equilibrium in self-play. This algorithm does exactly what the name implies, except the other way around: It starts out with the assumption that the opponents are equilibrium players, and thus plays their part of the pre-computed equilibrium strategy (teaching). If this hypothesis later on is refuted, it then proceeds to assume the opponent is stationary and adapts accordingly by playing a best-response to the empirical frequency of play (following). But again, the point about teaching and following behaviour is not explicitly made, the algorithm merely works this way to ensure the above mentioned properties. In a multitude of papers found in [6], [7] and [8], the authors propose a criterion that states that an algorithm should achieve a close to optimal payoff against certain classes of opponents with high probability. It is then possible to use this criterion to demand a best response value against a multitude of opponents, which can lead to interesting step-wise teaching and following behaviour. For example in [6], the algorithm (1) first considers that the opponent is stationary (and plays a best response), (2) afterwards considers that the opponent is a follower (and plays a mixed strategy variant of Bully) and (3) if no considerations can be made, concludes that the best we can do is follow (and plays Fictitious Play). But again, the point about teaching and following is not explicitly made. As we will motivate later, we believe that this approach, using beliefs about our opponent to decide whether to teach or to follow, is a suitable approach for our problem.

Figure 1: Non-teaching game

	Left	Right
Top	**1**,0	**0**,0
Bottom	**0**,0	**1**,0

Lastly, there is economic research about the subject which, for the purpose of this section, should not be omitted. In this approach (market) leaders and followers make up the complex dynamics of a system that arises. The point of departure is the model, such as the Stackelberg leadership model, where leader and follower are defined by the game itself and distinguished by the first-mover advantage. An important question here is whether or not the notion of teaching and following can completely exist outside the model(game)-level. This is actually questionable, as we will demonstrate by the following example. Consider the game shown in Figure 1, which we have chosen to name the "non-teaching game". Now consider that we are the row player, and our opponent is the column player, and we are playing an infinitely repeated game where we want to maximize our average returned payoff. In this game, the opponent is indifferent about all possible outcomes of the game. Forcing outcomes by leading (Bully) or retaliating (Godfather) is impossible in this game, since the opponent does not prefer any outcome over another. The security value for this game is 0.5 by adopting the mixed strategy (0.5,0.5) (the security value is the value the agent can guarantee regardless of the opponent by playing purely defensively). Arguably, the best strategy to adopt in this game is to start out with this defensive strategy and afterwards play a best response based on the frequency of play of the opponent (for example in the case he plays Left more than Right), or in other words following behaviour. These types of games are evidence that teaching might not be feasible in all games.

3. BASIC CONCEPTS

In the previous section, we saw that the distinction between teaching and following behaviour is not so clear cut as one might suspect. In this section we will provide the reader with a criterion that tries to capture the essence of teaching and following. In our setting we consider infinitely repeated, complete, perfect information normal-form games (complete and perfect information implies that all agents have knowledge about the payoffs of the game and the actions that have taken place), with 2 players and n actions. These games are defined in the normal game-theoretic sense, that is to say they consist of a finite set of players, a finite set of actions for each player and a real valued payoff function that maps for each player an action profile to a real

number. The goal is to maximize the average reward the player receives. The n player extension might be interesting for possible future research, but for now we do not want to complicate matters too much.

During our approach we will sometimes stop and deliberate on two important aspects of teaching and following, which is the *what* ("what to teach?") and the *when* ("when to teach and when to follow?"). Here we can already partly state the "what", namely we should teach something that is within the capabilities of our opponent. This idea to get a best response value against strategies that belong to a certain class of strategies is discussed by Shoham and Brown in [9, pp. 222–223], where they discuss the concept of targeted learning and use a criterion named (efficient) *targeted optimality*. The following definition is similar, except that we replaced the somewhat vague notion of 'class of opponents' to a set of strategies, which can be any subset of the full strategy set available.

Definition 1. Given a (finite or infinite) strategy set S, a strategy is said to be *targeted optimal* if it holds that for any choice of $\epsilon > 0$ and $\delta > 0$ there should exist a number of rounds τ, polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, such that for every number of rounds $t \geq \tau$ the strategy against an arbitrary strategy $\sigma \in S$ achieves average payoff of at least $V_{BR}(\sigma) - \epsilon$ with probability $1 - \delta$, where $V_{BR}(\sigma)$ is the value of the best response given that the opponent plays σ. If, during run-time, for a choice of ϵ and δ the average payoff when playing our strategy remains ϵ-close to the best response value for every number of rounds $t \geq \tau$, where τ is defined as previous, we say that the property of optimality is *maintained*.

Notice that (ϵ, δ)-optimality is quite a weak notion of optimality. To explain this choice in the context of teaching and following, the choice of ϵ can be explained by the fact that sometimes we need room to identify whether or not the opponent can be taught. The latter choice can be explained by the fact that we can never be certain whether or not the opponent actually belongs to the target class. Here, δ can be seen as a 'measure of stubbornness' to determine when to abandon our hopes to achieve an average payoff ϵ-close to the best response value (namely when we are certain enough that the opponent does not belong to the target class). We believe that this measure is something which we need when it comes to teaching. In the next part of this section, definition 1 will be used to define a novel criterion that tries to capture teaching and following behaviour into a unified view.

3.1 A new criterion for teaching and following

The first step in the construction of our criterion is to use the notion of targeted optimality and create a new notion in which it is applied sequentially. We propose this new criterion as *sequentially targeted optimality* (we drop the 'efficient' adjective to keep the criterion name more compact).

Definition 2. A strategy σ is said to be *sequentially targeted optimal* given strategy sets S^p and S^s if it holds that this strategy first deploys a strategy, referred to as σ^p, and σ^p should be targeted optimal given strategy set S^p. If for a choice of ϵ and δ during run-time the property of optimality is not maintained (either because (1) the strategy of the opponent indeed belongs to S^p but with probability δ we have not achieved an average payoff ϵ-close to the best response value or (2) the strategy of the opponent does not belong to S^p), then our strategy should deploy another strategy, referred to as σ^s, and σ^s should be targeted optimal given S^s. If a strategy is sequentially targeted optimal with respect to S^p and S^s, we refer to the first deployed strategy σ^p as the primary strategy, and the second deployed strategy σ^s as the secondary strategy.

The reason that we also applied the weaker notion of (ϵ, δ)-optimality to the secondary strategy is simply because we want to have room for an algorithm to also adhere to other criteria (and not just one criterion which overrules any other possible criterion). Notice that this criterion already states some of the aspects of teaching and following. It states the "what": we try to achieve the best possible payoff (or at least arbitrary close to) given that we condition on the opponent. It also (partly) states the "when": first we could have a period in which we try to 'teach' the opponent, and if that fails, we could have a period in which we try to 'follow'. However, if we just use an arbitrary primary strategy set and secondary strategy set to create a sequential targeted optimal strategy, this resulting strategy can definitely not be labelled a teaching- and following strategy in all cases. This is because we have not laid any restrictions on these strategy sets and because we have to show that teaching can indeed be beneficial. However, formalizing a notion of teaching and following strategies is problematic, since it is often a grey area as we saw in section 2. To overcome this problem, we will try to define when a sequential targeted optimal strategy is a sequentially teaching-following strategy as a whole, without defining its specific parts S^p and S^s. To do this, we will first introduce the notion of self-teachability.

Definition 3. A strategy σ is *self-teachable* if it is sequentially targeted optimal given S^p and S^s, using primary strategy σ^p and secondary strategy σ^s, if it holds that $\sigma^p \in S^s$ and $\sigma^s \in S^p$.

Loosely speaking, a strategy is self-teachable if we are able to 'follow' (and get our desired best response value) on the strategy which we use to 'teach' and we are able to 'teach' (and get our desired best response value) on the strategy which we use to 'follow'. Thus, if a strategy is self-teachable it contains some sort of symmetry within the different strategies that are deployed. Using this notion of 'symmetry', we propose a novel criterion that tries to capture both teaching and following behaviour, which is given by the *sequential teaching-following* criterion in the next definition.

Definition 4. A strategy is said to be a *sequential teaching-following strategy* if it is self-teachable in a set of games G (that is, it achieves the property of self-teachability in all these games) using strategy sets S^p and S^s and if it holds that in all games belonging to G, the guaranteed best response value of playing against a strategy from S^p is at least as high as the guaranteed best response value of playing against a strategy from S^s:

$$\min_{\sigma \in S^p} V_{BR}(\sigma) \geq \min_{\sigma' \in S^s} V_{BR}(\sigma')$$

If a strategy is a sequential teaching-following strategy, we refer to the primary strategy as the teacher strategy and the secondary strategy as the follower strategy.

This criterion states when a sequential targeted optimal strategy is a sequential teaching-following strategy without

Figure 2: Matching pennies

	Left	Right
Top	**-1**,1	**1**,-1
Bottom	**1**,-1	**-1**,1

explicitly specifying its parts S^p and S^s and it states a certain beneficialness which is restricted to a set of games. The beneficialness is stated in terms of payoff guarantees (and not for example in terms of maximum payoff or expected payoff), because minimum payoff is an important concept in repeated games to identify enforceable outcomes. The restriction to a set of games is because we already talked about the feasibility of teaching: not all games are suited for teaching. It also allows us to play around more with the concept, since we can form sequential teaching-following strategies that use non-mixed strategies like Bully and Godfather as teaching strategies for example, without necessarily consorting to mixed variants. This is because in some games, the only equilibrium strategies are mixed. One such well known example is the matching pennies game shown in Figure 2. Moreover, there is nothing restricting anyone to drop the requirement by creating a sequential teaching-following strategy that conditions over every game. We believe that this notion captures the essence of teaching and following: here teaching and following are defined as behaviours that are able to coordinate together (both players are able to get a best response) and they can be separated by the fact that teacher behaviour has a certain beneficialness to it.

The symmetry we demanded in our previous definition of teaching-following strategies may seem overly restrictive, since we demanded a 2-way interaction: teaching should be good against following and vice versa. However, this demand not only serves as a way to distinguish teaching from following strategy, but also to ensure certain beneficial properties in self-play.

PROPOSITION 1. *When using a sequential teaching-following strategy in self-play, if it is the case that one player maintains its teacher strategy $\sigma^p \in S^s$ while the other maintains his follower strategy $\sigma^s \in S^p$, then both players converge to a Nash equilibrium.*

PROOF. Since the strategy σ^p is targeted optimal given strategy set S^p for any arbitrary choice of $\epsilon > 0$, and strategy σ^s is targeted optimal given strategy set S^s for any arbitrary choice of $\epsilon' > 0$, we know that the first player will achieve for any ϵ an average payoff ϵ-close to $V_{BR}(\sigma^s)$ while the second player will achieve for any ϵ' a pay ϵ'-close to $V_{BR}(\sigma^p)$. This means that, given an arbitrary ϵ and ϵ', it holds that for the first player there are no strategies available such that more than ϵ expected payoff can be gained and for the second player there are no strategies available such that more than ϵ' expected payoff can be gained. Thus both players can not gain more than $\max(\epsilon, \epsilon')$ by deviating unilaterally, which implies a $\max(\epsilon, \epsilon')$-Nash equilibrium. Since the players maintain their strategies, we can let $\epsilon \to 0$ and $\epsilon' \to 0$, and thus $\max(\epsilon, \epsilon') \to 0$, which means that in the limit the players converge to a Nash equilibrium. □

This proposition is important when we want to show when a specific teaching-following strategy converges to a Nash equilibrium in self-play. As we will see later, in order to guarantee convergence to a Nash equilibrium in self-play we also need to consider the case in which both the players maintain their teaching strategy (if possible) and the case in which both players maintain their following strategy.

The teaching-following criterion we supplied tried to incorporate intuitive aspects of teaching and following, such as the "what" and the "when". Based on the criterion, it can be argued that in infinitely repeated games, it can be beneficial to first try to teach an outcome that allows us to receive a greater guaranteed outcome. This is especially the case for conservative agents that care more about payoff guarantees than payoff maximization. Many known strategies can be extended to have a teaching phase, so there is not really anything to lose given that the game is not finite. If the rate of convergence plays a role, the criterion also states that the properties should be achieved in efficient time. Moreover, as we will see later on with our algorithm, combining two strategies with the use of the criterion will cause the resulting strategy to maintain many of the properties of the original strategies. In other words, the criterion not only tries to capture the essence of teaching and following, but it is also a beneficial criterion for algorithms to adhere to. Moreover, it allows authors to create strategies in terms of 'weaknesses': what works good against what in which situations? In the next section we will create an algorithm that adheres to our proposed criterion.

4. IMPLEMENTATION

In this section, we will first look at the teaching and following component of our algorithm individually and afterwards we will combine them to create an algorithm that is both able to teach and follow in repeated games by adhering to our teaching-following criterion.

4.1 Teaching strategy

For the teaching part of our strategy, we will use a variant of Bully. We already saw that intuitively this strategy is indeed a teaching strategy, since it assumes it has Stackelberg leader advantage. On the other hand, Bully does not work well in all games, in particular games that require mixed equilibria. In the long run this will imply that our strategy is not able to teach beneficial outcomes in all games.

The idea is that Bully, in some games, works specifically well against opponents that are willing 'to go along with the proposal', such as learning rules that play a best response to the distribution of play. As it turns out, the class of strategies that are 'susceptible' to Bully is very broad and covers many examples found in literature. We refer to these strategies as *pure consistent* strategies, which is a superclass of the consistent strategies defined in [4]. The difference is that pure consistent strategies should achieve a best response against pure strategies, in stead of arbitrary mixed strategies in the case of consistent strategies. We also extend the definition with the notion of a polynomial rate of convergence, which will play a role in the next proposition.

Definition 5. A strategy is said to be *ϵ-pure consistent* if there exists a T such that against any pure strategy σ_{-i} and for any $t > T$ the strategy achieves a payoff ϵ-close to $V_{BR}(\sigma_{-i})$ with probability $1 - \epsilon$. A strategy is *pure consistent* if it is ϵ-pure consistent for every positive ϵ and is said to have a *polynomial rate of convergence* if T is polynomial in $\frac{1}{\epsilon}$.

It can easily be shown that any consistent strategy (like Fictitious play), universal consistent strategy (like no-regret learners) and rational strategy (mentioned in [1], not to be confused with the economical definition of rationality) are pure consistent, as well as countless more strategies. The reason for this is because a pure strategy is very easy to learn for the opponent. This is again one of the beautiful aspects of teaching and following: if the message we are trying to teach is simple, the class we can target is much larger than in the case in which we are trying to teach a more complex message.

As our teacher strategy, we use a modified version of Bully. This is because Bully is not well defined in cases in which our opponent is indifferent about several outcomes. To cope with this, we define our teacher value and action in the following way:

Definition 6. The teacher value, $V_{teacher}$, is defined as:

$$V_{teacher} = \max_i V_i(i, j_i^*)$$

where

$$j_i^* = \operatorname*{argmin}_{j \in J_i} V_i(i, j)$$

and

$$J_i = \{\, a \mid V_{-i}(i, a) = \max_j V_{-i}(i, j) \,\}$$

In short, $V_{teacher}$ is defined as the best possible payoff the agent can guarantee by assuming it has first-mover advantage and by assuming that the opponent plays a best response to this pure strategy which is least beneficial to us. The action belonging to $V_{teacher}$ is defined as $a_{teacher}$.

Observe that $V_{teacher}$ is indeed a best response value against an arbitrary pure consistent opponent (notice that it can still be considered a best response value in repeated games if the opponent is (universally) consistent, as long as we are teaching a feasible and enforceable outcome; more on this observation later). However, if it is the case that $V_{teacher} < V_{Maximin}$ (recall for example the matching pennies game in Figure 2), it is arguably better to play our (possibly mixed) Maximin strategy. As we will see later, this will not pose a problem since the notion of teaching-following can be restricted to a set of games. The proof that we will use is unique in the sense that it does not rely on probability bounds to show a probability dependent payoff guarantee. This is because our opponent is using a learning/adaptive strategy (which cannot be simply captured by a Random variable). However, observe that if we play a pure strategy against a pure consistent strategy, the strategy we play also seems to show 'consistent behaviour'. This idea will be the basis of the upcoming proof, in which we will show targeted optimality against the set of pure consistent strategies by adopting the strategy in which we repeatedly play $a_{teacher}$.

PROPOSITION 2. *For any choice of $\epsilon > 0$ and $\delta > 0$ against an opponent that uses a pure consistent strategy σ_{-i} with a polynomial convergence rate, there exists a finite T, polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, such that playing $a_{teacher}$ repeatedly will for any $t > T$ result in an average payoff of at least $V_{teacher} - \epsilon$ with probability $1 - \delta$ against this opponent.*

PROOF. We will first show that for any given value of ϵ, there exists an $\epsilon' > 0$, such that if it is the case that our opponent with probability equal or greater than $1 - \epsilon'$ receives an average payoff ϵ'-close to his optimal payoff, we receive an average payoff ϵ-close to $V_{teacher}$. Since our opponent has a polynomial rate of convergence, we use a polynomial function $T_{-i}(\frac{1}{\epsilon'})$ to denote the actual time steps needed to achieve the property of pure consistency. Let p_i and p_{-i} be the payoff belonging to the action profile $(a_{teacher}, BR(a_{teacher}))$. Without loss of generality, we consider that there is another action profile in the vector, (a_i, a_{-i}) with payoff p'_i and p'_{-i} respectively such that p'_i is the worst payoff in the vector for our agent and p'_{-i} the (second) best for the other agent. Let's also consider that $p_i > p'_i + \epsilon$, since otherwise any combination of actions by the opponent would guarantee that the average payoff we receive is larger or equal than $V_{teacher} - \epsilon$. Similarly we have that $p_{-i} > p'_{-i}$, since by definition of $a_{teacher}$ we have that any action with payoff equal to p_{-i} will net our agent a payoff of at least $V_{teacher}$. For every possible ϵ, the worst-case candidate h to violate the property is playing k proportion $(a_{teacher}, BR(a_{teacher}))$ and $(1-k)$ proportion (a_i, a_{-i}) such that it holds that our opponent still receives an average payoff ϵ'-close to his optimal payoff. Since in this case $V_{teacher} = p_i$ and $V_{BR}(a_{teacher}) = p_{-i}$, we have to find an ϵ' such that the proportion k is high enough such that:

$$k * p_{-i} + (1 - k) * p'_{-i} + \epsilon' \geq p_{-i}$$

implies that the following also holds:

$$k * p_i + (1 - k) * p'_i + \epsilon \geq p_i$$

Solving for ϵ', we see that

$$\epsilon' \leq \epsilon * \kappa$$

where

$$\kappa = \left(\frac{p_{-i} - p'_{-i}}{p_i - p'_i} \right)$$

Since we know that $p_i > p'_i$, $p_{-i} > p'_{-i}$ and $\epsilon > 0$, this outcome is strictly positive. Thus for ϵ' any value in the interval $(0...b]$, where $b = \epsilon * \kappa$ guarantees that if our opponent (with probability $1 - \epsilon'$) receives a payoff ϵ'-close to his optimal payoff then our agent receives a payoff ϵ-close to $V_{teacher}$. Notice that this happens after $T_{-i}(\frac{1}{\epsilon'})$ iterations.

The second step in our proof is to observe that this result is general enough to apply to any game, since we can just drop the assumption that p'_i and p'_{-i} belong to the same payoff profile. It is not hard to see that fixating the proportion that $(a_{teacher}, BR(a_{teacher}))$ is played in combination with an arbitrary action profile allows us to find a larger value for ϵ' than in the case of repeatedly getting the worst possible payoff for our agent and the second best for the other agent. In other words, this is the largest possible range we can find for ϵ' that is small enough to ensure the property. Moreover, we can make the observation that it also holds that for every later iteration than $T_{-i}(\frac{1}{\epsilon'})$, the average payoff will not decrease. For a small enough value of ϵ' for the opponent (namely small enough such that there exists no other payoff in the payoff vector that is smaller than $\max_{a \in A_2} V_{-i}(a_{teacher}, a)$ and larger or equal than $\max_{a \in A_2} V_{-i}(a_{teacher}, a) - \epsilon'$) the opponent can do no better to maintain or increase the proportion k in which $BR(a_{teacher})$ is played. Thus, for a small enough value of ϵ for our agent, the proportion in which we receive $V_{teacher}$ is also maintained or increased. Since in the above

proof the calculation for ϵ' was based on achieving the worst possible payoff in the remaining proportion of rounds, it is impossible that our average payoff also drops lower; it is enough that the proportion in which $V_{teacher}$ is achieved remains constant or increases.

The final step is to prove the proposition. Using the earlier defined function T_{-i} and our found value for κ, we see that after $T_{-i}\left(\max(\frac{1}{\delta}, \frac{1}{\kappa * \epsilon})\right)$ time steps, we receive for any later time step an average payoff ϵ-close to $V_{teacher}$ with probability $1-\delta$. Since T_{-i} is a polynomial function, we also achieve this polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$ (notice that κ is just a game-constant). □

This proof concludes the teaching part of our strategy and enables us to move on to the following strategy.

4.2 Following strategy

For our following strategy, we have a number of possibilities, since many strategies achieve a best response value against pure strategies within polynomial time (for example Fictitious Play). However, we have chosen to select the AWESOME strategy to fill in this role, which is discussed in [2]. We stress that for the sake of understanding the message we are trying to convey in this paper no thorough understanding of AWESOME is required. The most important aspect of AWESOME is the fact that it has two key properties, namely AWESOME (1) converges to a Nash equilibrium in self-play, which, as we will prove later, cause our sequential teaching-following strategy to converge as well; and (2) converges to a best-response against arbitrary stationary opponents. The resulting teaching-following strategy will (more or less) also have this property. Unfortunately, proving targeted optimality against pure strategies when using AWESOME is not so easy as it may seem, and requires knowledge of valid schedules and the specific steps taken in the algorithm. Moreover, the exact amount of rounds needed in which we acquire targeted optimality is not relevant in the case of our algorithm, since we will play AWESOME for the rest of the game once we adopt it. Thus instead of giving the full proof, we give a brief proof outline.

PROPOSITION 3. *When using the AWESOME algorithm, for any $\delta > 0$ and $\epsilon > 0$, there exists a number of rounds τ, polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, such that for any number of rounds $t \geq \tau$ the strategy against an arbitrary pure strategy σ achieves average payoff of at least $V_{BR}(\sigma) - \epsilon$ with probability $1-\delta$, where $V_{BR}(\sigma)$ is the value of the best response against σ.*

The proof is heavily based on the fact that the observed distribution of play of the opponent is identical to the true distribution of play (contrary to mixed strategies). After every restart, AWESOME will first consider that the opponent is an equilibrium player. This hypothesis is refuted after a fixed amount of rounds, based on the monotonically decreasing closeness parameter (belonging to the schedule) that denotes the maximum allowed distance between distributions in the equilibrium playing phase, and it is based on the distance between the pure strategy distribution and the equilibrium strategy distribution. Afterwards AWESOME will consider that the opponents are stationary, which we will refer to as the stationary playing phase. First AWESOME will play a random action that either is a best response or not. In the first case, AWESOME will play this action for the rest of the game since it will never switch actions and thus the algorithm will never restart on behalf of itself nor the opponent. If this is not a best response, we will eventually switch actions after a fixed amount of rounds based on the number of players, the maximum number of actions, the payoff difference between our best and worst outcome in the game and our monotonically decreasing closeness parameter (belonging to the schedule) that denotes the maximum allowed distance between distributions in the stationary playing phase. If this function decreases fast enough, we will restart the algorithm. Using this information, we can find the number of restarts (and thus eventually the number of iterations) needed to ensure targeted optimality against pure strategies.

Using this proof we can immediately see that AWESOME is not only targeted optimality given the class of pure strategies, but also pure consistent. This property implies that by Definition 3 our eventual algorithm will be self-teachable.

4.3 Algorithm

The combination of the teacher and follower strategy gives us a new strategy that is able to teach pure strategy outcomes to adversaries that are willing to go along with this (pure consistent strategies) and is able to follow otherwise with a strategy we targeted in the teaching phase (in this case AWESOME). Observe that, as we will show later, the games in which this resulting strategy may work does not include games in which a mixed equilibrium strategy is required. The resulting algorithm is shown in 'Algorithm 1'. The input parameter $\langle(\epsilon^p, \delta^p), (\epsilon^s, \delta^s)\rangle$ should always be the same for any sequential targeted optimal algorithm: it contains a pair of ϵ and δ values for both the primary and secondary strategy. These parameters, as previously discussed, depict the closeness of the average payoff required and the probability that this will be reached. As we have seen, the lower the values, the longer the teaching/following process will take. The meaning of the κ variable can be found in Proposition 2 and the function T_{-i} is a polynomial function that estimates the rate of convergence of the opponent, and can effectively limit the target class to slow or fast learners (notice that we cannot make the teaching phase too short, since we also have to retain the self-teachability criterion). We again see a beautiful aspect of teaching arise: if the opponent is a slow learner, we might stop on teaching our opponent prematurely. Since the best response value against an arbitrary pure strategy is $V_{Minimax'}$, where $V_{Minimax'}$ is the pure strategy Minimax value, we know that this strategy is a teaching-following for all games in which $V_{teacher} \geq V_{Minimax'}$ (observe that $V_{Minimax'} \geq V_{Maximin}$, which settles our earlier conncern that repeatedly playing $a_{teacher}$ is not a best response in games in which $V_{teacher} < V_{Maximin}$

Algorithm 1 Sequential teaching-following strategy

Require: $\langle(\epsilon^p, \delta^p), (\epsilon^s, \delta^s)\rangle$
Ensure: $\epsilon^p > 0$, $\delta^p > 0$, $\epsilon^s > 0$, $\delta^s > 0$
1: $t \leftarrow 0$
2: **while** $(t < T_{-i}\left(\max(\frac{1}{\delta^p}, \frac{1}{\kappa * \epsilon^p})\right) \vee (\text{AvgPayoff} \geq V_{teacher} - \epsilon^p)$ **do**
3: $playaction(a_{teacher})$
4: $t \leftarrow t + 1$
5: **end while**
6: $playstrategy$(AWESOME)

Figure 3: Battle of the sexes

	Left	Right
Top	**3**,1	**0**,0
Bottom	**0**,0	**1**,3

Figure 4: Stackelberg game

	Left	Right
Top	**1**,0	**3**,2
Bottom	**2**,1	**4**,0

such as the matching pennies game shown in Figure 2). From a game-theoretic viewpoint, this result also makes perfect sense, since in this case we are indeed teaching a feasible and enforceable outcome, which then in turn can constitute a repeated Nash equilibrium as justified by the Folk theorem (for readers unfamiliar with this observation, we refer to [9, pp. 151-153] where this is very well explained). This observation can be used to prove convergence to a Nash equilibrium in self-play.

PROPOSITION 4. *In infinitely repeated games, our teaching-following algorithm, restricted to its set of games, will necessarily converge to a Nash equilibrium in self-play if it holds that ϵ_i^p and ϵ_{-i}^p are sufficiently small.*

PROOF. First let us define what 'sufficiently small' means: the values for ϵ_i^p and ϵ_{-i}^p are sufficiently small if for both players it holds that there exists no other payoff-profile in the payoff matrix for which both players receive a payoff of at least $V_{teacher} - \epsilon^p$. Notice that this is not a big restriction, since we can just compute this and pick such a small value for ϵ^p accordingly.
We distinguish the following 3 cases in self-play:

1. Both players maintain their primary strategy σ^p. This happens when both agents coincidently achieve an ϵ^p-close best response value while making false assumptions about their opponent. However, our demand for the values of ϵ^p ensure that we are indeed teaching $V_{teacher}$ and not settling on another payoff profile. Since we know that this outcome is both feasible and enforceable in our set of games, we know that we are playing a repeated Nash equilibrium.

2. Both players achieve their best response value when one player uses primary strategy σ^p while the other uses secondary strategy σ^s. Since both players are playing a best response to each other in these games, we know that σ^p is targeted optimal given σ^s and vice versa, which implies by Proposition 1 a Nash equilibrium.

3. Both players maintain their secondary strategy σ^s for the rest of the game. Convergence to a Nash equilibrium in this specific case is proven in [2].

□

Moreover, our algorithm more or less retains all the properties of AWESOME. For example, it can be easily shown that if the strategy of the opponent converges to a stationary strategy, our algorithm will converge to a best-response given this stationary strategy or we will achieve an average payoff ϵ^p-close to $V_{teacher}$.

Our teaching-following strategy enables us to teach a repeated Nash equilibrium which provably can be learned by a very broad class of opponents (contrary to just playing AWESOME) in efficient time and allows us to switch if the former fails. On top of the beneficial theoretical properties of our algorithm, we believe we can make our discussion of our algorithm even more convincing by looking at some specific games.

The following games are examples in which our algorithm is able to perform particularly well.

1. In the battle of the sexes game, shown in Figure 3, our algorithm is able to teach ('force') the (most) beneficial outcome of 3 to follower strategies that are willing to go along, while other strategies that are able to coordinate might reach a point on the Pareto boundary which is less beneficial (such as 1).

2. Our algorithm is able to signal repeated Nash equilibrium outcomes that are easy to learn by the opponent and can ensure greater payoff than the equilibrium of the stage game. This is the case with the Stackelberg game shown in 4 (with 'Stackelberg game' we do not mean the formal definition, but rather we refer to [9, p. 200] where they use this name to distinguish a particular simultaneous action Cournot game). In this particular game, our sequential teaching-following strategy is able to teach the outcome that will give our agent a payoff of 3, where as the equilibrium strategy of the stage game gives us a lower payoff of 2.

This section was mainly concerned with presenting an algorithm that is able to teach and follow with the use of our proposed criterion. In the next section we will take a step back to take a look at our criterion again, which will open the way for some general discussion.

5. GENERAL DISCUSSION

In this paper we used the notion of sequential targeted optimality to create a teaching-following criterion as a way of capturing both teaching and following behaviour in repeated games. However, some choices we made during the construction of our criterion could be made differently. An important choice we made was when we defined the notion of self-teachability. The only demand we had is that teaching and following behaviour are able to coordinate together, and that the teacher strategy sets itself apart from the follower strategy in terms of payoff guarantee in certain games. This definition can potentially imply that in some games what we understand as a 'teaching' strategy can conversely function as a 'following' strategy in other games. Since this definition still fully captures the coordination aspect of teaching and following this is not really a problem, but admittedly there might be something more to the broad meaning of a teaching strategy and the broad meaning of a following strategy. Another choice immediately becomes apparent when we define the beneficialness of the teaching part over the following part. We used payoff guarantees to define this beneficialness, which makes sense from the viewpoint of a conservative agent. On the other hand, expected payoff or maximal payoff guarantees also make sense when we consider for example greedy agents or risk-taking agents. We made this choice mainly because a minimal payoff guarantee allows us

to identify cases in which playing a strategy will necessarily lead to an enforceable outcome. But again we stress that this was nothing more than a choice.

Another important point of discussion is the fact that the notion is restricted to a set of games. By showing that in some games teaching strategies (other than our Maximin strategy) are not really feasible, we tried to make the point more clear that we really need this restriction. However, this restriction also has its problems. For example: what does it mean that a strategy is restricted to a set of games? Does it mean that the strategy is useless in other games? We have not really give an interpretation to this restriction. It becomes even more troublesome when the payoff matrices are not known. When do we know which strategy to use? We stress that this was never our intention to define; we are merely interested in defining the set of games in which 'it makes sense' to use such a strategy. The exact interpretation of this restriction is up to the creator of the strategy.

We also made a choice with the switching criterion in our definition of sequentially targeted optimality. As shown in [6], by smart use of the probability factors δ we can devise an algorithm that is targeted optimal simultaneously given different classes of opponents, instead of sequentially in our algorithm. If we would allow simultaneous optimization, it could lead to a potentially different definition of teaching and following.

As a final point of discussion, we note that our definition of sequentially teaching-following was not concerned with safety and convergence to Nash equilibria in self-play (although we have given conditions in which this can happen). We note that the latter is the least of our worries, since in a teacher and follower setting one might be less concerned about self-play. It is questionable why we even need to perform well given that we face ourselves, given that we are only concerned whether or not our opponent is a follower. The first point, a safety condition, is arguably more important. Any strategy should be safe to use, else we can just play our security strategy instead. However, we did not feel the need to include this in our criterion; this can simply be a separate criterion instead when devising a sequential teaching-following strategy.

6. FUTURE RESEARCH

There are many possibilities for future research. First of all, we would really like to see a sequential teaching-following strategy that uses Bully extended to the set of mixed strategies as its teaching strategy and for example AWESOME as its following strategy. This strategy targets in its teaching phase the set of consistent opponents (and not necessarily the *pure* consistent opponents) and its following phase the set of stationary opponents (and not necessarily pure strategies). However, we note that proving targeted optimality for AWESOME against stationary opponents can be tricky as it requires manipulation of many probability factors.

Another possible point of departure is to extend the notion of teaching-following to n-player games. In this particular case, we have to take into account the fact that our opponents might belong to different classes. The notion of targeted optimality has to be extended to cope with this fact. As shown in [8], checking if multiple opponents belong to a single class also becomes quite tricky, but is definitely an interesting direction to go in.

In this paper, our focus was on teaching and following in a sequential way. But it might be perfectly possible to teach and follow in different ways (such as periodic). This could be a direction for possible future research. For example, dropping sequentially optimality in favour of simultaneous optimality might cause interesting behaviour. In this case, if a solution of the game is reached, the agent still needs to worry about the fact whether or not the opponent might belong to a different target class. This might open up the way to new insights concerning the subject.

Another quite different point of departure is to investigate the exact nature of teaching and following. We used the self-teachability criterion, but we also mentioned in the introduction that teacher and follower strategies also have 'certain properties' that allow us to identify them as such (for example Bully and Godfather can be reasonably understood as teacher strategies). The challenge becomes to devise a formal notion of when a strategy is a teaching strategy and when a strategy is a following strategy.

A last point for possible future research we like to discuss is in settings where the payoff matrices (initially) are not known. If the payoff matrix of the adversary stays hidden throughout, it can be troublesome for teaching strategies, since (arguably) they rely heavily on the payoff matrix of the opponent. In these settings, it might be interesting to investigate how teaching and following can still arise.

7. REFERENCES

[1] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[2] V. Conitzer and T. Sandholm. AWESOME: A General Multiagent Learning Algorithm that Converges in Self-Play and Learns a Best Response against Stationary Opponents. In *Proceedings of the 20th International Conference on Machine Learning*, pages 83–90, 2006.

[3] J. W. Crandall. Learning to teach and follow in repeated games. In *AAAI Workshop on Multiagent Learning*, July 2005.

[4] D. Fudenberg and D. K. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19(5-7):1065–1089, 1995.

[5] M. L. Littman and P. Stone. Implicit negotiation in repeated games. In *Proceedings of The Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001)*, pages 393–404, 2001.

[6] R. Powers and Y. Shoham. New criteria and a new algorithm for learning in multi-agent systems. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, volume 17, 2004.

[7] R. Powers and Y. Shoham. Learning against opponents with bounded memory. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 817–822, 2005.

[8] R. Powers, Y. Shoham, and T. Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. In *Machine Learning*, volume 67, pages 45–76, 2007.

[9] Y. Shoham and L. Brown. *Multiagent Systems: Algorithmic, Game-Theoretic and Logical Foundations*. Cambridge University Press, 2009.

On the quality and complexity of Pareto equilibria in the Job Scheduling Game

Leah Epstein
Department of Mathematics
University of Haifa
31905 Haifa, Israel
lea@math.haifa.ac.il

Elena Kleiman
Department of Mathematics
University of Haifa
31905 Haifa, Israel
elena.kleiman@gmail.com

ABSTRACT

In the well-known scheduling game, a set of jobs controlled by selfish players wishes each to minimize the load of the machine on which it is executed, while the social goal is to minimize the makespan, that is, the maximum load of any machine. We consider this problem on the three most common machines models, identical machines, uniformly related machines and unrelated machines, with respect to both weak and strict Pareto optimal Nash equilibria. These are kinds of equilibria which are stable not only in the sense that no player can improve its cost by changing its strategy unilaterally, but in addition, there is no alternative choice of strategies for the entire set of players where no player increases its cost, and at least one player reduces its cost (in the case of strict Pareto optimality), or where all players reduce their costs (in the case of weak Pareto optimality).

We give a complete classification of the social quality of such solutions with respect to an optimal solution, that is, we find the Price of Anarchy of such schedules as a function of the number of machines, m. In addition, we give a full classification of the recognition complexity of such schedules.

Categories and Subject Descriptors

K.6.0 [**Management of Computing and information Systems**]: General—*Economics*; F.2.2 [**Nonnumerical Algorithms and Problems**]: [Sequencing and scheduling]

General Terms

Algorithms, Economics, Theory

Keywords

Economic paradigms: Economically-motivated agents, Game Theory (cooperative and non-cooperative), Price of Anarchy, Job Scheduling

1. INTRODUCTION

The rise of the Internet as a global platform for communication, computation, and commerce brought up the necessity to reconsider the prevalent paradigm in system design which assumes a central authority which constructs and manages the network and its participants, with a purpose of optimizing a global social objective. Designing a protocol intended for use in a global telecommunication network such as the Internet, we have to take into account that it consists of multiple independent and self-interested users, or players, which strive to optimize their private objective functions, also known as individual costs. In networks of such scale and complexity and in presence of raw economic competition between the parties involved, there is no possibility to introduce a single regulatory establishment enforcing binding commitments on the players.

Obviously, such collective behavior often leads to sub-optimal performance of the system, which is highly undesirable. In light of the above, there is an increased need to design efficient protocols that motivate self-interested agents to cooperate. Here "cooperation" may be defined as any enforceable commitment that makes it rational for the self interested players to choose a given strategic profile. In the settings in discussion, any meaningful agreement between the players must be self-enforcing. When deciding which particular strategy profile to offer for the users, the first and most basic requirement one has to consider is its stability, in a sense that no player would have an interest to unilaterally defect from this profile, given that the other players stick to it. This is consistent with the notion of Nash equilibrium (NE) [25], which is a widely accepted concept of stability in non-cooperative game theory. The second requirement is that the profile must be efficient. A fundamental concept of efficiency considered in economics is the Pareto efficiency, or Pareto optimality [24]. This efficiency criterion assures that it is not possible for a group of players to change their strategies so that every player is better off (or no worse off) than before.

One may justifully argue that Nash stability and Pareto optimality should be minimal requirements for any equilibrium concept intended to induce self-enforceability in presence of selfishness.

There are even stronger criteria for self-enforceability, requiring fairness in terms of fair competition without coalitions (like cartels and syndicates), and demanding from the profile to be resilient to groups (or coalitions) of players willing to coordinate their decisions, in order to achieve mutual beneficial outcomes. This is compatible with the definition of Strong Nash equilibrium (SNE) [3]. However, this requirement is sometimes too strong that it excludes many reasonable profiles.

We therefore restrict ourselves to profiles that satisfy the requirements of Nash stability and Pareto efficiency. In a sense, Pareto optimal Nash equilibria can be considered as intermediate concepts between Nash and Strong Nash equilibria; One may think of a Pareto optimal equilibrium as being stable under moves by single players or the grand coalition of all players, but not necessarily arbitrary coalitions. We distinguish between two types of Pareto efficiency. In a *weakly* Pareto optimal Nash equilibrium (WPO-NE) there is no alternative strategy profile beneficial for all players. A

Cite as: On the quality and complexity of Pareto equilibria in the Job Scheduling Game, Leah Epstein, Elena Kleiman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 525-532.

strictly Pareto optimal Nash equilibrium (SPO-NE) is also stable against deviations in which some players do not benefit but are also not worse off and at least one player improves his personal cost. Obviously, any *strictly* Pareto optimal equilibrium is also *weakly* Pareto optimal, but not wise-versa.

In this paper we consider strict and weak Pareto optimal Nash equilibria for scheduling games on the most common three machine models in the setting of pure strategies. This class of games is particularly important to our discussion as it models a great variety of problems in modern networks. Example applications include bandwidth sharing in ATM networks [7], market-based protocols for scheduling or task allocation [28], and congestion control protocols [18].

1.1 Model and Notation

We now define the general job scheduling problem. There are n jobs $J = \{1, 2, \ldots, n\}$ which are to be assigned to a set of m machines $M = \{M_1, \ldots, M_m\}$. We study three models of machines, that differ in the relation between the processing times of jobs on different machines. In the most general model of *unrelated* machines, job $1 \leq k \leq n$ has a processing time of p_{ik} on machine M_i, i.e., processing times are machine dependent. In the *uniformly related* (or *related*) machine model, each machine M_i for $1 \leq i \leq m$ has a speed s_i and each job $1 \leq k \leq n$ has a positive size p_k. The processing time of job k on machine M_i is then $p_{ik} = \frac{p_k}{s_i}$. If $p_{jk} = p_{j'k} = p_k$ for each job k and machines M_i and $M_{i'}$, the machines are called *identical* (in which case it is typically assumed the all speed are equal to 1).

An assignment or schedule is a function $\mathcal{A} : J \rightarrow M$. The load of machine M_i, which is also called the delay of this machine, is $L_i = \sum_{k:\mathcal{A}(k)=M_i} p_{ik}$. The cost, or the *social cost* of a schedule is the maximum delay of any machine, also known as the *makespan*, which we would like to minimize.

The job scheduling game JS is characterized by a tuple $JS = \langle N, (\mathcal{M}_k)_{k\in N}, (c_k)_{k\in N}\rangle$, where N is the set of atomic players. Each selfish player $k \in N$ controls a single job and selects the machine to which it will be assigned. We associate each player with the job it wishes to run, that is, $N = J$. The set of strategies $\mathcal{M}_k$ for each job $k \in N$ is the set M of all machines. i.e. $\mathcal{M}_k = M$. Each job must be assigned to one machine only. Preemption is not allowed. The outcome of the game is an assignment $\mathcal{A} = (\mathcal{A}_k)_{k\in N} \in \times_{k\in N} M_k$ of jobs to the machines, where $\mathcal{A}_k$ for each $1 \leq k \leq n$ is the index of the machine that job k chooses to run on. Let $\mathcal{S}$ denote the set of all possible assignments.

The cost function of job $k \in N$ is denoted by $c_k : \mathcal{S} \rightarrow \mathbb{R}$. The cost c_k^i charged from job k for running on machine M_i in a given assignment $\mathcal{A}$ is defined to be the load observed by machine i in this assignment, that is $c_k(i, \mathcal{A}_{-k}) = L_i(\mathcal{A})$, when $\mathcal{A}_{-k} \in \mathcal{S}_{-k}$; here $\mathcal{S}_{-k} = \times_{j\in N\setminus\{k\}}\mathcal{S}_j$ denotes the actions of all players except for player k. The goal of the selfish jobs is to run on a machine with a load which is as small as possible. Similarly, for $K \subseteq N$ we denote by $\mathcal{A}_K \in \mathcal{S}_{-K}$ the set of strategies of players outside of K in a strategy profile $\mathcal{A}$, when $\mathcal{S}_{-K} = \times_{j\in N\setminus K}\mathcal{S}_j$ is the action space of all players except for players in K. The social cost of a strategy profile $\mathcal{A}$ is denoted by $SC(\mathcal{A}) = \max_{1\leq k\leq n} c_k(\mathcal{A})$.

We will next provide formal definitions of Nash, Weak/Strict Pareto Nash and Strong Nash equilibria in the job scheduling game, using the notations given above.

DEFINITION 1. *(Nash equilibrium) A strategy profile $\mathcal{A}$ is a (pure) Nash equilibrium (NE) in the job scheduling game JS if for all $k \in N$ and for any strategy $\bar{\mathcal{A}}_k \in M$, $c_k(\mathcal{A}_k, \mathcal{A}_{-k}) \leq c_k(\bar{\mathcal{A}}_k, \mathcal{A}_{-k})$.*

It was shown that job scheduling games always have (at least one) pure Nash equilibrium [15, 11]. We denote the set of Nash equilibria of an instance G of the job scheduling game by NE(G).

DEFINITION 2. *(Strong Nash equilibrium) A strategy profile $\mathcal{A}$ is a Strong Nash equilibrium (SNE) in the job scheduling game JS if for every coalition $\phi \neq K \subseteq N$ and for any set of strategies $\bar{\mathcal{A}}_K \in \times_{j\in K}\mathcal{M}_j$ of players in K, there is a player $i \in K$ such that $c_i(\bar{\mathcal{A}}_K, \mathcal{A}_{-K}) \geq c_i(\mathcal{A}_K, \mathcal{A}_{-K})$.*

Existence of Strong Nash equilibrium in job scheduling games was proved in [1]. We denote the set of Strong Nash equilibria of an instance G of the job scheduling game by SNE(G).

Clearly SNE(G)$\subseteq$NE(G), as coalitions of size 1 can not improve by changing their strategy.

DEFINITION 3. *(Weak/Strict Pareto optimal profile) A strategy profile $\mathcal{A}$ is weakly Pareto optimal (WPO) if there is no strategy profile $\bar{\mathcal{A}}$ s.t. for all $k \in N$, $c_k(\bar{\mathcal{A}}) < c_k(\mathcal{A})$.*

A strategy profile $\mathcal{A}$ is strictly Pareto optimal (SPO) if there is no strategy profile $\bar{\mathcal{A}}$ and $k^* \in N$ s.t. for all $k \in N\setminus k^*$, $c_k(\bar{\mathcal{A}}) < c_k(\mathcal{A})$ and $c_{k^*}(\bar{\mathcal{A}}) \leq c_{k^*}(\mathcal{A})$.

We denote by SPO(G) and WPO(G), respectively, the sets of strictly and weakly Pareto optimal profiles of an instance G of the job scheduling game. Clearly, SPO(G)$\subseteq$ WPO(G).

A strategy profile $\mathcal{A} \in$ NE(G)$\cap$ WPO(G) is called Weak Pareto optimal Nash equilibrium (WPO-NE), and a strategy profile $\mathcal{A} \in$NE(G) $\cap$ SPO(G) is called Strict Pareto optimal Nash equilibrium (SPO-NE), and these are the profiles that we focus on.

We note that every strong equilibrium is also weakly Pareto optimal, as the requirement in Definition 2 applies to the grand coalition of all players. Hence SNE(G)$\subseteq$ WPO(G). The existence of Strong Nash equilibria in job scheduling games assures the existence of weak Pareto optimal Nash equilibria.

On the other hand, in general, neither Nash equilibria nor Strong Nash equilibria are necessarily strictly Pareto optimal. Existence of strict Pareto optimal Nash equilibria in scheduling games (among others) was proved in [16].

An important issue concerns the quality of these solution concepts. As there is a discrepancy between the private goals of the players and the global social goal, we would like to measure the loss in the performance of the system as it is reflected by the closeness of the costs of these concepts to the cost of the optimal solution, when the accepted methodology is worst-case approach.

The quality measures which consider Nash equilibria are the Price of Anarchy introduced by Koutsoupias and Papadimitriou [20] and the more optimistic Price of Stability suggested by Anshelevich et al. [2], which are defined as the worst-case ratio between the social cost of the worst/best Nash equilibrium to the social cost of an optimal solution, which is denoted by OPT. Formally,

DEFINITION 4. *(Price of Anarchy and Stability) The Price of Anarchy (PoA) of the job scheduling game JS is defined by*

$$PoA(JS) = \sup_{G\in JS} \sup_{\mathcal{A}\in NE(G)} \frac{SC(\mathcal{A})}{\text{OPT}(G)}.$$

If instead we consider the best Nash equilibrium of every instance, this leads to the definition of the Price of Stability (PoS):

$$PoS(JS) = \sup_{G\in JS} \inf_{\mathcal{A}\in NE(G)} \frac{SC(\mathcal{A})}{\text{OPT}(G)}.$$

This concept is applied analogously to Strong Nash equilibria as well as to weakly/strictly Pareto optimal Nash equilibria yielding

the Strong Price of Anarchy $SPoA(JS)$ and the Strong Price of Stability $SPoS(JS)$ as well as the weak and strict Pareto Prices of Anarchy $WPO\text{-}PoA(JS)$, $SPO\text{-}PoA(JS)$ and Stability $WPO\text{-}PoS(JS)$, $SPO\text{-}PoS(JS)$. By definition, it is clear that $SPoA(JS) \leq WPO\text{-}PoA(JS) \leq PoA(JS)$. As any *strictly* Pareto optimal NE is also a *weakly* Pareto optimal NE, it must be the case that $WPO\text{-}PoA(JS) \geq SPO\text{-}PoA(JS)$. However, we can show that in the job scheduling game there is no immediate relation between the $SPO\text{-}PoA(JS)$ and the $SPoA(JS)$, as there are Strong Nash equilibria that are not *strictly* Pareto optimal, while there are *strictly* Pareto optimal Nash equilibria that are not strong equilibria.

Some natural questions in this context are whether the Pareto Prices of Anarchy are significantly smaller than the standard Price of Anarchy, whether the weak Pareto Price of Anarchy is much larger than the Strong Price of Anarchy, and finally, whether there is any relation between the Strong Price of Anarchy and the strict Pareto Price of Anarchy. In other words, does the requirement that the equilibrium must be Pareto optimal leads to greater efficiency, and does the further demand that the equilibrium must be stable against arbitrary coalitions is helpful.

1.2 Related work and our contribution

Pareto efficiency of resource assignments is a well referred issue in economics, especially in welfare economics. Pareto efficiency is a highly desirable trait, however Dubey [9] has shown that Nash equilibria may generally be Pareto inefficient based on the difference between the conditions to be satisfied by Nash equilibria and those to be satisfied by Pareto optima.

Job scheduling is a classical problem in combinatorial optimization. The analysis of job scheduling in the algorithmic game theory context was initiated by Koutsoupias and Papadimitriou in their seminal work [20], which was followed by many others (see e.g. [8, 22, 1, 13]). In our overview of the known results we will limit our discussion only to results concerning pure strategies. We will begin with the results on quality measures that concern Nash equilibria of the game. For m identical machines, the PoA is $2 - \frac{2}{m+1}$ which can be deduced from the results of [14] (the upper bound) and [26] (the lower bound). For related machines the PoA is $\Theta(\frac{\log m}{\log \log m})$ [19, 8, 20]. In the model of unrelated machines the PoA is unbounded [5], which holds already for two machines. From the results of [11] it is evident that in all three models the PoS is 1. The study of quality measures that concern Strong Nash equilibria of this game was initiated by Andelman at el. [1]. For identical machines, they proved that the $SPoA$ equals the PoA, which in turn equals $2 - \frac{2}{m+1}$. For related machines, Fiat et al. [1] showed that the $SPoA$ is $\Theta(\frac{\log m}{(\log \log m)^2})$. Surprisingly, the $SPoA$ for this problem is bounded by the number of machines m, as shown in [13], and this is tight [1]. Andelman at el. also showed that $SPoS$ is 1.

The previous work on Pareto efficiency of Nash equilibria in algorithmic game theory was mainly concerned with weak Pareto equilibria, probably since a solution which is not weakly Pareto optimal is clearly unstable. A textbook in economics states the following: "The concept of Pareto optimality originated in the economics equilibrium and welfare theories at the beginning of the past century. The main idea of this concept is that society is enjoying a maximum ophelimity when no one can be made better off without making someone else worse off" [21]. Thus the strict Pareto is a stronger and more meaningful efficiency notion, as it captures an important aspect of human social behavior. Another issue is that the weak Pareto implies that everyone prefers some assignment to any other. In reality, such unanimity of preferences among all persons is very rare. To conclude, both concepts are important, and we focus on both of them in this work.

Pareto optimality of Nash equilibria has been studied in the context of congestion games, see Chien and Sinclair [6] and Holzman and Law-Yone [17]. The former gave conditions for uniqueness and for weak and strict Pareto optimality of Nash equilibria, and the latter characterized the weak Pareto Prices of Anarchy and Stability. The existence, and complexity of recognition and computation of weak Pareto Nash equilibria in congestion games was studied recently by Hoefer and Skopalik in [27].

In [16] Harks at el. show that a class of games that have a *Lexicographical Improvement Property* (which our game indeed has) admits a generalized strong ordinal potential function. They use this to show existence of Strong Nash equilibria with certain efficiency and fairness properties in these games, strict Pareto efficiency included. They do so by arguing that a player wise cost-lexicographically minimal assignment is also strictly Pareto optimal (and so it is optimal w.r.t. the social goal function as well).

Weak Pareto Nash equilibria in routing and job scheduling games were considered recently in [4] by Aumann and Dombb. As a measure for quantifying the distance of a best/worst Nash equilibrium from being weakly Pareto efficient, they use the smallest factor by which any player improves its cost when we move to a different strategy profile, which they refer to as "Pareto inefficiency". They do not consider however the quality of Pareto optimal Nash equilibria with respect to the social goal.

Among other results, it is shown in [4] that any Nash equilibrium assignment is necessarily weakly Pareto optimal for both identical and related machines. Moreover, for any machine model, any assignment which achieves the social optimum must be weakly Pareto optimal. One such assignment is one whose sorted vector of machine loads is lexicographically minimal is necessarily weakly Pareto optimal (see also [1, 11]). Milchtaich [23] has proved related results for the case of non-atomic players, where the processing time of each player is negligible compared to the total processing time.

We consider these issues for SPO-NE assignments. We show that while the property of identical machines remains true, this is not the case for related machines, that is, not every Nash equilibrium assignment is strict Pareto optimal. For unrelated machines, while there always exist an assignment which is a social optimum and a SPO-NE, assignments with lexicographically minimal sorted vector of machine loads are not necessarily strictly Pareto optimal. In this paper we fully characterize the weak and strict Pareto Prices of Anarchy of the job scheduling game in cases of identical, related and unrelated machines. The characterization of the Prices of Stability follows from previous work as explained above.

Next, we consider the complexity of recognition of weak and strict Pareto optimality of NE. Note that the recognition of NE can be done in polynomial time for any machine model by examining potential deviations of each job. As for strong equilibria, it was shown by Feldman and Tamir [12] that it is NP-hard to recognize an SNE for $m \geq 3$ identical machines and for $m \geq 2$ unrelated machines. For two identical machines, they showed that any NE is a SNE, so recognition can be done in polynomial time (for $m \geq 3$, it was shown in [1] that not every NE is a SNE). For the only remaining case of two related machines, it was shown [10] that recognition is again NP-hard. We show that the situation for Pareto optimal equilibria is slightly different. In fact, recognition of WPO-NE or SPO-NE can be done in polynomial time for identical machines and related machines. For unrelated machines, we show that the recognition of WPO-NE is NP-hard in the strong sense and the recognition of SPO-NE is NP-hard.

We reflect upon the differences between the results for weak and strict Pareto equilibria also compared to strong equilibria, and make conclusions regarding the relations between the quality measures in this game. See Table 1 for a summary of the results.

2. PARETO PRICES OF ANARCHY IN THE JOB SCHEDULING GAME

2.1 Identical and Related machines

A result from [4] shows that any NE schedule for identical and related machines is weakly Pareto optimal. This result implies that $WPO\text{-}PoA(JS) = PoA(JS)$. For the case of identical machines, they give an even stronger result: every schedule where every machine receives at least one job is weakly Pareto optimal. Note that if $n < m$, then a schedule is weakly Pareto optimal if and only if at least one machine has a single job (to obtain strict Pareto optimality for this case, or to obtain a NE, each job needs to be assigned to a different machine).

In the strict Pareto case, while the general result for identical machines still holds, and the set of NE schedules is equal to the set of SPO-NE schedules for identical machines (as we prove next), it is not necessarily true for related machines. We exhibit an example of a schedule which is a NE but it is not strictly Pareto optimal.

Consider a job scheduling game with two related machines of speeds 1,2 and two jobs of size 2. There are two types of pure NE schedules: in the first one, both jobs are assigned to the fast machine, and in the other one job runs on each machine. The first one is not a SPO-NE, as switching to a schedule of the second type strictly reduces the cost for one of the jobs, while not harming the other. Moreover, the sorted machine load vector of the first type of schedules is $(2, 0)$, while the load vector of the second type is $(2, 1)$, so the schedule with the lexicographically minimal machine load vector is not a SPO-NE (even though it is a SNE).

This difference in the results for related machines is explained by the fact that conditions for weak Pareto allow Pareto improvements where not all jobs strictly improve while the strict Pareto does not. If a NE schedule has an empty machine, and a job arrived to such a machine as a result of a deviation to a different schedule, where all jobs strictly reduce their costs, then the reduction in the cost of this job contradicts the original schedule being a NE. However, if the job only needs to maintain its previous cost, then there is no contradiction.

We will prove the following theorem, extending the result of [4] which will allow us to claim that for identical machines, $SPO\text{-}PoA(JS) = PoA(JS)$.

THEOREM 5. *Any schedule for identical machines, where no machine is empty, is strictly Pareto optimal.*

This is a stronger result than the one in [4], since it deals with strict Pareto. The idea of the proof goes along the lines of [4], but we need to modify it so that it applies for the stronger conditions of strict Pareto optimal schedules. First we prove the following property, which we will also use to characterize the $WPO\text{-}PoA$ for related machines.

THEOREM 6. *Consider a schedule X that is not a* SPO-NE*, and denote the set of non-empty machines (which receive at least one job) in X by μ_X. Let Y be a different schedule where no job has a larger cost than it has in X and at least one job has a smaller cost. Denote the set of non-empty machines in Y by μ_Y. Then,*

$$\sum_{i\in\mu_X} s_i < \sum_{i\in\mu_Y} s_i,$$

where s_i is the speed of machine i.

PROOF. Consider a transition from schedule X to schedule Y, and denote by x_i^j the sum of the sizes of jobs that are moved from machine $i \in \mu_X$ to machine $j \in \mu_Y$ (for $j = i$, this gives the sum of sizes of jobs that are assigned to this machine in both schedules). Let ℓ_t, for $t \in \mu_X$, be the sum of sizes of jobs that run on machine t in X, and let ℓ'_t, for $t \in \mu_Y$, be the sum of sizes of jobs that run on machine t in Y. We extended the definition so that if $t \notin \mu_X$, then $\ell_t = 0$, and if $t \notin \mu_Y$, then $\ell'_t = 0$.

Consider the total sum of sizes of jobs assigned to a machine in X or in Y, then the following claim holds:

CLAIM 7. *For every $i \in \mu_X$, $\sum_{j\in\mu_Y} x_i^j = \ell_i$, or $\sum_{j\in\mu_Y} \frac{x_i^j}{\ell_i} = 1$. For every $j \in \mu_Y$, $\sum_{i\in\mu_X} x_i^j = \ell'_j$, or $\sum_{i\in\mu_X} \frac{x_i^j}{\ell'_j} = 1$.*

By the definition of the costs in Y compared to X, we get that:

CLAIM 8. *If $x_i^j > 0$, then $\frac{\ell'_j}{s_j} \le \frac{\ell_i}{s_i}$, and there exist $i \in \mu_X$,$i \in \mu_Y$ such that $\frac{\ell'_j}{s_j} < \frac{\ell_i}{s_i}$.*

The following also holds:

CLAIM 9. *For every $i \in \mu_X$, $j \in \mu_Y$: $\frac{x_i^j}{\ell_i} \le \frac{s_j}{s_i} \cdot \frac{x_i^j}{\ell'_j}$, and there exist i, j such that $\frac{x_i^j}{\ell_i} < \frac{s_j}{s_i} \cdot \frac{x_i^j}{\ell'_j}$.*

PROOF. As $i \in \mu_X$, $\ell_i > 0$, as $j \in \mu_Y$, $\ell'_i > 0$. If $x_i^j > 0$, it is derived from Claim 8, if $x_i^j = 0$ it holds trivially. Since there is at least one job for which the cost in Y is strictly smaller than its cost in X, then the second property must hold. □

Summing up the inequalities in Claim 9 over all $j \in \mu_Y$, in combination with Claim 7, we get that for any $i \in \mu_X$:

$1 = \sum_{j\in\mu_Y} \frac{x_i^j}{\ell_i} \le \sum_{j\in\mu_Y} \frac{s_j}{s_i} \cdot \frac{x_i^j}{\ell'_j}$, where there is at least one $i \in \mu_X$ for which this inequality is strict. Equivalently, $s_i \le \sum_{j\in\mu_Y} \frac{x_i^j \cdot s_j}{\ell'_j}$. Summing up the last inequality over all $i \in \mu_X$ combined with the fact that for some i this inequality is strict, changing the order of summation, and using Claim 7 we get: $\sum_{i\in\mu_X} s_i < \sum_{i\in\mu_X} \sum_{j\in\mu_Y} \frac{x_i^j \cdot s_j}{\ell'_j} = \sum_{j\in\mu_Y} \sum_{i\in\mu_X} \frac{x_i^j \cdot s_j}{\ell'_j} = \sum_{j\in\mu_Y} s_j$, which concludes our proof.

We now return to the proof of Theorem 5.

PROOF. We show that any schedule X for identical machines where μ_X=M is strictly Pareto optimal. Assume by contradiction that this is not the case, and hence there exists a different schedule Y where at least one job improves, while all the other jobs are not worse off. As the machines in question are identical, $s_1 = s_1 = \ldots = s_m$ holds, thus $\sum_{i\in\mu_X} s_i = m$ and $\sum_{j\in\mu_Y} s_j \le m$. By Theorem 6 we get that $m = \sum_{i\in\mu_X} s_i < \sum_{j\in\mu_Y} s_j \le m$, which is a contradiction, and we conclude that such Y cannot exist. □

COROLLARY 10. *Every schedule on identical machines which is a* NE *is also a* SPO-NE*. Thus in this case $SPO\text{-}PoA$=$WPO\text{-}PoA$=PoA.*

PROOF. Consider a NE schedule. If there is an empty machine, then each machine has at most one job (otherwise, if some machine has two jobs then any of them can reduce its cost by moving to an empty machine), and thus each job has the smallest cost that it can have in any schedule. Otherwise, the property follows from Theorem 5. □

	# of machines	Strict Pareto			Weak Pareto		
		SPO-PoA	SPO-PoS	Recognition	WPO-PoA	WPO-PoS	Recognition
identical	m	$2-\frac{2}{m+1}$	**1** [16]	P	$2-\frac{2}{m+1}$	**1** [1, 4]	P
related	m	$\Theta(\frac{\log m}{\log\log m})$	**1** [16]	P	$\Theta(\frac{\log m}{\log\log m})$	**1** [1, 4]	P
unrelated	$m=2$	2	**1** [16]	NP-hard	2	**1** [1, 4]	NP-hard
	$m\geq 3$	m			∞		

Table 1: Summary of Results

We next consider related machines and prove that the three measures are equal in this case as well.

THEOREM 11. *In the job scheduling game on related machines $SPO\text{-}PoA$=$WPO\text{-}PoA$=PoA.*

PROOF. As any SPO-NE is also a WPO-NE, and every WPO-NE is a NE, the following sequence of inequalities holds: $SPO\text{-}PoA \leq WPO\text{-}PoA \leq PoA$. We will prove that this is actually a sequence of equalities. It is enough to prove that $PoA \leq SPO\text{-}PoA$. We will do it by showing that the lower bound example for the PoA given in [8] is also a lower bound for the $SPO\text{-}PoA$, by proving that it is strictly Pareto optimal.

For completeness, we first present the lower bound of [8]. Consider a job scheduling game on m related machines. The machines are partitioned into $k+1$ groups, each group j ,$0 \leq j \leq k$ has N_j machines. The sizes of the groups are defined in inductive manner: $N_k = \Theta(\sqrt{m})$, and for every $j < k$: $N_j = (j+1)\cdot N_{j+1}$ (and thus $N_0 = k! \cdot N_k$). The total number of machines $m = \sum_{j=0}^{k} N_j = \sum_{j=0}^{k} \frac{k!}{(k-j)!} \cdot N_k$. It follows that $k \sim \frac{\log m}{\log\log m}$. The speed of each machine in group j is $s_j = 2^j$.

A schedule is defined as follows: each machine in group j has j jobs, each with size 2^j. Each such job contributes 1 to the load of its machine. The load of each machine in group N_j is then j, and therefore the makespan which is accepted on the machines in group N_k is k. Note that all the machines in group N_0 are empty.

We denote this schedule by X. It was proven in [8] that X is a pure NE. We claim that it is also strictly Pareto optimal.

CLAIM 12. *X is strictly Pareto optimal.*

PROOF. Assume by contradiction that X is not a SPO-NE, so there exists another schedule Y where at least one job improves, and all the other jobs are not worse off. Observe that all the machines in group N_0 necessarily remain empty in Y; each job that runs on a machine in group N_j for $1 \leq j \leq k$ pays a cost of j in X, and if it is assigned on a machine from group N_0 in Y it has to pay a cost of 2^j, and $2^j > j$ for $j \geq 1$, which makes it strictly worse off. This means that $\mu_Y \subseteq \mu_X$. On the other hand, according to Theorem 6 which we proved earlier, $\sum_{i\in\mu_X} s_i < \sum_{i\in\mu_Y} s_i$ must hold, and we get a contradiction. Hence, the schedule in this example is strictly Pareto optimal. □

An optimal schedule has a makespan of 2. To obtain such a schedule, we move all jobs from machines in N_j ($j \cdot N_j$ jobs, each of size 2^j) to machines in N_{j-1}, for $1 \leq j \leq k$. Every machine gets at most one job, and the load on all machines is less or equal to $\frac{2^j}{2^{j-1}} = 2$. The $SPO\text{-}PoA$ is therefore $\Omega(\frac{\log m}{\log\log m})$.

We conclude that $SPO\text{-}PoA$=$WPO\text{-}PoA$=PoA.

It was proved in [13] that schedule X is not a SNE, as a coalition of all k jobs from a machine in group N_k with 3 jobs from each of k different machines from group N_{k-2} can jointly move in a way that reduces the costs of all its members. In addition to determining the SPO-NE, this example illustrates the point that in the job scheduling game not every SPO-NE is necessarily a SNE. We saw that for related machines, the $SPO\text{-}PoA$=$WPO\text{-}PoA$ are the same as the PoA, while the $SPoA$ is lower.

2.2 Unrelated machines

We saw that already for related machines, not every SNE is a SPO-NE and vice versa. However, the results which we find for the $SPO\text{-}PoA$ on unrelated machines are similar to those which are known for the $SpoA$, that is, the $SPO\text{-}PoA$ is equal to m for any number of machines m. Interestingly, the $WPO\text{-}PoA$ for the setting $m = 2$ is exactly 2, like the $SPO\text{-}PoA$, but for $m \geq 3$ it is unbounded like the PoA.

THEOREM 13. *There exists a job scheduling game with 2 unrelated machines, such that $WPO\text{-}PoA \geq 2$. For any, $m \geq 3$ there exists a job scheduling game with m unrelated machines, such that $WPO\text{-}PoA$ is unbounded.*

PROOF. Consider a job scheduling game with two unrelated machines and two jobs, where $p_{11} = p_{21} = p_{12} = 1$ and $p_{22} = 2$. A schedule where job 1 is assigned to M_1 and job 2 is assigned to M_2 with a makespan of 2 is a WPO-NE; No job would benefit from moving to a different machine, and job 1 will not profit by switching to a different schedule. In an optimal schedule for this game, job k, $k \in \{1, 2\}$ is assigned to M_k, and the makespan is 1. We get that $WPO\text{-}PoA \geq 2$.

Now, consider a job scheduling game with $m \geq 3$ unrelated machines and $n = m$ jobs, where for each job k, $1 \leq k \leq m$: $p_{kk} = \varepsilon$, and $p_{jk} = 1$ for all $j \neq k$, for some arbitrary small positive ε. A schedule where job 1 is assigned to run on M_1, job m is assigned to M_2 and each job k for $2 \leq k \leq m-1$ is assigned to M_{k+1}, is a WPO-NE. It is weakly Pareto optimal since job 1 cannot decrease its cost by changing to any other assignment. The only way that another job could decrease its cost would be by moving to the machine where its cost is ε, but the load on all those machines is 1. Therefore, the schedule is a NE. The makespan of this schedule is 1. An optimal schedule for this game, where each job $1 \leq k \leq m$ is assigned to machine M_k, has a makespan of ε. In total, we have $WPO\text{-}PoA \geq \frac{1}{\varepsilon}$, which is unbounded letting ε tend to zero. □

We next prove a matching upper bound for $m = 2$.

THEOREM 14. *For any job scheduling game with 2 unrelated machines, $WPO\text{-}PoA \leq 2$.*

PROOF. Consider a schedule on two unrelated machines which is a WPO-NE. Without loss of generality, assume that the load of M_1 is not larger than the load of M_2, and denote the loads of the machines are by L_1 and L_2, respectively. The makespan of this schedule is then L_2. We show $L_2 \leq 2\text{OPT}$. We first show $L_1 \leq$

OPT. If $L_2 \geq L_1 >$ OPT then an optimal schedule has the property that every job has a smaller cost in it than it has in the current schedule (a cost of at most OPT $< L_1 \leq L_2$), in contradiction to the fact that this schedule is a WPO-NE.

To complete the proof, we upper bound L_2. If $L_2 \leq$ OPT, then we are done, otherwise, $L_2 >$ OPT, and there must exist a job k assigned to M_2 which is assigned to M_1 in an optimal schedule (since the load resulting from jobs assigned to M_2 in an optimal schedule is no larger than OPT). Thus, $p_{1k} \leq$ OPT, and in the alternative schedule, where this job moves to M_1, the new load of M_1 is at most $L_1 + p_{1k} \leq 2$OPT. However, we know that the given schedule is a NE, which means that $L_1 + p_{1k} \geq L_2$, giving $L_2 \leq 2OPT$. Therefore, $WPO\text{-}PoA \leq 2$. □

From Theorems 13 and 14 we conclude that for $m = 2$, $WPO\text{-}PoA = 2$, and for $m \geq 3$, $WPO\text{-}PoA = \infty$.

We prove next that like the $SPoA$, the $SPO\text{-}PoA$ is m. We should note that the previous results for the $SPoA$ cannot be used here. As we saw, the sets of SNE and SPO-NE have no particular relation. The proofs used for the $SPoA$ do not hold for the $SPO\text{-}PoA$ and need to be adapted. The lower bound of m on the $SPoA$ by Andelman et al. [1] is not strictly Pareto optimal (see below), and in the proof of the upper bound by Fiat et al. [13] the claim is proved by considering alternative schedules where the jobs which change their strategies are proper subsets of jobs (so other jobs may increase their costs).

THEOREM 15. *The $SPO\text{-}PoA$ for m unrelated machines in any job scheduling game is at most m.*

PROOF. Consider a schedule $\mathcal{A}$ on m unrelated machines which is a SPO-NE. Assume that the machines are sorted by non-increasing order of loads, that is, $L_1 \geq L_2 \geq \ldots \geq L_m$. The makespan of $\mathcal{A}$ is therefore L_1.

First, note that $L_m \leq OPT$. If $L_m > OPT$ then an optimal schedule has the property that every job has a smaller cost in it, contradicting the strict Pareto optimality of $\mathcal{A}$. Next, we will prove that $L_i - L_{i+1} \leq OPT$ holds for any $1 \leq i \leq m-1$. Assume by contradiction that there exists i so that $L_i - L_{i+1} > OPT$. We let $L_{i+1} = \delta$. By our assumption $L_i > \delta + OPT$ holds.

Now, consider another schedule $\mathcal{A}'$, where each one of the jobs from machines M_j for $1 \leq j \leq i$ in $\mathcal{A}$ is running on the machine on which it runs in an optimal schedule (all the other jobs hold their positions). We observe that none of these jobs runs on machines $M_{i+1}, \ldots, M_m$ in $\mathcal{A}'$ (or in the optimal schedule under consideration); The processing time of each such job in $\mathcal{A}'$ is at most OPT, and as $L_k \leq \delta$ for $i+1 \leq k \leq m$, its cost in $\mathcal{A}$ if it switches to the machines out of $M_{i+1}, \ldots, M_m$ on which its processing time is at most OPT, then the load of this machine would be at most $\delta + OPT$, while its cost in $\mathcal{A}$ was strictly larger than $\delta + OPT$, contradicting $\mathcal{A}$ being a NE.

We conclude that these jobs are scheduled in $\mathcal{A}'$ on machines $M_1, \ldots, M_i$, where the load of each one of the machines is at most OPT, and that the loads and the allocations on machines $M_{i+1}, \ldots, M_m$ do not change from $\mathcal{A}$ to $\mathcal{A}'$.

This means that in $\mathcal{A}'$ the costs of all jobs from machines $M_1, \ldots, M_i$ in $\mathcal{A}$ are strictly improved, and the costs of all jobs from machines $M_{i+1}, \ldots, M_m$ in $\mathcal{A}$ do not change, which contradicts $\mathcal{A}$ being a SPO-NE. Hence, such i does not exist. Applying this inequality repeatedly, we get that $L_1 \leq L_m + (m-1)$OPT, which in combination with the fact that $L_m \leq OPT$ gives us $SPO\text{-}PoA \leq m$. □

THEOREM 16. *There exists an instance of job scheduling game with m unrelated machines for which $SPO\text{-}PoA \geq m$.*

PROOF. Consider a job scheduling game with m unrelated machines and $n = m$ jobs, where for each job k, $2 \leq k \leq m$: $p_{kk} = k - k\varepsilon$, $p_{k(k-1)} = 1$ and $p_{ik} = \infty$ for all $i \neq k-1, k$. For job 1, $p_{11} = 1 - \varepsilon$ (for some small positive $\varepsilon < \frac{1}{m}$), $p_{m1} = 1$ and $p_{i1} = \infty$ for all $i \neq 1, m$.

In an optimal schedule for this game each one of the jobs $2 \leq k \leq m$ runs alone on machine M_{k-1} and job 1 runs on M_m, which yields a makespan of 1.

On the other hand, a schedule where each one of the jobs $1 \leq k \leq m$ runs alone on machine M_k has a makespan of $m - m\varepsilon$. We will show that this schedule is a SPO-NE. The schedule is a NE, since for each job, moving to the only additional machine on which its processing time is not infinite increases it cost by at least ε. Consider an alternative schedule where no job increases its cost. Job 1 is currently assigned on a machine with load $1 - \varepsilon$, which is the minimal possible cost for it, and this minimum is unique. Thus any alternative schedule must keep job 1 assigned alone to the first machine. We can prove by induction on the indices of jobs that every job has to stay assigned to its current machine alone; once job k must stay on its machine alone, job $k+1$ does not have an alternative machine, and adding another job to the machine that it is assigned to (M_k) would increase its cost. Thus such a schedule does not exist. This gives that $SPO\text{-}PoA \geq m$. □

We conclude that for for any m, $SPO\text{-}PoA = m$.

This is a proper place to mention that the lower bound example from [1] showing that $SPoA \geq m$ looks similar to our example at a first glance. The difference in processing times is in the definition $p_{kk} = k$, for $1 \leq k \leq m$. However, this example does not apply here, as the schedule of cost m which it gives is not strictly Pareto optimal; if we switch to the optimal schedule, where job 1 runs on M_m and each job $2 \leq k \leq m$ runs on M_{k-1}, all jobs $2 \leq k \leq m$ strictly improve their costs and job 1 is not worse off.

This is another example which demonstrates the fact that in the job scheduling game we consider not every SNE is necessarily a SPO-NE. However, we showed that this is the case already for related machines.

3. RECOGNITION OF PARETO OPTIMAL EQUILIBRIA

In this section we consider the computational complexity of SPO-NE and WPO-NE for all machine models. Specifically, we investigate the problem of recognition of such schedules.

THEOREM 17. *There exists a polynomial time algorithms which receives a schedule on related machines (or on identical machines) and check whether the schedule is a* SPO-NE *and whether it is a* WPO-NE.

PROOF. Consider a schedule $\mathcal{A}$, and recall that one can determine in polynomial time whether a given schedule is a NE. Since any NE on identical machines is a WPO-NE and a SPO-NE, the recognition of such schedules is equivalent to recognition of NE. This is also the case for related machines and WPO-NE.

For the recognition of SPO-NE on related machines, we use the following algorithm. First, check whether the schedule is a NE (if not, then output a negative answer). If the schedule is a NE and it does not contain an empty machine, return a positive answer. Otherwise, for every job k, such that k is assigned to a machine which has at least two jobs assigned to it, test if moving it to an empty machine of maximum speed does not increase its cost. If there exists a job for which the cost is not increased, return a negative answer, and otherwise, a positive answer. Note that if there exists

an empty machine, but no machine has two jobs assigned to it, then the returned answer is positive.

Now we prove correctness of the last algorithm. If there are no empty machines then any NE is a SPO-NE (by Theorem 6). For the remaining cases of the algorithm, we prove the following claim.

CLAIM 18. *Given $\mathcal{A}$, which is a* NE*, there exists an alternative schedule $\mathcal{A}'$ where no job increases its cost and at least one job reduces its cost if and only if there exists a job k which is assigned to a machine with at least one other job in $\mathcal{A}$, and moving it to an empty machine of maximum speed does not increase its cost.*

PROOF. We first assume that such a job k exists. Consider the schedule $\tilde{\mathcal{A}}$ in which k is assigned to a machine of maximum speed which is empty in $\mathcal{A}$, and the rest of the assignment is the same as in $\mathcal{A}$. There is at least one job which is assigned to the same machine as k in $\mathcal{A}$, whose cost is strictly reduced (since the load of its machine decreases when k is moved to another machine). The cost of k does not increase, and any job assigned to any machine other than the machine of k in $\mathcal{A}$ and the machine of k in $\tilde{\mathcal{A}}$ keeps its previous cost.

Next, assume that $\mathcal{A}'$ exists, and assume that among such schedules, $\mathcal{A}'$ has a minimum number of jobs which are assigned not to the same machine as in $\mathcal{A}$. Using Theorem 6, we get that $\mathcal{A}$ necessarily has an empty machine $M_{i'}$ which is non-empty in $\mathcal{A}'$. Let k be a job assigned to $M_{i'}$ in $\mathcal{A}'$ and let M_i be the machine to which it is assigned in $\mathcal{A}$.

If machine M_i does not have an additional job in $\mathcal{A}$, and since its cost on $M_{i'}$ (possibly with additional jobs) is no larger, we get $s_{i'} \geq s_i$. However, the schedule is a NE, so k cannot reduce its cost by moving to an empty machine. Therefore, its cost on $M_{i'}$ is the same as its cost on M_i, $s_{i'} = s_i$ and k is assigned to $M_{i'}$ alone in $\mathcal{A}'$. The jobs assigned to M_i in $\mathcal{A}'$ are not assigned to $M_{i'}$ or to M_i in $\mathcal{A}$. This is true since $M_{i'}$ is empty in $\mathcal{A}$ and M_i only has the job k in $\mathcal{A}$. We construct a schedule $\hat{\mathcal{A}}$ where the jobs assigned to M_i and $M_{i'}$ in $\mathcal{A}'$ are swapped and the other jobs are assigned to the same machines as in $\mathcal{A}$. The number of jobs assigned to a different machine from their machines in $\mathcal{A}$ is reduced by 1 (due to k being assigned to the same machines in $\hat{\mathcal{A}}$ and $\mathcal{A}$), which contradicts the choice of $\mathcal{A}'$.

Thus, there exists an additional job k' assigned to M_i in $\mathcal{A}$. Since moving k to some empty machine does not increase its cost, then moving it to an empty machine with maximum speed clearly does not increase its cost. □

Given the claim, if every non-empty machine has a single job then the schedule is a SPO-NE. Otherwise, the algorithm tests the existence of a job k as in the claim.

THEOREM 19. *i. The problem of checking whether a given schedule on unrelated machines is a* WPO-NE *is strongly co-NP-complete. ii. The problem of checking whether a given schedule on unrelated machines is a* SPO-NE *is co-NP-complete.*

PROOF. Given a schedule and an alternative schedule, checking whether the alternative schedule implies that the given schedule is not a NE or not (weakly or strictly) Pareto optimal can be done in polynomial time, and therefore the problems are in co-NP.

To prove hardness of the recognition of WPO-NE, we reduce from the 3-PARTITION problem, which is strongly NP-hard. In this problem we are given an integer B and $3M$ integers $a_1, a_2, \ldots, a_{3M}$, where $\frac{B}{4} < a_k < \frac{B}{2}$ for $1 \leq k \leq 3M$, $\sum_{k=1}^{3M} a_k = MB$, and we are asked whether there exists a partition of the integers into M sets, where the sum of each subset is exactly B. We construct an input with $m = 4M$ machines. There are $4M$ jobs, $3M$ of them are based on the instance of 3-PARTITION and the last M jobs are dummy jobs. For $1 \leq k \leq 3M$, we have $p_{ik} = B + 1$ for $1 \leq i \leq 3M$, and $p_{ik} = a_k$ for $3M + 1 \leq i \leq 4M$. For $3M + 1 \leq k \leq 4M$, we have $p_{ik} = B$ for $1 \leq i \leq 3M$, and $p_{ik} = B + 1$ for $3M + 1 \leq i \leq 4M$. The given schedule is one where job k is assigned to machine k. All machines have a load of $B+1$, so the schedule is a NE. We show that the schedule is weakly Pareto optimal if and only if a 3-partition as required does not exist. Assume first that a 3-partition exists. We define an alternative schedule. In this schedule, each one of the last M machines runs one subset of jobs of the first $3M$ jobs, out of the M subsets of the 3-partition. The sum of the corresponding subsets of numbers in the input of 3-PARTITION is B and therefore, their total processing time on such a machine is B. Each dummy job runs on a different machine out of the first $3M$ machines, having a cost of B. Thus, all jobs have a smaller cost in the alternative schedule, so the original one is not Pareto optimal.

On the other hand, if there exists an alternative schedule where all jobs reduce their costs, then all the first $3M$ jobs must be assigned to the last M machines (since on the other machines even if such a job is assigned to alone it still has a cost of B). For job k, no matter which such machine receives it, it has a processing time of a_k on it, so all jobs have a total processing time of MB. Since all numbers are integers, the only way that every job reduces its load is that each machine will have a load of exactly B, which implies a 3-partition.

To prove hardness of the recognition of SPO-NE, we can use the reduction of [12] showing that the recognition of SNE is hard. For completeness we present an alternative reduction. To prove hardness of the recognition of SPO-NE, we reduce from the PARTITION problem, which is NP-hard. In this problem we are given an integer B and N integers $a_1, a_2, \ldots, a_N$, where, $\sum_{k=1}^{N} a_k = 2B$, and we are asked whether there exists a partition of the integers into two sets, where the sum of each subset is exactly B. We construct an input with $m = 2$ machines (it is possible to use the same input for any larger number of machines, giving all jobs infinite processing times on every machine except for the first two machines). We have $N + 2$ jobs. Job k, for $1 \leq k \leq N$, $p_{1k} = a_k + \frac{1}{2N}$ while $p_{2k} = a_k$. Job $N + 1$ has $p_{1(N+1)} = B$ and $p_{2(N+1)} = B + \frac{1}{2}$. Job $N + 2$ has $p_{1(N+2)} = \infty$ and $p_{2(N+1)} = B$ so it must be assigned to M_2. We are given the schedule where the first N jobs are assigned to M_1 and the two last jobs are assigned to M_2. The loads of both machines are $2B + \frac{1}{2}$, thus this schedule is a NE. If there exists a partition, consider the alternative schedule where each machine receives one subset of jobs whose total size in the original input is B, and job $N + 1$ is assigned to M_1. Let K_1 be the cardinality of the set of jobs assigned to M_1 in the alternative schedule. Then the resulting load of M_1 is $2B + \frac{K_1 - 1}{2N}$. Since M_2 receives at least two jobs, then $K_1 \leq N$, so the load is strictly below $2B + \frac{1}{2}$. The load of M_2 is exactly $2B$. Thus, the original schedule is not strictly (or weakly) Pareto optimal. On the other hand, if the original schedule is not strictly (or weakly) Pareto optimal, then in an alternative schedule, job $N + 1$ must be assigned to M_1, and the total processing time of jobs assigned with it must be strictly below $B + 1$. The total processing time of jobs assigned to M_2 must be strictly below $B + 1$ as well, and so there are two sets whose sizes (in the original input) are at most B, which implies a partition.

Note that this reduction can be used to prove the (weak) co-NP-completeness of the recognition of WPO-NE schedules. Thus both problems are hard for any number of machines. □

4. CONCLUSIONS

In this paper we have studied the quality and complexity of the strict and weak Pareto optimal Nash equilibria in job scheduling games, in the settings of identical, related and unrelated machines.

We found that in the models of identical and related machines, strict and weak Pareto optimal Nash equilibria can be as bad as pure Nash equilibria, however in the model of unrelated machines, while for weak Pareto optimal Nash equilibria and $m \geq 3$ this is still the case, strict Pareto optimal Nash equilibria (and even weak Pareto optimal equilibria, for $m = 2$) are as good as Strong Nash equilibria w.r.t. the Price of Anarchy. This implies that for unrelated machines, cooperation between all players (as opposed to cooperation between subsets of players) still gives solutions of high quality.

As for identical and related machines, recognition of weakly or strictly Pareto optimal equilibria can be done in polynomial time, unlike strong equilibria. Despite the slightly worse quality of such equilibria compared to strong equilibria (due to the results for the Price of Anarchy on related machines), we conclude that weak and strict Pareto optimal equilibria are of interest for identical and related machines.

5. ACKNOWLEDGMENT

The authors would like to thank Asaf Levin for many helpful discussions.

6. REFERENCES

[1] N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. *Games and Economic Behavior*, 65(2):289–317, 2009.

[2] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, É.Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[3] R. J. Aumann. Acceptable points in general cooperative n-person games. In A. W. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV, Annals of Mathematics Study 40*, pages 287–324. Princeton University Press, 1959.

[4] Y. Aumann and Y. Dombb. Pareto efficiency and approximate pareto efficiency in routing and load balancing games. In *Proc. of the 3rd International Symposium on Algorithmic Game Theory (SAGT'10)*, 2010.

[5] B. Awerbuch, Y. Azar, Y. Richter, and D. Tsur. Tradeoffs in worst-case equilibria. *Thoeretical Computer Science*, 361(2-3):200–209, 2006.

[6] S. Chien and A. Sinclair. Strong and pareto price of anarchy in congestion games. In *Proc. of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pages 279–291, 2009.

[7] A. F. T. Committee. ATM forum traffic management specification version 4.0, 1996.

[8] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms*, 3(1), 2007.

[9] P. Dubey. Inefficiency of Nash equilibria. *Mathematics of Operations Research*, 11(1):1–8, 1986.

[10] L. Epstein, M. Feldman, and T. Tamir. Approximate strong equilibria in job scheduling games: an analysis for two uniformly related machines. Manuscript, 2009.

[11] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibrium in load balancing. *ACM Transactions on Algorithms*, 3(3):32, 2007.

[12] M. Feldman and T. Tamir. Approximate strong equilibrium in job scheduling games. *Journal of Artificial Intelligence Research*, 36:387–414, 2009.

[13] A. Fiat, H. Kaplan, M. Levy, and S. Olonetsky. Strong price of anarchy for machine load balancing. In *Proc. of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07)*, pages 583–594, 2007.

[14] G. Finn and E. Horowitz. A linear time approximation algorithm for multiprocessor scheduling. *BIT Numerical Mathematics*, 19(3):312–320, 1979.

[15] D. Fotakis, S. C. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. *Theoretical Computer Science*, 410(36):3305–3326, 2009.

[16] T. Harks, M. Klimm, and R. H. Möhring. Strong Nash equilibria in games with the lexicographical improvement property. In *Proc. of the 5th International Workshop on Internet and Network Economics (WINE'09)*, pages 463–470, 2009.

[17] R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21(1-2):85–101, 1997.

[18] R. M. Karp, E. Koutsoupias, C. H. Papadimitriou, and S. Shenker. Optimization problems in congestion control. In *Proc. of 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00)*, pages 66–74, 2000.

[19] E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis. Approximate equilibria and ball fusion. *Theory of Computing Systems*, 36(6):683–693, 2003.

[20] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proc. of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, pages 404–413, 1999.

[21] D. T. Luc. Pareto optimality. In A. Chinchuluun, P. M. Pardalos, A. Migdalas, and L. Pitsoulis, editors, *Pareto optimality, game theory and equilibria*, pages 481–515. Springer, 2008.

[22] M. Mavronicolas and P. G. Spirakis. The price of selfish routing. *Algorithmica*, 48(1):91–126, 2007.

[23] I. Milchtaich. Network topology and the efficiency of equilibrium. *Games and Economic Behavior*, 57(2):321–346, 2006.

[24] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[25] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[26] P. Schuurman and T. Vredeveld. Performance guarantees of local search for multiprocessor scheduling. *INFORMS Journal on Computing*, 19(1):52–63, 2007.

[27] A. Skopalik and M. Hoefer. On the complexity of pareto-optimal Nash and strong equilibria. In *Proc. of the 3rd International Symposium on Algorithmic Game Theory (SAGT'10)*, pages 312–322, 2010.

[28] W. E. Walsh and M. P. Wellman. A market protocol for decentralized task allocation. In *Proc. of the 3rd International Conference on Multiagent Systems (ICMAS1998)*, pages 325–332, 1998.

Game Theory-Based Opponent Modeling in Large Imperfect-Information Games*

Sam Ganzfried and Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
{sganzfri, sandholm}@cs.cmu.edu

ABSTRACT

We develop an algorithm for opponent modeling in large extensive-form games of imperfect information. It works by observing the opponent's action frequencies and building an opponent model by combining information from a precomputed equilibrium strategy with the observations. It then computes and plays a best response to this opponent model; the opponent model and best response are both updated continually in real time. The approach combines game-theoretic reasoning and pure opponent modeling, yielding a hybrid that can effectively exploit opponents after only a small number of interactions. Unlike prior opponent modeling approaches, ours is fundamentally game theoretic and takes advantage of recent algorithms for automated abstraction and equilibrium computation rather than relying on domain-specific prior distributions, historical data, or a hand-crafted set of features. Experiments show that our algorithm leads to significantly higher win rates (than an approximate-equilibrium strategy) against several opponents in limit Texas Hold'em — the most studied imperfect-information game in computer science — including competitors from recent AAAI computer poker competitions.

Categories and Subject Descriptors

I.2.m [**Computing Methodologies**]: Artificial Intelligence

General Terms

Algorithms, Economics

Keywords

Game theory, multiagent learning

1. INTRODUCTION

While much work has been done in recent years on abstracting and computing equilibria in large extensive-form games, relatively little work has been done on exploiting suboptimal opponents (aka *opponent modeling*). While playing an equilibrium guarantees at least the value of the game in a two-player zero-sum game, often much higher payoffs can be obtained by deviating from equilibrium to exploit opponents who make significant mistakes. For example, against a poker opponent who always folds, the strategy of always raising will perform far better than any equilibrium strategy (which will sometimes fold with bad hands).

Texas Hold'em poker has emerged as the main testbed for evaluating algorithms in extensive-form games. In addition to its tremendous popularity, it also contains enormous strategy spaces, imperfect information, and stochastic events; such elements also characterize most of the challenging problems in computational game theory and multiagent systems. In light of these factors and the AAAI annual computer poker competition, poker has emerged as an important, visible challenge problem for AI as a whole, and multiagent systems in particular.

It is worth noting, however, that a fair amount of prior work has been done on opponent exploitation in significantly smaller games. For example, Hoehn et al. [7] run experiments on Kuhn poker, a small two-player poker variant with about 20 states in its game tree. Recent work has also been done on opponent exploitation in rock-paper-scissors [12] and the repeated prisoners' dilemma [2]. However, these algorithms do not scale to large games. In contrast, the game tree of limit Texas hold'em has about 10^{18} states.

A potential drawback of evaluating algorithms on one specific problem is that we run the risk of developing algorithms that are so game specific that they will not generalize to other settings. Heeding this risk, in this work we abandon many of the game-specific assumptions taken by prior approaches. Rather than relying on massive databases of human poker play [3, 14] and expert-generated features or prior distributions [7, 16], we will instead rely on game-theoretic concepts such as Nash equilibrium and best response, which apply to all games.

In addition, we require our algorithms to operate efficiently in real time (online), as opposed to algorithms that perform offline computations assuming they have access to a large number of samples of the opponent's strategy in advance [9, 13]. That prior work also assumed access to historical data which included the private information of the opponents (i.e., their hole cards) even when such information was only observed by the opponent. In many multiagent settings, an agent must play against opponents about whom he has little to no information in advance, and must learn to

*This material is based upon work supported by the National Science Foundation under IIS grants 0905390 and 0964579. We also acknowledge Intel Corporation and IBM for their machine gifts.

Cite as: Game Theory-Based Opponent Modeling in Large Imperfect-Information Games, Sam Ganzfried and Tuomas Sandholm, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 533-540.

exploit weaknesses in a small number of interactions. Thus, we assume we have no prior information on our opponent's strategy in advance, and our algorithms will operate online.

Our main algorithm, called *Deviation-Based Best Response (DBBR)*, works by noting deviations between the opponent's strategy and that of a precomputed approximate equilibrium strategy, and constructing a model of the opponent based on these deviations. Then it computes and plays a best response to this opponent model (in real time). Both the construction of the opponent model and the computation of a best response take time linear in the size of the game tree and can be performed quickly in practice. As discussed above, we evaluate our algorithm empirically on limit Texas Hold'em; it achieves significantly higher win rates against several opponents — including competitors from recent AAAI computer poker competitions — than an approximate equilibrium strategy does.

2. GAME THEORY BACKGROUND

In this section, we review relevant definitions and results from game theory.

2.1 Extensive-form games

An *extensive-form* game is a general model of multiagent sequential decision-making with imperfect information[1]. As with perfect-information games, extensive-form games consist primarily of a game tree; each non-terminal node has an associated player (possibly *chance*) that makes the decision at that node, and each terminal node has associated utilities for the players. Additionally, game states are partitioned into *information sets*, $I_i \in \mathcal{I}_i$, where a player cannot distinguish among the states in the same information set. Therefore, the player whose turn it is to move must choose actions with the same distribution at each state in the information set.

In this paper, we will only concern ourselves with two-player, zero-sum[2], extensive-form games (though our algorithm extends naturally to multiplayer and non-zero-sum games as well). Furthermore, we will make the standard assumption of *perfect recall*: no player forgets information that he previously knew.

A *history*, $h \in H$, is a sequence of actions. A (mixed) *strategy* for player i, σ_i, is a function that assigns a probability distribution over all actions at each information set belonging to i; by convention the opponent's strategy is denoted σ_{-i}. Let Σ_i denote the (mixed) strategy space of player i. A *strategy profile* σ is a vector of strategies, one for each player.

In this paper we will assume that the moves of all players other than chance are observed by all players; for example, in poker all moves other than the initial dealing of the cards are publicly observed. In this setting, we can partition all game states into *public history sets*, PH_i, where states in the same public history set correspond to the same history of publicly observed actions. Note that each public history set must consist of a set of information sets of player i. For public history set $n \in PH_i$, let A_n denote the set of actions of player i at n. In general when we omit subscripts, player i will be implied.

[1]Much of our description of extensive-form games is adapted from [11].

[2]An extensive-form game is zero-sum if the sum of the payoffs at each terminal node equals zero.

2.2 Best responses and Nash equilibria

Player i's *best response* to σ_{-i} is any strategy in

$$\arg\max_{\sigma'_i \in \Sigma_i} u_i(\sigma'_i, \sigma_{-i}).$$

A *Nash equilibrium* is a strategy profile σ such that σ_i is a best response to σ_{-i} for all i. An *ϵ-equilibrium* is a strategy profile in which each player achieves a payoff of within ϵ of his best response. Formally, an ϵ-equilibrium is a strategy profile σ^* such that, for all i, we have

$$u_i(\sigma^*_i, \sigma^*_{-i}) \geq \max_{\sigma_i \in \Sigma_i} u_i(\sigma_i, \sigma^*_{-i}) - \epsilon.$$

All finite games have at least one Nash equilibrium. In the case of zero-sum extensive-form games with perfect recall, there are efficient techniques for finding an ϵ-equilibrium, such as linear programming (LP) [10], the excessive gap technique (EGT) [6], and counterfactual regret minimization (CFR) [17]. However, the latter two scale to much larger games; they scale to 10^{12} states in the game tree, while the best current LP techniques do not scale beyond 10^8 states.

Best responses can be computed much more efficiently than Nash equilibria. Computing a best response involves a single matrix-vector multiplication followed by a traversal up the game tree, both of which take linear time in the size of the game tree.

2.3 Abstraction

Despite the tremendous progress in equilibrium-finding in recent years, many interesting real-world games (such as poker) are so large that even the best algorithms have no hope of computing an equilibrium directly. The standard approach of dealing with this is to apply an *abstraction* algorithm, which constructs a smaller game that is similar to the original game; then the smaller game is solved, and its solution is mapped to a strategy profile in the original game. The approach has been applied to two-player Texas Hold'em poker, first with a manually generated abstraction [1], and currently with abstraction algorithms [4]. Many abstraction algorithms work by coarsening the moves of chance, collapsing several information sets of the original game into single information sets of the abstracted game. We will sometimes refer to information sets in abstracted games as *buckets*.

The game tree of limit Texas hold'em has about 10^{18} states, and recent solution techniques can compute approximate equilibria for abstractions with up to 10^{12} states [5, 17]. Such algorithms typically take several weeks to compute an ϵ-equilibrium for reasonably small ϵ. On the other hand, best responses in such an abstraction can be computed in about an hour. If coarser abstractions are used, best responses can be computed in minutes or even seconds, and can potentially be used as a subroutine in adaptive real-time algorithms.

3. IMPOSSIBILITY OF SAFE EXPLOITATION

While deviating from equilibrium to exploit an opponent can often lead to a significantly higher payoff, it also runs the risk that the exploitative strategy can itself become exploitable. For example, the opponent could play a certain

strategy for several iterations to trick the exploiter, then exploit him in turn; this is referred to as the *get-taught-and-exploited problem* [15].

One might think that this problem can be avoided by only risking the amount won so far. For example, suppose we are repeating a two-player zero-sum game (with value zero) 100 times, and have won $50 so far through 50 iterations. Then if we attempt to exploit the opponent for the next 50 iterations by playing a strategy with exploitability at most $1 per iteration, it appears that we may be able to safely exploit the opponent by deviating from equilibrium while still guaranteeing the value of the game. Unfortunately, this intuition is not correct; it is possible that the opponent was in fact playing an equilibrium all along and that we were just lucky for the first 50 iterations. If we then deviate from equilibrium, our overall strategy could actually have a negative payoff in expectation against an equilibrium opponent. Formally:

PROPOSITION 1. *It is not possible to exploit an opponent by deviating from equilibrium while simultaneously guaranteeing obtaining the value of the game in expectation.*

Thus, we must turn to algorithms that are exploitable to some extent in the worst case if we hope to exploit the opponent more than any equilibrium strategy does.

4. DBBR: AN EFFICIENT REAL-TIME OPPONENT MODELING ALGORITHM

In this section we present our algorithm, *Deviation-Based Best Response (DBBR)*. It works by observing the opponent's action frequencies over the course of game, then using these observations to construct a model of the opponent's strategy. Essentially, we would like to conservatively assume that the opponent is playing the best (i.e., least exploitable) strategy that is consistent with our observations of his play. The obvious way to accomplish this would be to add linear constraints to the LP for finding an equilibrium [10] that force the opponent model to conform with our observations. However, as discussed in Section 2.3, such a computation could take several weeks, and would not be practical for real-time play in large games.

To obtain a more practical algorithm, we must find a faster way of constructing an opponent model from our observations. DBBR constructs the model by noting deviations of our opponent's observed action frequencies from equilibrium frequencies. For example, in poker suppose an equilibrium strategy raises 50% of the time when first to act, while the opponent raises only 30% of the time. While the opponent might be raising any 30% of hands, a safe guess might be to assume that he is raising his 'best' 30% of hands; we can construct such a strategy by starting with the equilibrium strategy, then removing the 'worst' 20% of hands from the raising range. Our algorithm is based on this intuition.

4.1 Overview of the algorithm

Pseudocode for a high-level overview of DBBR is given in Algorithm 1. In the first step, an approximate equilibrium σ^* of the game is precomputed offline. Next, when the game begins, the frequencies of the opponent's actions at different public history sets are recorded. These are used to compute the opponent's *posterior action probabilities*: the probabilities with which he chooses each action at each public history set $n \in PH_{-i}$. (We say that the elements of PH_{-i} are numbered according to breadth-first-search (BFS) traversal order.) Next, we compute the probability the opponent is in each bucket at n given our model of his play so far; we refer to these probabilities as the *posterior bucket probabilities*. We then compute a full model of the opponent's strategy by considering the deviations between the opponent's posterior action probabilities and those of σ^* at n. Based on these deviations, we iterate over all buckets and shift weight away from the action probabilities in σ^* until we obtain a strategy consistent with our model of the opponent's action probabilities. Finally, after we have iterated over all public history sets, we compute a best response to the opponent model. The next subsections will discuss the different components of the algorithm in detail.

Algorithm 1 High-level overview of *DBBR*

```
Compute an approximate equilibrium of the game.
Maintain counters from observing opponent's play throughout the match.
for n = 1 to |PH_{-i}| do
    Compute posterior action probabilities at n.
    Compute posterior bucket probabilities at n.
    Compute full model of opponent's strategy at n.
end for
return Best response to the opponent model.
```

4.2 Computing posterior action probabilities

In the course of our play against the opponent, we observe how often he chooses each action a at each public history set n; we denote this quantity by $c_{n,a}$. One idea would be to assume the opponent will play action a with probability

$$\frac{c_{n,a}}{\sum_{a'} c_{n,a'}}.$$

However, doing this could be problematic for a few reasons. First, we might not have any observations at a given set n, in which case this quantity would not even be defined. More generally, the quality of our observations might vary dramatically between public history sets; for example, we have a lot more confidence in sets for which we have 1000 observations than sets for which we have just 1 or 2, and we would like our algorithm to reflect this. A similar observation was the motivation behind a recent paper [8], though that work assumed that the opponent's private information was observable.

Our algorithm works by choosing a combination of the observed probability and the probability under the equilibrium strategy σ^*, where the weight on the observed frequencies is higher at public history sets for which we have more observations. Specifically, we use a Dirichlet prior distribution, where we assume we have seen N_{prior} fictitious hands at the given public history set for which the opponent played according to σ^*. Let $p^*_{n,a}$ denote the probability that σ^* plays action a at public history set n. We compute the posterior action probabilities, $\alpha_{n,a}$, as follows:

$$\alpha_{n,a} = \frac{p^*_{n,a} \cdot N_{prior} + c_{n,a}}{N_{prior} + \sum_{a'} c_{n,a'}}. \quad (1)$$

4.3 Computing posterior bucket probabilities

Since we are constructing the model of the opponent's strategy using a BFS ordering of the public history sets,

we assume that we have already set his strategy for all ancestors of the current set n (including the parent n'). Let $s_{n',b,a}$ denote our model of the probability that the opponent plays his portion of the strategy sequence leading to n', then chooses action a in bucket b at state n'; this quantity has already been computed by the time we get to n in the algorithm. We can use these probabilities to construct the posterior probability, $\beta_{n,b}$, that the opponent is in bucket b (i.e., in poker, the opponent has those private cards) at public history set n. Pseudocode for this procedure is given in Algorithm 2, where h_b denotes the probability that chance makes the moves needed to put the opponent in bucket b.

Algorithm 2 ComputeBucketProbs(n)

for $b = 1$ **to** $|B_n|$ **do**
  $n' \leftarrow parent(n)$
  $a \leftarrow$ action taken to get from n' to n.
  $\beta_{n,b} \leftarrow h_b \cdot s_{n',b,a}$
end for
Normalize the values β_n so they sum up to 1.

4.4 Computing the opponent model

In this section we will present three different techniques for computing the opponent model. Recall that our high-level goal is to compute the 'best' (i.e., least exploitable) strategy for the opponent that is consistent with our observations of his behavior. We could accomplish this by performing an equilibrium-like computation; however, such a computation is too challenging to be performed in real time.

Rather than find the strategy consistent with our observations that is least exploitable, we will instead find the strategy that is 'closest' to the precomputed equilibrium. It turns out that this can be accomplished efficiently in practice, and intuitively we would expect strategies closer to equilibrium to be less exploitable.

4.4.1 Weighted L_1-distance minimization

Recall that the L_1 distance between two vectors x and y is defined as

$$||x - y||_1 = \sum_{i=1}^{k} |x_i - y_i|. \tag{2}$$

While this function treats all indices of the vector equally, in some cases we might want to put more weight on some components than on others. If p is a probability distribution over the integers from 1 to k, we define the *weighted* L_1 *distance* between x and y as

$$\sum_{i=1}^{k} p_i \cdot |x_i - y_i|. \tag{3}$$

Now, suppose we are at public history set n, where $\beta_{n,b}$ denotes the posterior probability that we are in bucket b, as computed by Algorithm 2. If we let the y_i's in Equation 3 correspond to the equilibrium probabilities of taking each action, and let the p_i's correspond to the $\beta_{n,b}$'s, then we can formulate the problem of finding the strategy closest to the precomputed equilibrium, subject to the posterior action probabilities $\alpha_{n,a}$, as an L_1-distance minimization problem.

Formally, we can formulate the optimization problem as follows, for a given public history set n:

$$\begin{aligned}
\text{minimize} \quad & \sum_{b \in B_n} \sum_{a \in A_n} \left[\beta_{n,b} \cdot |x_{n,b,a} - \sigma^*_{n,b,a}|\right] && (4)\\
\text{subject to} \quad & \sum_{b \in B_n} [\beta_{n,b} \cdot x_{n,b,a}] = \alpha_{n,a} \text{ for all } a \in A_n \\
& \sum_{a \in A_n} x_{n,b,a} = 1 \text{ for all } b \in B_n \\
& 0 \le x_{n,b,a} \le 1 \text{ for all } a \in A_n, b \in B_n
\end{aligned}$$

Recall that B_n denotes the set of all buckets we could be in at public history set n, while A_n denotes the set of actions at n. The variables $x_{n,b,a}$ correspond to the model of the opponent's strategy that we are trying to compute. Note that we can do this optimization separately for each public history set n; it makes more sense to do many smaller optimizations than to do a huge one for all public history sets at once, since the computations of the actions taken at different states do not depend on each other.

So as discussed above, we will perform a separate optimization at each n according to the program of Equation 4. It turns out that this can be cast as a linear program (LP) and solved efficiently using CPLEX's dual simplex algorithm for solving LPs. Doing this for each public history set n yields the opponent model x. Note that the program could have many solutions, and that CPLEX will just output the first solution it encounters (and not necessarily the solution that performs best in practice). This means that there might actually exist a strategy that minimizes L_1 distance from equilibrium that performs better in practice than the strategy output by CPLEX.

4.4.2 Weighted L_2-distance minimization

While Section 4.4.1 uses the weighted L_1 distance to measure the proximity of two strategies, we could also use other distance metrics. In this section we will consider another common distance function: the weighted L_2 distance.

Similarly to Equation 2, the L_2 distance between x and y is defined as

$$||x - y||_2 = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}. \tag{5}$$

Analogously to the L_1 case, we define the *weighted* L_2 *distance* between x and y as

$$\sqrt{\sum_{i=1}^{k} p_i \cdot (x_i - y_i)^2}. \tag{6}$$

The new program for computing the opponent model at n is the following:

$$\begin{aligned}
\text{minimize} \quad & \sum_{b \in B_n} \sum_{a \in A_n} \left[\beta_{n,b} \cdot (x_{n,b,a} - \sigma^*_{n,b,a})^2\right] && (7)\\
\text{subject to} \quad & \sum_{b \in B_n} [\beta_{n,b} \cdot x_{n,b,a}] = \alpha_{n,a} \text{ for all } a \in A_n \\
& \sum_{a \in A_n} x_{n,b,a} = 1 \text{ for all } b \in B_n \\
& 0 \le x_{n,b,a} \le 1 \text{ for all } a \in A_n, b \in B_n
\end{aligned}$$

Note that we can omit the square root, since it is a monotonic operator. The resulting formulation in Equation 7 is a

quadratic program (QP), which can also be solved efficiently in practice using CPLEX. As in the L_1 case, we can formulate and solve a separate optimization problem for each public history set n to compute the opponent model x.

4.4.3 Our custom weight-shifting algorithm

While the previous two sections described how to compute an opponent model using two popular distance functions, perhaps we can do even better by designing our own custom algorithm that takes into account the conservative reasoning about the opponent that we discussed earlier. In this section we will describe such an algorithm. In particular, it takes into account the fact that we already know an approximate ranking of the buckets at each public history set from the approximate equilibrium σ^*.

For example, suppose the opponent is only raising 30% of the time when first to act, while σ^* raises 50% of the time in that situation (as given in the example at the beginning of this section). Instead of doing a full L_1 or L_2-minimization explicitly, we could use the following heuristic algorithm: sort all buckets by how often the opponent raises with them under σ^*, then greedily keep removing buckets from his raising range until the weighted sum (using the $\beta_{n,b}$'s as weights) equals 30%. This is a simple greedy algorithm, which can be run significantly more efficiently in practice than the L_1 and L_2-minimization procedures described in the last two subsections, which must repeatedly use CPLEX at runtime.

For simplicity, we present our algorithm for the case of three actions, although it extends naturally to any number of actions. First we initialize the opponent's strategy at n, σ_n, to the equilibrium σ^*. We also initialize our current model of his action probabilities γ_n to $p^*_{n,a}$, the equilibrium action probabilities.

Next, we check whether the opponent is taking action 3 more often than he should at n by comparing $\alpha_{n,3}$ to $\gamma_{n,3}$. If he is, we are going to want to increase the probabilities he plays action 3 in various buckets; otherwise, we will decrease these probabilities. For now, we will assume that $\alpha_{n,3} > \gamma_{n,3}$ (the other case is handled analogously).

We start by adding weight to the bucket that plays action 3 with the highest probability at n; denote this bucket by $\hat{b}$. If

$$\gamma_{n,3} + \beta_{n,\hat{b}} \cdot (1 - \sigma_{n,\hat{b},3}) < \alpha_{n,3}, \tag{8}$$

we set $\sigma_{n,\hat{b},3} = 1$, since that will not cause $\gamma_{n,3}$ to exceed $\alpha_{n,3}$ once it is adjusted. Otherwise, we increase $\sigma_{n,\hat{b},3}$ by $\frac{(\alpha_{n,3}-\gamma_{n,3})}{\beta_{n,\hat{b}}}$. (Recall that $\beta_{n,\hat{b}}$ denotes the posterior probability that the opponent holds bucket $\hat{b}$ at n, as computed in Algorithm 2.) Let Δ denote the amount by which we increase $\sigma_{n,\hat{b},3}$. We will also increase the action probability $\gamma_{n,3}$ by $\beta_{\hat{b}} \cdot \Delta$.

Next we must compensate for this increase of the probability of playing action 3 in bucket $\hat{b}$ by decreasing the probabilities of playing actions 1 and/or 2. Let $\underline{a}$ denote the action (1 or 2) played with lower probability in σ_n in bucket $\hat{b}$, and let $\overline{a}$ denote the other action. If $\sigma_{n,\hat{b},\underline{a}} \geq \Delta$, then we set $\sigma_{n,\hat{b},\underline{a}} = \sigma_{n,\hat{b},\underline{a}} - \Delta$ and update $\gamma_{n,\underline{a}}$ accordingly. Otherwise, we set $\sigma_{n,\hat{b},\underline{a}} = 0$ and remove the remaining probability $\Delta - \sigma_{n,\hat{b},\underline{a}}$ from $\sigma_{n,\hat{b},\overline{a}}$.

If the inequality of Equation 8 held above, then our opponent model probabilities still do not agree with the posterior action probabilities, and thus we must continue shifting probability mass; we continue by setting $\hat{b}$ to the bucket that plays action 3 with the second highest probability at n, and repeating the above procedure. Otherwise, we are done setting the probabilities for action 3, and we perform a similar procedure to shift weight between the probabilities that he plays actions 1 and 2 until they agree with α_n.

We have now constructed an opponent model that agrees with our posterior action probabilities. Note that we had to iterate over possibly all of the buckets at public history set n. Since each bucket is contained in only one public history set, the algorithm's run time is linear in the size of the game tree.

Additionally, although we presented this algorithm for the case of three actions at n, it easily generalizes to more actions. Rather than just designating $\overline{a}$ and $\underline{a}$, we will sort all actions in the order of how often they are played in bucket $\hat{b}$, and proceed through this list adjusting probabilities as in the three-action case.

4.5 Full algorithm

In practice, constructing an opponent model and computing a best response at each repetition of the game (e.g., hand in poker) might be too slow. This can be mitigated by doing so only every k repetitions. In addition, we may want to start off playing the equilibrium σ^* for several repetitions so that we can obtain a reasonable number of samples of the opponent's play, rather than trying to exploit him immediately. Overall, our full algorithm will have three parameters: T denotes how many repetitions to first play the equilibrium σ^* before starting to exploit, k denotes how often to recompute an opponent model and best response, and N_{prior} from Equation 1 is the parameter of the action probability prior distributions. Pseudocode for the algorithm is given in Algorithm 3, where M is the number of repetitions in the match.

Algorithm 3 DBBR(T,k,N_{prior})

```
for iter = 1 to T do
    Play according to the precomputed equilibrium strategy σ*
end for
opponent_model = ComputeOppModel(N_prior)
σ_BR = ComputeBestResponse(opponent_model)
for iter = T + 1 to M do
    if iter is a multiple of k then
        opponent_model = ComputeOppModel(N_prior)
        σ_BR = ComputeBestResponse(opponent_model)
    end if
    Play according to σ_BR
end for
```

5. EXPERIMENTS AND DISCUSSION

We used two-player Limit Texas Hold'em as our experimental domain. It is a large-scale game with 10^{18} states in the game tree. It is the most-studied full-scale poker game in computer science, and is also played by human professionals.

5.1 Limit Texas Hold'em

The rules of the game are as follows. Each player at the table is dealt two private *hole cards*, and the players initially have 1 and 2 chips invested in the pot respectively. Then

there is a round of betting, after which three cards (called the *flop*) are dealt face up in the middle of the table. Then there is another round of betting, followed by another card dealt face up (the *turn*); then one more round of betting, followed by a fifth card face up (the *river*), followed by a final round of betting.

During each betting round, each player has three possible options. (1) *fold*: pass and forfeit his chance of winning the pot. (2) *call*: put a number of chips equal to the size of the current bet into the pot. (3) *raise*: put a fixed number of additional chips in the pot beyond what was needed to call.

If one player folds during the course of betting, then the other player wins the entire pot. If neither player has folded, the player with the best five-card hand (constructed from his two hole cards and the five community cards) wins the pot. In case of a tie, the players split the pot evenly.

As in the AAAI computer poker competitions, in our experiments, each *match* consists of 3000 *duplicate hands*: 3000 hands are played normally, then the players switch positions and play the same 3000 hands (with no memory of the previous hands). This is a well-known technique for reducing the variance so that fewer hands are needed to obtain statistical significance. Whenever we match two players, we have them play several duplicate matches and report the standard error.

5.2 Experimental results

We ran our algorithm against several opponents; the results are shown in Table 1. The first four opponents — Random, AlwaysFold, AlwaysCall, and AlwaysRaise — play naïvely as their names suggest. GUS2 and Dr. Sahbak were entrants in the 2008 AAAI computer poker competition, and Tommybot was an entrant in the 2009 competition; we selected these bots to experiment against because they had the worst performances in the competitions, and we expect opponent modeling to provide the biggest improvement against weak opponents. Against stronger opponents one might prefer to always play the precomputed equilibrium rather than turning on the exploitation. This can be accomplished by periodically looking at the win rate, and only attempting to exploit the opponent if a win rate above some threshold is attained.

GS5 is a bot we entered in the 2009 AAAI computer poker competition that plays an approximate-equilibrium strategy. It was computed using an abstraction which had branching factors of 15, 40, 6, and 6 respectively in the four betting rounds. The parameter values we used in DBBR (as described in Section 4.5) were $T = 1000$, $k = 50$, $N_{prior} = 5$, with GS5 playing the role of the initial approximate-equilibrium strategy (i.e., we ran GS5 for the first 1000 hands of each match and recomputed an opponent model and best response every 50 hands subsequently). Since each match consists of 3000 duplicate hands, this means that GS5 and DBBR play the same strategy for the first third of each match.

We set $T = 1000$ since it is essential that our algorithm obtains a reasonable number of samples of the opponent's play (in different parts of the game tree) before attempting to exploit. As discussed in the next paragraph, our main motivation in setting k was to allow us to update the opponent model as frequently as we could while remaining under the competition time limit. For N_{prior}, we wanted to choose a small number so that our observations would quickly trump the prior for common public history sets, but so that the prior would have more weight if we had just one or two observations. Note that setting $N_{prior} = 5$ means that our prior and our observations will have equal weight in our model when we have observed the opponent's action 5 times at the given public history set. Changing the parameter values could certainly have a large effect on the results, and should be studied further.

Unfortunately GS5 was too large to use as the approximate-equilibrium strategy in our real-time opponent modeling updates. Therefore, we also precomputed an approximate-equilibrium σ^* that used a much smaller abstraction than GS5: the branching factors of its abstraction were 8, 12, 4, and 4. While σ^* is clearly an inferior strategy to GS5, it was small enough to allow us to construct opponent models and compute best responses in just a few seconds, keeping us within the time limit of the AAAI competition.

We experimented with all three of the approaches for computing the opponent model described in Section 4.4: the three algorithms DBBR-L_1, DBBR-L_2, and DBBR-WS (i.e., 'Weight-Shifting') correspond to the three different algorithms in that section. We ran all three of these algorithms against each of the opponents described above (with the exception of Tommybot, which we were not able to play against DBBR-L_1 and DBBR-L_2 due to technical issues).

As shown in Table 1, our main algorithm DBBR-WS performed significantly better against all of the opponents than GS5 did (in one case, the win rate was over twice as high). Furthermore, DBBR-WS beat GUS2 by more than any other bot in the 2008 competition did, and its win rates against Dr. Sahbak and Tommybot were surpassed by the win rate of just a single bot.

5.3 Comparing the opponent modeling algorithms

It is not totally clear from the results in Figure 1 which of the three algorithms for constructing the opponent model — L_1, L_2, or our weight-shifting algorithm — is best. For example, DBBR-WS obtains a win rate of 1.391 sb/h against AlwaysRaise while DBBR-L_1 obtains a win rate of 0.878 sb/h, but DBBR-L_1 obtains a win rate of 2.164 sb/h against Random while DBBR-WS obtains only 1.769 sb/h. Similarly, for all other pairings there exist opponents such that one bot achieves a higher win rate against one opponent, but not against the other opponent. So there is no clear total ordering of the three algorithms.

That being said, DBBR-L_2 does at least as well (or essentially the same) against all of the opponents as DBBR-L_1, except for Dr. Sahbak; this suggests that DBBR-L_2 is a stronger program. As between DBBR-L_2 and DBBR-WS, it really seems to depend on the opponent. DBBR-WS performs significantly better against AlwaysRaise, GUS2, and Dr. Sahbak and slightly better against AlwaysFold than DBBR-L_2; however, DBBR-L_2 performs significantly better against Random and slightly better against AlwaysCall than DBBR-WS. So DBBR-WS performs significantly better against three of the six opponents than DBBR-L_2 (and essentially the same against two opponents), suggesting that it is a better algorithm.

In addition, DBBR-WS performs significantly better against both of the actual opponents from the AAAI competition (GUS2 and Dr. Sahbak) than DBBR-L_2, which suggests that it might perform better in practice against realistic op-

	Random	AlwaysFold	AlwaysCall	AlwaysRaise	GUS2	Dr. Sahbak	Tommybot
GS5	0.854 ± 0.008	0.646 ± 0.0009	0.582 ± 0.005	0.791 ± 0.009	0.636 ± 0.004	0.665 ± 0.027	0.552 ± 0.008
DBBR-WS	1.769 ± 0.025	0.719 ± 0.002	0.930 ± 0.014	1.391 ± 0.034	0.807 ± 0.011	1.156 ± 0.043	1.054 ± 0.044
DBBR-L_1	2.164 ± 0.036	0.717 ± 0.002	0.935 ± 0.017	0.878 ± 0.032	0.609 ± 0.054	1.153 ± 0.074	
DBBR-L_2	2.287 ± 0.046	0.716 ± 0.002	0.931 ± 0.026	1.143 ± 0.084	0.721 ± 0.050	1.027 ± 0.072	

Table 1: Win rate in small bets/hand of the bot listed in the row. The ± given is the standard error (standard deviation divided by the square root of the number of hands).

ponents. This fact, combined with the fact that DBBR-WS is more efficient than the other algorithms, which have to perform many optimizations using CPLEX at runtime, suggest that DBBR-WS is a better algorithm to use in practice.

Note that this does not imply that the weighted L_1 and L_2 distance functions are poor distance metrics; it just means that the particular solution output by CPLEX does not do as well as the solution output by DBBR-WS. It is very possible that if CPLEX used different LP/QP algorithms, it might find a solution that does significantly better. This would certainly be a worthwhile avenue for future work.

5.4 Win rates over time

One might expect that DBBR[3] would immediately begin exploiting the opponents at hand 1001 — when it switches from playing an approximate equilibrium to opponent modeling — and that the win rate would increase steadily. In fact, this happened in the matches against most of the bots. For example, Figure 1(a) shows that DBBR's profits against AlwaysFold increase linearly over time, and Figure 1(d) shows that DBBR's win rate increases in a concave fashion.

Surprisingly, we observed a different behavior in the matches against AlwaysRaise and GUS2. In both of these matches, the win rate decreases significantly for the first several hundred hands before it starts to increase, as shown in Figure 1. This happens because the approximate-equilibrium strategy plays some action sequences with very low probability, leading it to not explore the opponent's full strategy space in the 1000 hands. This will lead to a significant disparity between the prior and actual strategies of the opponent at hand 1001 if the opponent's strategy differs significantly from the approximate equilibrium in those unexplored regions. This in turn may cause DBBR to think it can immediately exploit the opponent in certain ways, which turn out to be unsuccessful; but eventually as DBBR explores these sequences further and gathers more observations, it figures out successful exploitations.

The following hand from our experiments between DBBR and AlwaysRaise exemplifies this phenomenon. The hand was the 1006'th hand of the match. There were many raises and re-raises during the preflop, flop, and turn betting rounds. When the river card came, DBBR had only a ten high (a very weak hand in this situation). However, based on its observations during the first 1005 hands, it knew that AlwaysRaise had a very wide range of hands given this betting sequence, many of which were also weak hands (though probably still stronger than ten high). On the other hand, DBBR had very few observations of how AlwaysRaise responds to a series of raises on the river, since GS5 made those plays very rarely during the first 1000 hands; hence, DBBR resorted to the prior to model the opponent, which had the opponent folding all of his weak hands to a raise (since GS5 would do this). So DBBR thought that raising would get the opponent to fold most of his hands, while in reality AlwaysRaise continues to raise with all of his hands. In this particular hand, DBBR lost a significant amount of money due to the additional raises he made on the river with a very weak hand.

[3]The results in this section refer to our main algorithm, DBBR-WS.

6. CONCLUSION

We presented DBBR, an efficient real-time algorithm for opponent modeling and exploitation in large extensive-form games. It works by observing the opponent's action frequencies and building an opponent model by combining information from a precomputed equilibrium strategy with the observations. This enables the algorithm to combine game-theoretic reasoning and pure opponent modeling, yielding a hybrid that can effectively exploit opponents after a small number of interactions.

Our experiments in full-scale two-player limit Texas Hold-'em poker show that DBBR is effective in practice against a variety of opponents, including several entrants from recent AAAI computer poker competitions. DBBR achieved a significantly higher win rate than an approximate-equilibrium strategy against all of the opponents in our experiments. Furthermore, it achieved a higher win rate against the opponents from previous competitions than all of the entrants from that year's competition achieved (except for at most one). We compared three different algorithms for constructing the opponent model, and conclude that our custom weight-shifting algorithm outperforms algorithms that employ weighted L_1 and L_2-distance minimization.

While DBBR is able to effectively exploit weak opponents, it might actually become significantly exploitable to strong opponents (e.g., opponents who operate in a finer-grained abstraction). Thus, we would like to only attempt to exploit weak opponents, while playing the equilibrium against strong opponents. This can be accomplished by periodically looking at the win rate, and only attempting to exploit the opponent if a win rate above some threshold is attained. Our current work involves developing automated schemes that alternate between DBBR and equilibrium play based on the specific opponent at hand. In addition, DBBR could be extended to the setting where the opponent's private information from the previous game iteration is sometimes observed. Finally, future work could look at more robust versions of DBBR, where the opponent model allows the opponent to sometimes deviate from his observed action probabilities, or a safer strategy than the actual best response is used.

7. REFERENCES

[1] Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating game-theoretic optimal strategies for full-scale poker. *IJCAI*, 2003.

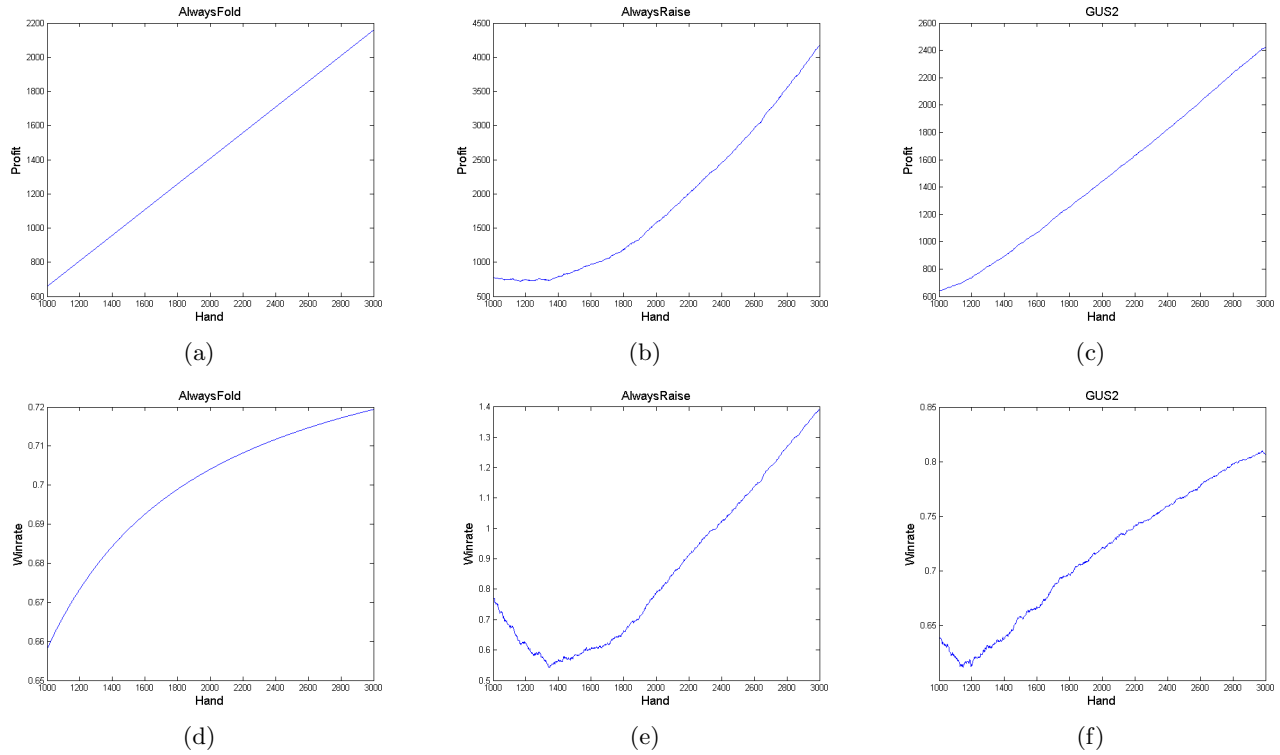


Figure 1: Profits and win rates over time of DBBR-WS against several opponents. Results against AlwaysFold are shown in Figures 1(a) and 1(d), results against AlwaysRaise are shown in Figures 1(b) and 1(e), and results against GUS2 are shown in Figures 1(c) and 1(f). The top three graphs show profit over time, and the bottom three show win rates over time.

[2] Doran Chakraborty and Peter Stone. Convergence, targeted optimality, and safety in multiagent learning. *ICML*, 2010.

[3] Aaron Davidson, Darse Billings, Jonathan Schaeffer, and Duane Szafron. Improved opponent modeling in poker. *IJCAI*, 2000.

[4] Andrew Gilpin and Tuomas Sandholm. A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. *AAAI*, 2006.

[5] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker. *AAAI*, 2007.

[6] Andrew Gilpin, Samid Hoda, Javier Peña, and Tuomas Sandholm. Gradient-based algorithms for finding Nash equilibria in extensive form games. *WINE*, 2007. Extended version in *Math. of OR*, 2010.

[7] Bret Hoehn, Finnegan Southey, Robert C. Holte, and Valeriy Bulitko. Effective short-term opponent exploitation in simplified poker. *AAAI*, 2005.

[8] Michael Johanson and Michael Bowling. Data biased robust counter strategies. *AISTATS*, 2009.

[9] Michael Johanson, Martin Zinkevich, and Michael Bowling. Computing robust counter-strategies. *NIPS*, 2007.

[10] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *GEB*, 1996.

[11] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. *COLT workshop on Online Learning with Limited Feedback*, 2009.

[12] Peter McCracken and Michael Bowling. Safe strategies for agent modelling in games. *AAAI Fall Symposium on Artificial Multi-agent Learning*, 2004.

[13] Marc Ponsen, Marc Lanctot, and Steven de Jong. MCRNR: Fast computing of restricted Nash responses by means of sampling. *AAAI workshop on Interactive Decision Theory and Game Theory Workshop*, 2010.

[14] Marc Ponsen, Jan Ramon, Tom Croonenborghs, Kurt Driessens, and Karl Tuyls. Bayes-relational learning of opponent models from incomplete information in no-limit poker. *AAAI*, 2008.

[15] Tuomas Sandholm. Perspectives on multiagent learning. *Artificial Intelligence*, 2007.

[16] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. *UAI*, 2005.

[17] Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. *NIPS*, 2007.

False-name bidding in first-price combinatorial auctions with incomplete information

Atsushi Iwasaki, Atsushi Katsuragi, and Makoto Yokoo
Department of Informatics, Kyushu University
Fukuoka, 819-0395, Japan
{iwasaki@, katsuragi@agent., yokoo@}is.kyushu-u.ac.jp

ABSTRACT

False-name bids are bids submitted by a single agent under multiple fictitious names such as multiple e-mail addresses. False-name bidding can be a serious fraud in Internet auctions since identifying each participant is virtually impossible. It is shown that even the theoretically well-founded Vickrey-Clarke-Groves auction (VCG) is vulnerable to false-name bidding. Thus, several auction mechanisms that cannot be manipulated by false-name bids, i.e., *false-name-proof* mechanisms, have been developed.

This paper investigates a slightly different question, i.e., how do they affect (perfect) Bayesian Nash equilibria of first-price combinatorial auctions? The importance of this question is that first-price combinatorial auctions are by far widely used in practice than VCG, and can be used as a benchmark for evaluating alternate mechanisms. In an environment where false-name bidding are possible, analytically investigating bidders' behaviors is very complicated, since nobody knows the number of real bidders. As a first step, we consider a kind of minimal settings where false-name bids become effective, i.e., an auction with two goods where one naive bidder competes with one shill bidder who may pretend to be two distinct bidders. We model this auction as a simple dynamic game and examine approximate Bayesian Nash equilibria by utilizing a numerical technique. Our analysis revealed that false-name bidding significantly affects the first-price auctions. Furthermore, the shill bidder has a clear advantage against the naive bidder.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics, Theory

Keywords

Auction theory, mechanism design, first-price auctions, and false-name bidding

Cite as: False-name bidding in first-price combinatorial auctions with incomplete information, Atsushi Iwasaki, Atsushi Katsuragi and Makoto Yokoo, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011),* Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 541-548.

1. INTRODUCTION

In a *combinatorial auction*, also called *package auction*, multiple goods are simultaneously for sale, and, in general, bidders can express arbitrary valuation functions over subsets of the goods. This allows bidders to express substitutability and complementarity of the goods in their valuations. A recent book by Cramton *et al.* [3] gives a thorough survey of the theory and practice of combinatorial auctions. False-name bids [16] are bids submitted by a single agent under multiple fictitious names such as multiple e-mail addresses. False-name bidding can be a serious fraud in combinatorial auctions on the Internet, since identifying each participant is virtually impossible.

The Vickrey-Clarke-Groves (VCG) auction is best motivated by its dominant strategy property under incomplete information, that is, truth-telling by all bidders in the auction leads outcomes (allocation of goods) to be efficient. However, VCG has several limitations in environments with complementarities among goods. One is vulnerability to false-name bidding. As mentioned above, since such dishonest actions are very difficult to detect, they can cause even more serious problems in auctions on the Internet. Several auction mechanisms that cannot be manipulated by false-name bids (i.e., *false-name-proof* mechanisms) have been developed [15, 7, 6]. We say a mechanism is *false-name-proof* if, for each bidder, declaring his true valuation function using a single identifier is a dominant strategy, even though the bidder can choose to use multiple identifiers.

In this paper, we investigate a slightly different question. We know false-name manipulations can affect a dominant-strategy equilibrium of strategy-proof mechanisms, i.e., VCG is not false-name-proof. How do they affect (perfect) Bayesian Nash equilibria of other non-direct-revelation mechanisms, in particular, the first-price combinatorial auction mechanism? The importance of such analysis is that first-price combinatorial auctions are by far widely used in practice than VCG, and the obtained results can be used as a benchmark for evaluating other auctions/mechanisms. In first-price auctions, bidders simply submit sealed bids, they are allocated the goods so that the combination of bids maximizes the seller's revenue, and each winning bidder pays the amount of the associated bid. However, it is not so far investigated how false-name bidding affects first-price combinatorial auctions.

In an environment where false-name bids are possible, analytically investigating bidders' behaviors is very complicated, since bidders are asymmetric and nobody knows the number of real bidders. Bidders are asymmetric if their val-

ues are drawn from asymmetric distributions. Much of the motivation in investigating false-name bidding arises from environments where bidders have complementarities among goods. The equilibria in first-price auctions do not have a well-known closed-form solution. Accordingly, many approaches by computer scientists and economists have been developed to approximate an equilibrium strategy. Seminal works by Wellman and his colleagues have developed techniques to obtain an analytically intractable Nash equilibrium in *empirical mechanism design* [14, 12, 8]. Those have recently been used to design and evaluate alternate mechanisms [13, 10]. Armantier *et al.* advocated a similar technique called a *constrained strategic equilibrium* approach [1].

False-name bidding affects first-price auctions in a different way than VCG auctions. At first glance, false-name bidding seems not effective in first-price auctions. In a first-price combinatorial auction, if a bidder wins, he pays the amount of his bid. Assume a (potential shill) bidder can win two goods X and Y with bid $b^{\{X,Y\}}$. Assume he uses false-names, splits his bid, and obtains X and Y separately by bid $b^{\{X\}}$ and $b^{\{Y\}}$, respectively. As far as $b^{\{X,Y\}} = b^{\{X\}} + b^{\{Y\}}$, his payment does not change. However, the behaviors of other bidders might be influenced by false-name bidding. Let us assume there exists a competing bidder (denoted as bidder 1) who also wants X and Y. For bidder 1, his bidding strategy changes if his belief about his opponents changes. In short, his bid decreases when he thinks he is facing two opponents, each of whom wants either X or Y, compared to the case where he thinks he is facing one opponent who wants both X and Y. This is because, when there exist two (real) bidders, each tries to *free-ride* the other bidder's effort; neither raises his bid in the hopes that the other raises his bids high enough to beat bidder 1 [11]. Thus, the total of these two bidders' bids tends to be small. Then, bidder 1 can safely decrease his bid. Consequently, when the shill bidder pretends to be two bidders, bidder 1 decreases his bid. The shill bidder can take advantage of this fact.

It is very complicated to construct a game of auctions with false-name bidding. In the analysis of auctions with incomplete information, it is assumed that each bidder knows how many bidders are participating before an auction begins. This is because we require the cumulative distribution function of each bidder as common knowledge to solve such auctions. Therefore, we must properly model how many identifiers a shill bidder uses and when bidders know the number of bidders.

As a first step, we consider a very simple and stylized model where false-name bids become effective, i.e., an auction with two goods where one naive bidder competes with one shill bidder who may pretend to be two distinct bidders. We model this auction as a dynamic game with incomplete information. We then examine approximate Bayesian Nash equilibria when bidders' preferences are drawn from asymmetric distributions, by utilizing the CSE approach [1].

This paper provides novel insights into the properties of first-price auctions in environments where false-name bidding is possible. The numerical results suggest that false-name bidding in first-price auctions can dramatically reduce the revenue and does not reduce the surplus so much. Furthermore, a shill bidder can highly increase his profit using two identifiers, while a naive bidder can keep his profit, though he is less likely to defeat the shill bidder.

Let us briefly describe the organization of this paper. Section 2 formalizes the first-price combinatorial auctions and the solution concepts. Section 3 constructs a dynamic game of auctions with false-name bidding. Section 4 shows the numerical results of equilibrium bidding strategies. Section 5 examines the effect of false-name bidding in terms of the major properties of auctions. Section 6 concludes this paper.

2. PRELIMINARIES

2.1 First-price combinatorial auctions

In a first-price single-item auction, each agent i submits sealed bid b_i for a good valued by agent i at v_i. Among all agents, the agent with the highest bid wins the good (ties are broken randomly). In a combinatorial auction setting, the auction is also called a *menu auction*. Bernheim and Whinston developed a theory of sealed-bid, first-price combinatorial auctions [2]. Let us consider a first-price combinatorial auction with two goods X and Y.

1. Each bidder i submits sealed bids $b_i = (b_i^{\{X\}}, b_i^{\{Y\}}, b_i^{\{X,Y\}})$ on $\{X\}$ only, $\{Y\}$ only, and the set/bundle of $\{X, Y\}$.
2. The auctioneer chooses an allocation, so that the combination of bids maximizes the seller's revenue.
3. Each winning bidder pays the amount of the associated bid.

We also assume a *quasi-linear, private value* model with *no allocative externality*. The utility (profit) of bidder i, if he wins either X or Y with $b_i^{\{X\}}$ or $b_i^{\{Y\}}$, is $v_i^{\{X\}} - b_i^{\{X\}}$ or $v_i^{\{Y\}} - b_i^{\{Y\}}$; and the utility of bidder i, if he wins both goods with $b_i^{\{X,Y\}}$, is $v_i^{\{X,Y\}} - b_i^{\{X,Y\}}$.

A losing bidder obtains nothing, pays zero, and thus, his utility is zero, since we assume *normalization*. The seller revenue, i.e., the utility of the auctioneer, is the sum of the payments of the winning bidders.

2.2 Equilibrium concepts

We use two of the most prevalent solution concepts from game theory: *Bayesian Nash equilibrium* (BNE) and *perfect Bayesian equilibrium* (PBE). BNE are used to analyze games with incomplete information, or Bayesian games, e.g., analysis of non-direct-revelation mechanisms, such as first-price auctions. A bidder's (expected) profit depends not only on the bids of other bidders but also on information that is only partly known to the bidder, i.e., a distribution function on the values of other bidders. Furthermore, PBE is a refinement of BNE for dynamic games, which is required to describe environments where false-name bidding is possible: a shill bidder can use multiple identifiers.

Let us define BNE in an auction with bidder 1 and 2 in environments where false-name bidding is not possible. Bidder i assigns a value of v_i for each combination of goods on sale drawn from a cumulative probability distribution with function F_{v_i} and associated probability density function f_{v_i}. Bidder i knows his own value v_i and only that any other bidder $j(\neq i)$'s value is independently distributed based on F_{v_j}. Thus, F_{v_j} for all other bidder j and the number of them are common knowledge. In general, the distribution of valuations is assumed to be the same for all bidders, i.e., symmetric. However, since it must be asymmetric in auctions with false-name bidding, we do not specify the distribution here.

A bidding strategy for bidder i is defined as a function s_i. For example, bidder i with v_i submits $b_i = s_i(v_i)$. The inverse function of s_i is denoted as s_i^{-1}. Any other bidder j's strategy s_j is assumed to be increasing and differentiable. To draw bidder i's bid b_i, we can obtain the cumulative probability distribution function F_{b_i} and the associated density function f_{b_i} for an arbitrary value of b:

$$F_{b_i}(b) = F_{v_i}(s_i^{-1}(b)) \text{ and } f_{b_i}(b) = \frac{f_{v_i}(s_i^{-1}(b))}{\frac{d}{db}s_i(s_i^{-1}(b))}.$$

The expected profits of bidder $i \in \{1, 2\}$ for given v_i and s_i are calculated as follows:

$$U_i(b_i, s_j; v_i) = (v_i - b_i)F_{b_j}(b_i) \text{ for all } i.$$

From these expected profit, we define a BNE in the auction with bidder 1 and 2.

Definition 1 (Bayesian Nash equilibrium) *A profile of bidding strategies (s_1^*, s_2^*) consists of a Bayesian Nash equilibrium in an auction with bidder 1 and 2 if*

$$\begin{aligned}&\forall v_1, \forall v_2, \forall s_1, \forall s_2,\\ &U_1(s_1^*(v_1), s_2^*; v_1) \geq U_1(s_1(v_1), s_2^*; v_1), \text{ and}\\ &U_2(s_2^*(v_2), s_1^*; v_2) \geq U_2(s_2(v_2), s_1^*; v_2).\end{aligned}$$

The profile of the strategies maximizes the expected profit of each bidder when the probabilistic distribution of values and the number of bidders are common knowledge.

If a shill bidder can use multiple identifiers, the bidders' equilibrium strategies become significantly more intricate. In the analysis of auctions with incomplete information, it is assumed that each bidder knows the number of participating bidders before an auction begins, as common knowledge. However, if the shill bidder may pretend to be multiple distinct bidders, it is essential for a naive bidder to consider the number of real bidders. For example, when the naive bidder faces two bids, he may think that those come from a shill bidder using two false identifiers or he may think they come from two distinct bidders. As a result, a bidder's (expected) profit comes to depend on the prior distribution of others' values and the partial information about the number of real bidders. To model this, we construct a dynamic game and focus on the PBE analysis in the later section.

PBE is the most commonly used for analyzing sequential (dynamic) games with observed actions and private types (values) [4]. Each bidder has a strategy s_i and *beliefs* that are represented as a cumulative probability distribution function about values of other bidders. A strategy profile s_i is a PBE if each bidder updates his beliefs using Bayes rule whenever possible (*consistency*) and, whenever it is bidder i's turn to move, s_i prescribes an action that maximizes i's expected payoff from then on, given i's beliefs (*sequential rationality*).

As a first step, we consider a very simple and stylized model where we restrict the number of false identifiers each bidder can use and each bidder's observable information. This is because computing a PBE is intractable in environments where false-name bidding becomes effective. Thus, in subgames of the restricted dynamic game, we can compute a BNE strategies by utilizing a numerical technique that enables one to approximate an analytically intractable Nash equilibrium in a broad class of games with incomplete information.

2.3 Constrained strategic equilibrium

This section briefly describes a solution concept for games with incomplete information, called *constrained strategic equilibrium* (CSE) [1]. The sequence of CSEs approximates an equilibrium and CSE provides a useful way to numerically compute BNE for games whose solutions cannot be analytically derived.

We consider a single play of an two-person simultaneous-move game. Let $N = \{1, 2\}$ denote a set of bidders (players). The subscript i denotes a specific player $i \in N$, and the subscript j refers to the player except i. CSE is defined as a Nash equilibrium of a modified game in which strategies are constrained to belong to an appropriate subset typically indexed by an auxiliary parameter vector. Let us denote S as a subset of all feasible strategy profiles and S^k as a set of constrained strategy profiles for parameter k. Formally,

Definition 2 (Constrained strategic equilibrium) *Let $S^k = \{S_1^k, S_2^k\}$ for a parameter k denote a set of constrained strategy. $S^{k*} \subset S^k$ is the set of CSEs if $\forall s_i^k \in S_i^k$ and $\forall i \in N$, $\tilde{U}_i(s_i^{k*}, s_j^{k*}) \geq \tilde{U}_i(s_i^k, s_j^{k*})$ where $\tilde{U}_i$ is the expected utility of player i.*

Armantier *et al.* [1] identified a compacity condition under which a sequence of CSEs converges toward a Nash equilibrium.

Proposition 1 ([1]) *If an expected utility $\tilde{U}$ is continuous and if a sequence of CSEs $\{s^{k*}\}_{k=1\to\infty}$ has a subsequence with limit $\bar{s} \in S$, then $\bar{s}$ is a Nash equilibrium.*

Corollary 1 ([1]) *If a set of strategy profiles S is compact, $\tilde{U}$ is continuous and there exists a CSE strategy s^{k*} for all $k > 0$, then there exists a Nash equilibrium in S, and any sequence of CSEs $\{s^{k*}\}_{k=1\to\infty}$ has a subsequence that converges toward a Nash equilibrium.*

The compacity of the strategy space is standard in games with incomplete information and it applies to a large class of games including several auction models, such as asymmetric first-price auctions. The numerical technique enables one to approximate an analytically intractable Nash equilibrium in such a class. CSE also has an approximation algorithm that can be applied for asymmetric games with incomplete information. Let us briefly show the algorithm:

1. Consider a family of parameterized constrained strategies: $s_i^k(v_i) = s_i(d_i^k, v_i) \in S_i^k$, with $d_i^k \in D_i^k \subset \mathbb{R}^{\gamma(k)}$. Note that $\gamma(k)$ is a function of the final dimension and is set to 2^{k-1}.

2. Maximize player i's expected utility after fixing parameter d_j^k. The approximation of the expected utility for d_j^k is defined as

$$\tilde{U}_i^M(s_i(d_i^k, v_i)) = \frac{1}{M}\sum_{m=1}^{M} V_i(d^k, \tilde{v}^m)G_i(s_i(d_i^k, \tilde{v}_i^m)).$$

where V_i is the utility function of player i with value $\tilde{v}^m$, which denotes a vector of two values drawn randomly ($N = 2$). M is the Monte Carlo size and G_i is the cumulative probability distribution function that player i wins the game (auction) when he takes $s_i(d_i^k, \tilde{v}_i^m)$.

3. Step 2 is repeatedly applied for each player. If d^k for all i is not updated, this algorithm stops.

In most applications, G_i cannot be calculated analytically and needs to be approximated by a kernel density estimation. The kernel density estimation is a non-parametric way of estimating G_i by L sample drawn from the distribution of values F. Let $\{v^1, \ldots, v^L\}$ denote L sample drawn from F. The kernel density estimation $\hat{f}_h(v)$ is shown by the following equation:

$$\hat{f}_h(v) = \frac{1}{Lh} \sum_{l=1}^{L} K(\frac{v - v^l}{h}),$$

where $K(\cdot)$ is the Gaussian distribution as a kernel function and h is a smoothing parameter called a bandwidth

Notice that the accuracy of kernel density estimation depends on bandwidth h. For example, if you enlarge h more than an appropriate value, $\hat{f}$ loses its feature – and vice versa. However, it is difficult to find the optimal bandwidth because it heavily depends on the structure of problems. In this paper, we use the following equation that achieves empirically good accuracy [5].

$$h = (\int K(t)^2 dt)^{1/5} (\frac{3}{8\sqrt{\pi}} \sigma^{-5})^{-1/5} L^{-1/5}.$$

where σ is the sample variance of L.

3. A DYNAMIC GAME WITH FALSE-NAME BIDDING

This section illustrates the PBE analysis through a 2- or 3-bidder combinatorial auction with two different goods, X and Y. Consider a dynamic game shown in Figure 1 with two stages: identifier-choice and bidding. First, with probability p, bidder 1 and 2 participate in an auction ($N = 2$), and with probability $1 - p$, bidder 1, 3, and 4 participate ($N = 3$). Assume that bidders have no knowledge about probability p. Second, at the identifier-choice stage, each bidder chooses how many identifiers he uses, and, in practice, only bidder 2 can choose one or two identifiers. Last, at the bidding stage, each bidder bids after observing the number of participating bidders, which may include false identifiers.

We also need to define a type that each bidder receives in games with incomplete information to provide each bidder with strategy space and information. We assume that the type determines the value for each combination of auctioned goods and the number of identifiers he can use.

Let us define types of bidder 1-4 and the observable information in the following. Bidder 1 values only the set of two goods drawn from the sum of two uniform distributions on interval $[0, 1]$, $Uni(0, 1)$:

$$(v_1^{\{X\}}, v_1^{\{Y\}}, v_1^{\{X,Y\}}) = (0, 0, Uni(0, 1) + Uni(0, 1)),$$

each of which is drawn independently. At the identifier-choice stage, he does nothing, since he can use only a single identifier. He also has a belief about how many bidders are participating as probability p^1, with which he is competing with bidder 2. Here, p^1 does not always be true, i.e., p^1 may not be equal to the true probability p. Before the bidding stage, he observes the number of bidders. When he observes one other bidder, he realizes that his opponent is bidder 2

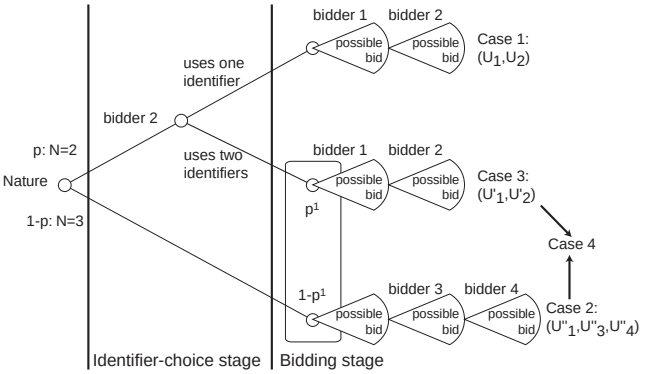


Figure 1: A dynamic game of an auction with false-name bidding

and bids to maximize his profit conditional on his belief about bidder 2's value. On the other hand, when he observes two other bidders, he realizes that his opponents are false identifiers of bidder 2 with probability p^1, or that they are distinct bidders of bidder 3 and 4, with $1-p^1$. Consequently, he bids to maximize his profit conditional on his joint belief about bidder 3 and 4s' value.

Bidder 2 positively values both $\{X\}$, $\{Y\}$, and $\{X, Y\}$. Each value on $\{X\}$ and $\{Y\}$ is independently drawn from $Uni(0, 1)$ and value on $\{X, Y\}$ is their sum:

$$(v_2^{\{X\}}, v_2^{\{Y\}}, v_2^{\{X,Y\}}) = (Uni(0, 1), Uni(0, 1), v_2^{\{X\}} + v_2^{\{Y\}}).$$

Thus, bidder 1 and 2 have a symmetric distribution on $\{X, Y\}$. At the identifier-choice stage, bidder 2 can use one or two identifiers and knows that $N = 2$ was chosen because he himself participated. He also exactly knows what information bidder 1 observes. When he uses one identifier, he knows that bidder 1 realizes that no bidder uses false identifiers. When he uses two identifiers, he knows that bidder 1 has that belief p^1 about bidder 2's presence. At the bidding stage, bidder 2 bids based on that information and his belief about bidder 1's value.

Bidder 3 and 4 can use only a single identifier. We only explain bidder 3's case only, since bidder 3 and 4 have almost identical information and value except the good he desires. Bidder 3 values only $\{X\}$ and $v_3^{\{X\}}$ is drawn from $Uni(0, 1)$. After the identifier-choice stage, he knows that $N = 3$ was chosen because he himself participated. At the bidding stage, bidder 3 bids based on that information and his joint belief about bidder 1 and 3s' values.

We explore the strategies in four specific subgames for some p and p^1 to effectively show how bidders' behaviors change. Then, we calculate bidders' expected profits in a subgame of the dynamic game by utilizing the CSE approximation algorithm.

Case 1: 2 bidders - 2 identifiers ($p = 1$).

> With probability $p = 1$, bidder 1 and 2 participate ($N = 2$) and bidder 2 always chooses to use a single identifier. Since bidder 1 and 2 use a single identifier and submit their bids, no false-name bidding occurs. They obtain profits of U_1 and U_2.

Case 2: 3 bidders - 3 identifiers ($p = 0$ and $p^1 = 0$).

With probability $p = 0$, bidder 1, 3 and 4 participate ($N = 3$), always use a single identifier, and submit their bids, knowing that bidder 1 believes that no false-name bidding occurs ($p^1 = 0$). They obtain profits of U_1'', U_3'' and U_4''.

Case 3: 2 bidders - 3 identifiers ($p = 1$ and $p^1 = 0$).

With probability $p = 1$, bidder 1 and 2 participate ($N = 2$), and bidder 2 always chooses to use two identifiers. Thus, three identifiers submit their bids. Since bidder 1 believes that no false-name bidding occurs ($p^1 = 0$), he chooses the same bidding strategy as in Case 2. Bidder 2 takes the best response to the bidding strategy of bidder 1. Bidder 1 and 2 obtain profits of U_1' and U_2'. Note that, for bidder 2, the expected profit when he uses two identifiers is always better than when he uses a single identifier; for bidder 2, the strategy using two identifiers is PBE.

Case 4: 2 or 3 bidders - 3 identifiers ($p = 1/2$ and $p^1 = 1/2$).

Case 4 stochastically combines Case 2 and 3 where, with $p = 1/2$, Case 3 occurs, and with $1 - p = 1/2$, Case 2 occurs. No bidder knows exactly the probability, but every bidder knows bidder 1's belief about Case 2 or 3 occurs ($p^1 = 1/2$). Except bidder 1, all bidder take a best response to bidder 1's bidding strategy in which he considers false-name bidding.

4. NUMERICAL RESULTS

This section illustrates the PBE bidding strategies in Case 1-4, which are the consequences of the dynamic game described in Section 3. For comparison, we also note the corresponding strategies in VCG auctions in Appendix A.1. Since the values of the bidders on $\{X, Y\}$ in Case 1 are drawn from symmetric distributions, PBE has a well-known closed-form solution. On the other hand, those in Case 2-4 are drawn from asymmetric distributions. Thus, we theoretically show the PBE bidding strategies in Case 1 and numerically show them in Case 2-4 by utilizing the CSE approximation method.[1] The required parameters are set to $k = 5$, $M = 1000000$, and $L = 1000$.

4.1 Case 1: 2 bidders – 2 identifiers

Let $v \in [0, 2]$ be a value on the bundle of $\{X, Y\}$ for bidder 1 and 2 and let $s(v) : R^+ \to R^+$ be a mapping function of the value to the bid. Since PBE in Case 1 has a closed-form solution, we can theoretically derive the equilibrium bidding strategy $s(v)$ [9]:

$$s(v) = \begin{cases} \frac{2}{3}v & \text{if } 0 \le v \le 1, \\ \frac{2}{3}(v + 1 + \frac{2v-1}{v^2-4v+2}) & \text{if } 1 < v \le 2. \end{cases}$$

The red line in Figures 2-5 shows this bidding strategy, which is labeled as "Case 1: symmetric bidder." Bidder 1 with low value ($v < 1$) shades his bid to two-thirds of his value, and bidder 1 with high value ($v > 1$), further shades his bid as his value increases.

[1]In addition to the CSE approximation method, we examined these cases by a similar algorithm to [12] and obtained almost identical results.

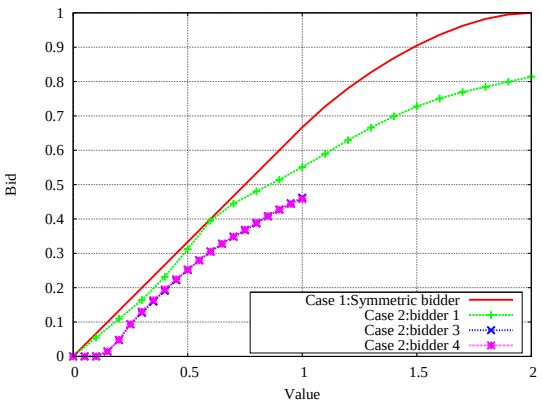


Figure 2: Bidding strategies in Case 2

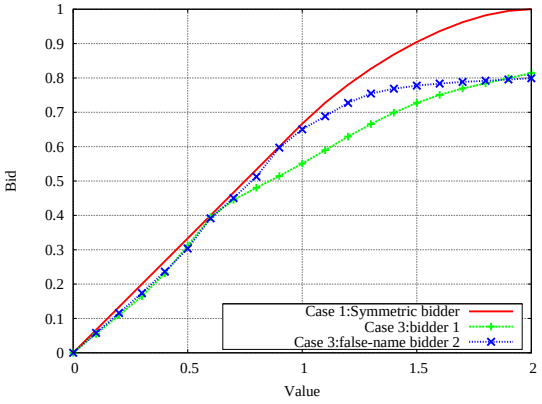


Figure 3: Bidding strategies in Case 3

4.2 Case 2: 3 bidders – 3 identifiers

Unlike Case 1, the values of all bidders are drawn from asymmetric distributions. In fact, bidder 1 values $[0, 2]$ only at $\{X, Y\}$, and bidder 3 and 4 values $[0, 1]$ at $\{X\}$ and $\{Y\}$, respectively. In general, there are no closed-form solutions for this case, but it can be easily solved by appropriate numerical methods. Figure 2 illustrates the bidding strategies of bidder 1, 3, and 4 with respect to their realizations of the values drawn from each distribution (blue and pink lines labeled as "Case 2: bidder 3" and "Case 2: bidder 4").

Bidder 1 with low value less than about 0.75, shades his bid to the same amount as in Case 1, and bidder 1 with high value reduces his bid more than in Case 1. The amount of reduction gradually increases as his value increases. Bidder 3 and 4 still shade their bids, in particular, with very low values, they prefer to bid zero.

This result is consistent with the *free-rider* problem in auctions [11]: A bidder does not raise his bid in the hopes that the other raises his bid high enough for that bidder to obtain a good. For example, bidder 3 and 4 value $\{X\}$ and $\{Y\}$, respectively. If the sum of their bids exceeds the amount of the bid on $\{X, Y\}$, bidder 3 and 4 win. Bidder 3 may expect bidder 4 to bid so high that bidder 1 loses and has an incentive to obtain $\{X\}$ with a low bid, and vice versa. Also, bidder 1 takes the best response to their shaded bids.

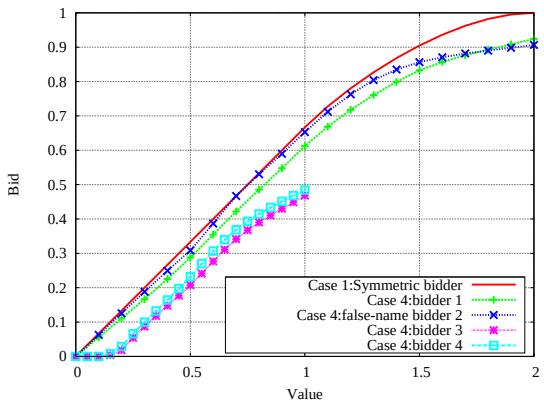


Figure 4: Bidding strategies in Case 4

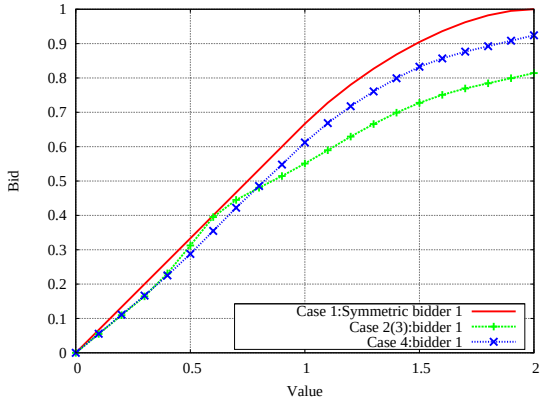


Figure 5: Bidding strategies of bidder 1

4.3 Case 3: 2 bidders – 3 identifiers

Case 3 has a more complicated strategy space, since bidder 2 can use two identifiers. Figure 3 illustrates the bidding strategies of bidder 1 and 2 (green and blue lines labeled as "Case 3: bidder 1" and "Case 3: false-name bidder 2"). Bidder 1's strategy is equivalent to the one in Case 2, since he believes that he is competing with two distinct bidders. Bidder 2 splits his bid into two bids on $\{X\}$ and $\{Y\}$ by using two identifiers. The blue line in Figure 3 indicates the sum of those two bids in $v_2^{\{X,Y\}}$.

Bidder 2 with low value ($v_2^{\{X,Y\}} < 0.75$) or very high value ($1.8 < v_2^{\{X,Y\}}$) prefers to bid almost the same amount as bidder 1. On the other hand, bidder 2 with intermediate values first submits a slightly higher bid than bidder 1, raises his bid, and gradually reduces toward bidder 1, as his value increases.

This result suggests that bidder 2 can increase his profit as a result of taking the best response to the distribution of bidder 1's value which is a joint distribution of two $Uni(0,1)$. With such a distribution, bidder 1 is most likely to have his value of 1 and is least likely to have 0 or 2. Thus, bidder 2 raises his bids around 1 to maximize his profit. Therefore, bidder 2 has enough opportunities of false-name bidding to increase his expected profit.

4.4 Case 4: 2 or 3 bidders – 3 identifiers

In Case 4, bidder 1 considers the possibility that two of his competing bids come from one shill bidder (bidder 2) conditional on his belief about the actual number of participating bidders, i.e., $p^1 = 1/2$. Figure 4 illustrates the bidding strategies of bidder 1-4. Note again that bidder 2's strategy is represented as the sum of two bids.

The doubt of bidder 1 that a shill bidder exists raises his bid, so it becomes much closer to that in Case 1. Bidder 1 averages his bidding strategies in Case 1 and 2 in Figure 5. Unlike bidder 1, bidder 2 can slightly raise his bid higher than bidder 1 because bidder 1 may not correctly suspect the number of real bidders. Thus, the opportunities of false-name bidding are reduced. As well as bidder 2, bidder 3 and 4 know bidder 1's strategy. With lower value, they bid slightly lower than in Case 2, but, with higher value, they bid slightly higher.

5. DISCUSSION

This section discusses obtained properties from the numerical results: the social surplus, the auctioneer's revenue, and the profits of bidders. We decide the values of bidders based on the settings in Case 1-4 and generate 10 million instances. Table 1 summarize the average properties when bidders take the equilibrium bidding strategies in first-price auctions. For comparison, we also note the corresponding results of VCG auctions in Appendix A.2.

From bidder 1's perspective, Case 1 ($N = 2$) and Case 2 ($N = 3$) are seemingly the same, since the (aggregated) values of the opponents are the same. However, in Case 2, bidder 3 and 4 try to free-ride each other and decrease their bids. Thus, bidder 1 lowers his bids to maximize his profit. As a result, bidder 1 successfully increases his profit from 0.233 to 0.334 (+43%). This also significantly decreases the revenue from 0.767 to 0.620 (-19%); the decrease of the surplus from 1.23 and 1.22 (-1%) is relatively small. The fact that the surplus does not significantly change means that the obtained allocation is nearly efficient. All bidders decrease their bids. Occasionally, bidder 1 wins when the efficient allocation is allocating goods to bidder 3 and 4, but this happens only when bidder 1's value is close to the sum of values of bidder 3 and 4.

Let us examine Case 3 where false-name bidding is possible, i.e., a naive bidder (bidder 1) and a shill bidder (bidder 2) participate. The naive bidder completely believes that he is competing with two bidders ($p^1 = 0$), and the shill bidder knows this fact and always uses two false identifiers. Recall that this behavior of bidder 2 consists of a PBE. The revenue of 0.681 is intermediate between Case 1 and 2 and it decreases from 0.767 in Case 1 to 0.681 (-11%). Here, bidder 1 believes that he is facing two small bidders. If they were real bidders, they would try to free-ride and lower their bids. Thus, bidder 1 also lowers his bid. However, bidder 2 optimizes his two bids against the wrong belief of bidder 1. Thus, false-name bidding by bidder 2 decreases the revenue. In contrast, the surplus hardly changes regardless of the existence of false-name bidding. This fact means that the obtained allocation is nearly efficient. All bidders decrease their bids. Occasionally, bidder 2 wins when the efficient allocation is allocating goods to bidder 1, but this happens only when bidder 2's value is close to bidder 1's value.

In addition, the existence of false-name bidding significantly affects the profits of bidders. Bidder 2 significantly increases his profit from 0.233 to 0.312 (+34%) by using

Table 1: Properties of Case 1-4 in first-price auctions

	Case 1	Case 2	Case 3	Case 4a	Case 4b
revenue	0.767	0.620	0.681	0.718	0.660
surplus (efficiency)	1.23 (100%)	1.22 (99%)	1.22 (99%)	1.23 (100%)	1.22 (99%)
profit (bidder 1)	0.233	0.334	0.226	0.244	0.319
profit (bidder 2)	0.233	-	0.312	0.269	-
profit (bidder 3)	-	0.134	-	-	0.122
profit (bidder 4)	-	0.134	-	-	0.122

false identifiers, but the profit of bidder 1 does not change much from Case 1, only from 0.233 to 0.226 (-3%). However, this is not what bidder 1 expected. If he were facing two real bidders, his profit would have been 0.334 (in Case 2). Bidder 2 steals a significant amount of bidder 1's profit by over-bidding bidder 1. Interestingly, the profit of bidder 2 in Case 3 (0.312) is relatively close to that of bidder 1 in Case 2 (0.334). Also, the profit of bidder 1 in Case 2 (0.334) is relatively close to the sum of profits of bidder 3 and 4 in Case 2 (0.384).

Let us examine Case 4 where a naive bidder (bidder 1) is suspicious of the number of real bidders. Bidder 1 is wondering if the two observed bids were submitted from two distinct bidders or one shill bidder. We categorize Case 4 as either Case 4a with false-name bidding or Case 4b without. These results are summarized in the last two columns of Table 1. In Case 4a, the revenue decreases from 0.767 in Case 1 to 0.718 (-6%), and it increases from 0.681 in Case 3 to 0.718 (+5%). Recall that bidder 1 in Case 4 takes an average bidding strategy of Case 1 and 2 under the suspicion of the actual number of participating bidders, i.e., $p^1 = 1/2$. Thus, bidder 1 raises his bids more than Case 3. By false-name bidding the profit of bidder 2 (0.269) is higher than in Case 1 (0.233). However, he cannot increase his profit (0.269) so much as in Case 3 (0.312). Accordingly bidder 1's suspicion effectively mitigates the decrease of revenue when a shill bidder may be present. The effect of false-name bidding is reduced by the fact that the naive bidder is aware of its possibility.

Let us turn to Case 4b where bidder 3 and 4 submit two distinct bids, considering the suspicion of bidder 1. Case 4b achieves the revenue of 0.660, and Case 2 does 0.620. The revenue increases by about +6%, but the profits of the bidders decrease, including bidder 3 and 4, who are the real bidders. In a contrast to Case 4a, the suspicion of bidder 1 increases the revenue and reduce the profits of all bidders.

It is worthy to note that, if the naive bidder can distinguish Case 4a and 4b for sure, false-name bidding is no longer profitable. This implies that, if your opponent is sure about your identity, it is useless that you pretend to be somebody else and there is no point using false-name bidding. However, since this is impossible on the Internet, a shill bidder can take advantage of false-name bidding. It is most effective when your opponent never imagines the possibility of disguise. Also, it is still effective if your opponent is aware of that possibility, but cannot distinguish a real person and a false identifier.

We have so far investigated situations where bidder 1's belief is correct ($p = p^1$), except Case 3 ($p = 1$ and $p^1 = 0$). Let us consider what happens if bidder 1's belief is incorrect ($p \neq p^1$). When p^1 increases in Case 3, bidder 1's belief gradually becomes correct for the probability of number of real bidders $p = 1$. Thus, the bidding strategies of bidder 1 and 2 change from Case 3 toward Case 1. If bidder 1 has $p^1 = 1$, the properties in Case 3 are identical to those in Case 1. On the other hand, when p^1 increases in Case 2 ($p = 0$ and $p^1 = 0$), bidder 1's belief gradually becomes incorrect. Then, if bidder 1's belief becomes $p^1 = 1/2$, the situation becomes identical to Case 4b. Bidder 1 increases his bid. As a result, the revenue increases and the profits of all bidders decrease.

6. CONCLUSION

This paper numerically analyzes how false-name bidding affects the outcomes in first-price combinatorial auctions. False-name bidding causes serious problems in the VCG auctions. However, to the best of the authors' knowledge, this is the first analysis about first-price combinatorial auctions. The game of first-price auctions is regarded as a game of incomplete information, which typically does not have a closed-form solution, except under such simplifying assumptions as symmetry among types of bidders. Thus, predicting the consequences of such games is often analytically intractable. In addition, the extension of games of auctions to games where false-name bidding is possible further complicates them. Therefore, we construct a dynamic game of auctions with false-name bidding and approximately solve the subgames in four specific settings.

We reveal how the existence of false-name bidding changes the equilibrium bidding strategies and its properties. The results suggest that false-name bidding in first-price auctions dramatically reduces the revenue and the profits of bidders who neither use nor are concerned about false-name bidding.

In future works, we will extend our analysis to a variety of empirical distributions and generalize the approximation algorithm to solve dynamic games of auctions with false-name bidding.

7. REFERENCES

[1] O. Armantier, J.-P. Florens, and J.-F. Richard. Approximation of Nash equilibria in Bayesian games. *Journal of Applied Econometrics*, 23(7):965–981, 2008.

[2] B. D. Bernheim and M. D. Whinston. Menu auctions, resource allocation, and economic influence. *The Quarterly Journal of Economics*, 101(1):1–31, 1986.

[3] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[4] D. Fudenberg and J. Tirole. Perfect Bayesian equilibrium and sequential equilibrium. *Journal of Economic Theory*, 53(2):236 – 260, 1991.

Table 2: Properties of Case 1-4 in VCG auctions

	Case 1	Case 2	Case 3	Case 4a	Case 4b
revenue	0.767	0.617	0.00	0.00	0.617
surplus (efficiency)	1.23 (100%)	1.23 (100%)	1.00 (81%)	1.00 (81%)	1.23 (100%)
profit (bidder 1)	0.233	0.233	0.00	0.00	0.233
profit (bidder 2)	0.233	-	1.00	1.00	-
profit (bidder 3)	-	0.192	-	-	0.192
profit (bidder 4)	-	0.192	-	-	0.192

[5] W. H. Greene. *Econometric Analysis*. Prentice Hall; 5th edition, 2002.

[6] A. Iwasaki, V. Conitzer, Y. Omori, Y. Sakurai, T. Todo, M. Guo, and M. Yokoo. Worst-case efficiency ratio in false-name-proof combinatorial auction mechanisms. In *AAMAS*, pages 633–640, 2010.

[7] A. Iwasaki, M. Yokoo, and K. Terada. A robust open ascending-price multi-unit auction protocol against false-name bids. *Decision Support Systems*, 39(1):23–39, 2005.

[8] P. R. Jordan, Y. Vorobeychik, and M. P. Wellman. Searching for approximate equilibria in empirical games. In *AAMAS*, pages 1063–1070, 2008.

[9] V. Krishna. *Auction Theory*. Academic Press, 2002.

[10] B. Lubin and D. C. Parkes. Quantifying the strategyproofness of mechanisms via metrics on payoff distributions. In *UAI*, pages 349–358, 2009.

[11] P. Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108(2):245–272, 2000.

[12] Y. Vorobeychik and M. P. Wellman. Stochastic search methods for Nash equilibrium approximation in simulation-based games. In *AAMAS*, pages 1055–1062, 2008.

[13] L. Wagman and V. Conitzer. Strategic betting for competitive agents. In *AAMAS*, pages 847–854, 2008.

[14] M. P. Wellman, J. Estelle, S. Singh, Y. Vorobeychik, C. Kiekintveld, and V. Soni. Strategic interactions in a supply chain game. *Computational Intelligence*, 21(1):1–26, 2005.

[15] M. Yokoo. The characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol. In *IJCAI*, pages 733–739, 2003.

[16] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.

APPENDIX

A. VCG COMBINATORIAL AUCTIONS

A.1 PBE bidding strategies

This subsection summarizes the consequences of Case 1-4 in VCG. Throughout Case 1 and 2, all bidders have a dominant strategy to bid their own values in VCG. In Case 3, bidder 1 believes that truth-telling is the dominant strategy, since he never considers the possibility of false-name bidding. In sharp contrast, bidder 2 uses two false identifiers and manipulates VCG so that he always obtain the goods and pays zero. This strategy is an equilibrium strategy as long as bidder 1 takes the truth-telling strategy.

For example, in our setting, the value on $\{X, Y\}$ of bidder 1 is drawn from a joint distribution on interval $[0, 2]$. Bidder 2 splits his bid and bids 2 on $\{X\}$ and $\{Y\}$, respectively. As long as bidder 1 keeps $s_1(v) = v$, bidder 2 has an equilibrium strategy to bid $b^{\{X\}}$ and $b^{\{Y\}}$, each of which is greater than or equal to $b_1^{\{X,Y\}}$, i.e., to bid 2 which is the maximum value of $b_1^{\{X,Y\}}$.

In Case 4, bidder 1 considers false-name bidding. When bidder 3 and 4 are present ($N = 3$), Truth-telling for them is clearly a dominant strategy as in Case 2. When bidder 2 is present ($N = 2$), as mentioned in Case 3, he has an equilibrium strategy so that he always win with zero payment as long as bidder 1 takes the truth-telling strategy. Against this strategy, we restrict our attention to a situation where bidder 1 takes the truth-telling strategy because bidder 1 has no chance to obtain any good. There exists no bidding strategy that outperforms the truth telling strategy, even if he is confident that his opponents are using false-name bidding. To be precise, we must consider situations so that bidder 1 over-bids to obtain the goods, which never increase his profit. Accordingly, for all bidders, an equilibrium bidding strategy in Case 4 is equivalent to Case 3.

A.2 Discussions

The results of VCG are much simpler than those of first-price auctions. The difference from Case 1 ($N = 2$) to Case 2 ($N = 3$) only depends on the payment rules. The revenue decreases from 0.767 to 0.617 (-20%), and the surplus remains unchange, since all bidders submit their own values in the equilibrium. The profit of bidder 1 also doesn't change, but the sum of the profits of bidder 3 and 4 in Case 2 exceeds the profit of bidder 2 in Case 1.

In Case 3 where false-name bidding is possible, the revenue is zero, and only the profit of bidder 2 is positive. As mentioned in Section 4, bidder 2 uses two false identifiers and manipulates the VCG outcome so that he always obtain the goods and pays zero. Also, the surplus drastically decreases from 1.23 to 1.00 (-19%). Note that the surplus of 1.00 equals the lowest achievable surplus in our setting. Furthermore, Case 4 inherits this result in Case 3. Case 4a corresponds to Case 3, and Case 4b corresponds to Case 2, even if bidder 1 is suspicious of the number of real bidders. These obtained results imply that false-name bidding is even more serious in VCG than in the first-price auctions, as existing theoretical considerations in mechanisms that have dominant-strategy equilibria. In other words, outcomes in VCG are significantly manipulated by false-name bidding, regardless whether bidder 1 considers the possibility of false-name bidding by bidder 2.

Teamwork

Metastrategies in the Colored Trails Game

Steven de Jong, Daniel Hennes, Karl Tuyls
Department of Knowledge Engineering
Maastricht University, Netherlands

Ya'akov (Kobi) Gal
Department of Information Systems Engineering
Ben-Gurion University of the Negev, Israel

ABSTRACT

This paper presents a novel method to describe and analyze strategic interactions in settings that include multiple actors, many possible actions and relationships among goals, tasks and resources. It shows how to reduce these large interactions to a set of bilateral normal-form games in which the strategy space is significantly smaller than the original setting, while still preserving many of its strategic characteristics. We demonstrate this technique on the Colored Trails (CT) framework, which encompasses a broad family of games defining multi-agent interactions and has been used in many past studies. We define a set of representative heuristics in a three-player CT setting. Choosing players' strategies from this set, the original CT setting is analytically decomposed into canonical bilateral social dilemmas, i.e., Prisoners' Dilemma, Stag Hunt and Ultimatum games. We present a set of criteria for generating strategically interesting CT games and empirically show that they indeed decompose into bilateral social dilemmas if players play according to the heuristics. Our results have significance for multi-agent systems researchers in mapping large multi-player task settings to well-known bilateral normal-form games in a way that facilitates the analysis of the original setting.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]; J.4 [**Social and Behavioral Sciences**]

General Terms

Design, Experimentation

Keywords

Colored Trails, Metastrategies

1. INTRODUCTION

Computer systems are increasingly being deployed in task settings where multiple agents interact and make decisions together—whether collaboratively, competitively or in between—in order to accomplish individual and group goals. Often, such interactions can be modeled and analyzed in terms of complex game-theoretic games. A fundamental problem when performing an analysis of such games is dealing with large action spaces. Once we go beyond typical two-player two-action normal form games, the curse of dimensionality occurs in terms of finding equilibria and analyzing dynamics. For example, when analyzing the evolutionary dynamics of auctions or Poker, we need to abstract over atomic actions by introducing metastrategies [18, 16, 14], thus reducing large-scale interactions to smaller games.

Recently a new testbed has been introduced to enable evaluation and comparison between computational strategies for a wide variety of complex multi-agent task settings, i.e. the Colored Trails (CT) framework [3].[1] CT has spawned many publications in diverse multi-agent settings, such as repeated negotiation, interruption management, team formation and space research [4, 12, 10]. CT is particularly attractive because it is grounded in a situated task domain and is rich enough to reflect features of real-life interactions. The CT framework encompasses a family of different games that provide an analogue to the ways in which goals, tasks and resources interact in real-world settings. CT is parametrized to allow for increasing complexity along a number of dimensions, such as task complexity, the availability of and access to information, the dependency relationships that hold between players, and the communication protocol. In abstracting from particular domains, CT provides a general framework to analyze multi-agent interactions.

In a similar vein as the aforementioned work in e.g. auctions or Poker, this paper suggests a way of reducing multi-player interactions in the CT framework to a set of smaller games. It provides a mapping between a particular CT task setting and normal-form games in a way that preserves much of the strategic qualities of the original setting. To this end, it defines a set of heuristic metastrategies for each player that are domain-independent and make minimal assumptions about the way other players make decisions. These metastrategies allow a reduction to canonical bilateral social dilemma games taking place between (pairs of) players, i.e., Stag Hunt, Prisoners' Dilemma, and Ultimatum games. In these games, the metastrategies correspond to Nash equilibria and/or Pareto-optimal strategies. The mapping from CT game instances to well-known social dilemmas allows to compare participants' behavior in CT with prior results from these smaller, more traditional settings. Given the mapping of the CT game to social dilemmas, our analysis is extended by assessing the effect of adding social factors to participants' decision-making.

In the paper, we also lay down a set of criteria that make generated CT game instances strategically interesting for human players (e.g., they enforce negotiation). Results from simulation experiments that sample thousands of such strategically interesting CT game instances confirm that participants' outcomes from playing metastrategies in the original game instances correspond to the outcomes from playing the same strategies in the reduced Prisoners' Dilemma, Stag Hunt, and Ultimatum games. The results in this paper have significance for agent-designers in that they facilitate the comparison of computational strategies in different task settings with results obtained in more traditional idealized settings. Moreover they allow to generate new types of interactions in task settings that meet scientifically and strategically interesting criteria.

[1] Colored Trails is free software and is available for download at http://www.eecs.harvard.edu/ai/ct. A complete list of publications can also be found at this link.

Cite as: Metastrategies in the Colored Trails Game, Steven de Jong, Daniel Hennes, Karl Tuyls, and Ya'akov Gal, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 551-558.

2. RELATED WORK

The idea to consider aggregate or metastrategies for facilitating (game-theoretic) analysis of a complex game is not new. In related work, strategies are often aggregated using heuristics, allowing the construction of e.g. *heuristic payoff tables* [18, 19]. Generally, a normal-form-game payoff matrix is replaced by a heuristic payoff table, since assembling all possible actions into a matrix is impractical for complex games (the resulting matrix would have too many dimensions). A heuristic allows to define metastrategies over the atomic actions, reducing the number of actions that have to be explicitly taken into account. A metastrategy typically represents a philosophy, style of play, or a rule of thumb.

Recent domains in which the heuristic approach has been followed include auctions [17, 11] and Poker [16]. In these domains, expert knowledge is available to assist in the establishment of suitable heuristics. For instance, in auctions, there are many well-known automated trading strategies such as Gjerstad-Dickhaut, Roth-Erev, and Zero Intelligence Plus [15, 14]. In Poker, experts describe metastrategies based on only a few features, such as players' willingness to participate in a game, and players' aggression-factor once they do participate. Examples of metastrategies in Poker, based on these features, are the tight-passive (a.k.a. Rock), tight-aggressive (a.k.a. Shark), loose-passive (a.k.a. Fish) and loose-aggressive (a.k.a. Gambler) metastrategies. Depending on the actions taken by a player over a series of games, it may be categorized as belonging to a specific type of player, i.e., as using a certain metastrategy. This allows researchers to analyze real-world Poker games, in which the metastrategy employed by each player in a particular series of games can be identified. Subsequently, obtained payoffs in this series of games may be used to compute heuristic payoff tables for each metastrategy [16]. These tables then allow to study the evolutionary dynamics of Poker.

In this paper, we pursue a similar approach, although a lack of heuristic expertise implies that we need to first perform an in-depth study of the game and possible means of aggregating strategies. Since expert knowledge on heuristics within the CT framework is not available, we cannot readily label a certain chip exchange as being, e.g., an egocentric or a social one. We aim to provide an analysis that does allow us to label chip exchanges in this manner.

We discuss three distinct levels within the CT framework. On the highest level, we have the complete *framework* itself, i.e., all possible CT games. The intermediate level identifies a certain *game* within the framework, e.g., the three-player variant we study in this paper. The lowest level is a *game instance*, e.g., one specific board configuration with a certain allocation of chips and a certain position for each of the three players and the goal. Going up from the lowest level, we see that players can perform certain *actions* in a CT game instance, can adhere to certain *strategies* in a CT game, and can use certain *metastrategies* in the CT framework.

While we restrict our analysis to one CT game (the three-player variant discussed below), the same analysis also applies to other games within the framework. Therefore, the analysis indeed leads to the identification of metastrategies. These metastrategies may be used as a solid basis to come up with heuristic payoff tables.

3. COLORED TRAILS

We focus on a three-player negotiation variant [2] of CT that includes a board of 4x4 squares, colored in one of five colors. Each player possesses a piece located on the board and a set of colored chips. A colored chip can be used to move a player's piece to an adjacent square (diagonal movement is not allowed) of the same color. The general goal is to position pieces onto or as close as possible to a goal location indicated by a flag. Each player receives points purely based on its own performance. There are three distinct players in the game: two proposers ($P1$ and $P2$) and a responder (R). Figures 1(a) and 1(c) show two examples of game instances. The two instances will be used as running examples throughout the paper. Game instances include game boards with goal and player locations, as well as the chip sets that have been allocated to each player. The CT game is divided into a sequence of three phases and ends with an automatic evaluation.

Initial phase. The game board and the chip sets are allocated to the players. This initial phase allows participants to locate their own piece on the board and reason about the game. For example, in Figure 1(a), proposer $P1$ is missing a green chip to get to the goal (by moving left-up-up), proposer $P2$ is missing a gray or green chip (moving up-up-right or up-right-up) to get to the goal, and responder R is missing a gray chip and a blue chip to get to the goal (moving right-3up-right). The game state is fully observable at this point, except that proposers cannot see each other's chips.

Proposal phase. The two proposers can make chip exchange offers to the responder. Both proposers make offers to the responder simultaneously; they cannot observe each other's offer.

Reaction phase. The responder is presented with the two proposals. It can only accept one or reject both proposals and is not allowed to make a counter-proposal.

Termination and scoring phase. In this phase, players automatically exchange chips if they have reached agreement, and the icon of each player is advanced as close as possible towards its goal (using the Manhattan path with the shortest distance) given the result of the negotiation. The game ends and scores are automatically computed for each player: for every step between the goal and the player's position, 25 penalty points are subtracted. For every chip the player has not used, it receives 10 extra points.

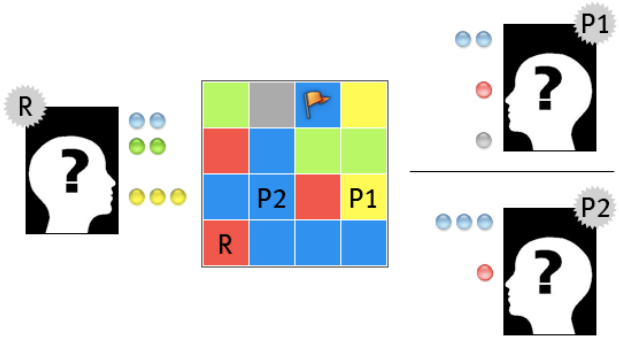
In the current paper, we use the following terminology associated with scores. First, the *base score* for a player $p \in \{R, P1, P2\}$ is the score the player receives when there is no agreement.[2] Second, the *gain* for a player p and a chip exchange proposal s denotes the difference in the score in the game (given that s is realized) and the base score, and is denoted as $G_p(s)$.The base score for p, i.e., the gain when there is no agreement, is denoted as $G_p(\emptyset)$.

For example, in Figure 1(a), $G_{P1}(\emptyset) = -20$. This is because if there is no agreement, the player can only move one square to the left by using its red chip. It is still two squares away from the goal, yielding $2 \times 25 = 50$ penalty points. It has 3 remaining chips, yielding $3 \times 10 = 30$ points. In this particular game, $G_{P2}(\emptyset) = -20$ as well, with the optimal move being one to the right, using one red chip. The responder has a base score of -25; it can spend two blue chips to go right and upward, yielding a distance of 3 to the goal (i.e., 75 penalty points) and 5 remaining chips (50 points). One possible proposal for $P1$ is to offer a red and a grey chip for a blue chip, a green chip, and three yellow chips from the responder. In this case the proposer can get to the goal, and receives a gain of 60. Meanwhile, the responder can use this exchange to get one square away from the goal, but it uses all of its chips then. The gain from this exchange to the responder is zero.

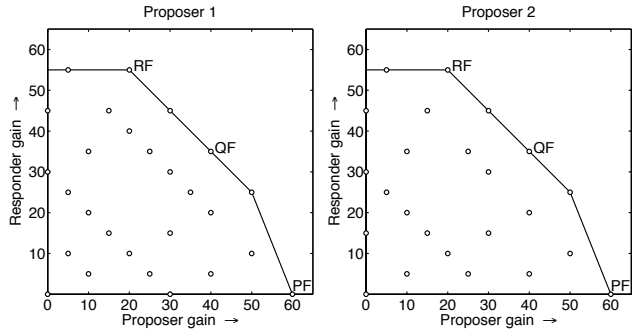
4. DEFINING METASTRATEGIES

Although the rules of the CT game are simple, it is not trivial to analyze. Both proposers need to reason about the tradeoff between

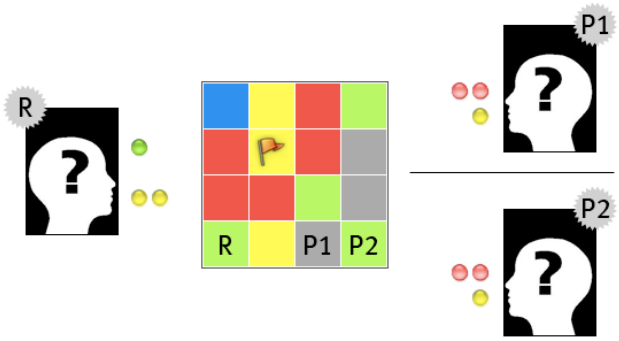
[2]Whenever we are not referring to one specific proposer, we will use the general notation 'P' when we imply '$P1$ and/or $P2$'.



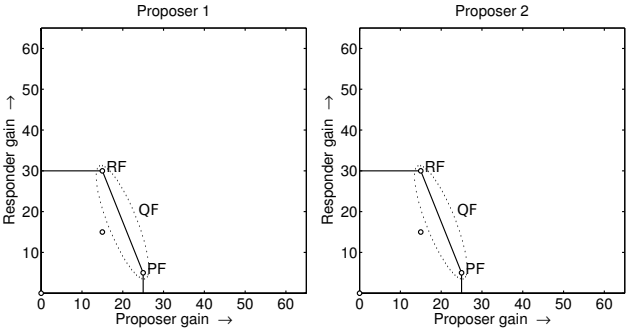
(a) First example Colored Trails game instance.



(b) Gain graph of the game instance presented in (a).



(c) Second example Colored Trails game instance.



(d) Gain graph of the game instance presented in (c).

Figure 1: Example Colored Trails game instances and gain graphs. In (a) and (c), the three players (R, $P1$ and $P2$) are shown, along with their chip sets. The two proposers cannot observe each others' chip sets. All players can see the board, on which their locations are indicated, as well as the goal state (a yellow flag). In (b) and (d), we show the gain graphs for both proposers. These graphs plot proposer gain versus responder gain for each possible proposal with non-zero benefit. The convex hull in this graph denotes the Pareto-front. The meta-strategies PF, RF and QF are located on this front, as indicated. In (b), QF is a pure meta-strategy; in (d), it is a mixed meta-strategy (of PF and RF), since there is no proposal on the convex hull between PF and RF.

making beneficial offers to the responder and offers that are beneficial for themselves, especially because they compete with each other for making the best offer to the responder. Moreover, the number of possible strategies is large. In the example instance presented in Figure 1(a), the number of unique proposals for $P1$ is 240, while $P2$ can choose from 144 unique proposals.[3] The responder can choose to accept or reject any of these offers, so the size of the strategy space for the responder is $240 \times 144 \times 2$. The size of the combined strategy space makes it difficult to analyze this game in a principled way. In this section we show how to reduce this large setting to smaller interactions in a way that preserves the strategic flavor of the original CT scenario.

The analysis presented in this section is not specifically tailored to (three-player) CT. Basically, any multi-agent one-shot negotiation setting may be analysed in the manner presented here; as with CT, agents may each have a large number of actions to choose from, making straightforward game-theoretic analysis very hard. We will discuss this after outlining the analysis.

4.1 Initial Assumptions

We first describe two assumptions we make about the various players in the game. We will relax the first assumption later.

Rational responder. The responder R has three possible actions, i.e., to accept the proposal of $P1$, to accept the proposal of $P2$, or to accept neither of them. For the responder, the game is thus similar to an Ultimatum game with proposer competition [9]. Initially, in our analysis, we assume that the responder plays according to a rational strategy. If both proposals do not provide it with a positive gain, it rejects both; if both proposals yield an equal gain, it accepts one of them with equal probability for both; if one proposal is strictly better, it accepts this proposal.

Semi-rational proposers. In order to select a strategy, i.e., a proposal to offer to the responder, proposers have to take into account the gain resulting from each proposal for themselves as well as the responder. For our analysis, we assume that proposer P limits the set of possible proposals to those that (1) lead to a non-negative personal gain, i.e., $G_P(s) \geq 0$, and (2) have a chance of being accepted by the responder, i.e., $G_R(s) \geq 0$. For example, in Figure 1(a), $P1$ ($P2$) has 79 (50) valid proposals given this limitation.

4.2 Analysis of Scenario

A CT game with only one proposer and one responder is highly similar to the canonical Ultimatum game. In this game, the optimal strategy s for the proposer P against a rational responder maximizes its gain while providing a non-negative gain for the responder (i.e., the optimal strategy is $\arg\max_s G_P(s)$). However, in the two-proposer setting we consider, proposers compete with each other, which means proposers have to take into account the gain of the responder. To facilitate analysis, we plot the gains $G_R(s)$ against $G_P(s)$ for each possible proposal s in a *gain graph*. Gain

[3]Two proposals are unique if they do not use the same chips.

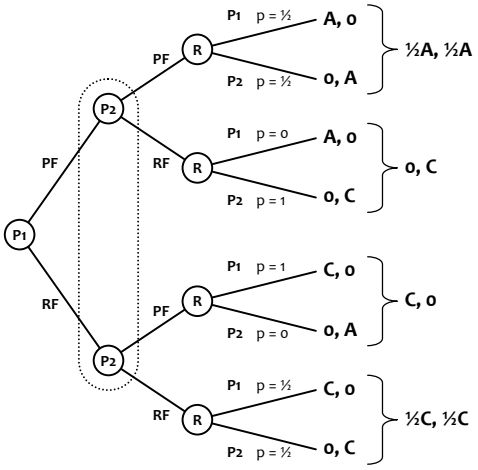


Figure 2: Extensive-form representation of the three-player negotiation variant of CT with two proposer metastrategies. The payoff for the rational responder R is not shown.

graphs for the two example games are given in Figures 1(b) and 1(d). In the interaction between a proposer and the responder, the Pareto-dominant proposals are located on the convex hull, as indicated in the figures.

Two proposer metastrategies. We note the following proposals located on the convex hull.

1. **Proposer focus (PF)**. PF is the strategy in which the proposer first maximizes its own gain, and then finds the maximum gain for the responder.

$$PF_P = \arg\max_{s'} G_R(s'), s' \in \arg\max_{s} G_P(s), s \in \mathcal{S}.$$

2. **Responder focus (RF)**. RF is the strategy in which the proposer first maximizes the responder's gain, and then finds the maximum gain for itself.

$$RF_P = \arg\max_{s'} G_P(s'), s' \in \arg\max_{s} G_R(s), s \in \mathcal{S}.$$

We call these proposals *metastrategies*, as their definition does not depend on the actual CT setting. The proposals corresponding to the metastrategies PF and RF for the example CT games appear in Figures 1(b) and (d). In the example instance of Figure 1(b), the strategy PF for P1 corresponds to the chip exchange we mentioned before (in which P1 offers one red chip and one gray chip in exchange for a blue, a green, and three yellow chips, leading to a gain, if accepted, of 60 for P1 and 0 for R), while RF corresponds to giving a blue chip, a red chip, and a gray chip in exchange for two green chips (leading to a gain of 20 for P1 and 55 for R here).

Interactions between metastrategies. Suppose that proposers play only the metastrategies PF and RF. We show an extensive-form representation of the resulting CT scenario in Figure 2 (for proposer 2). We do not list the payoff for the responder from playing its rational strategy. In the figure, the two decision nodes of $P2$ are grouped into into one information set, because the players make their proposals simultaneously. Once $P2$ has chosen, the static and rational strategy of the responder (which is indicated in the figure) leads to certain expected gains.[4] Here, A denotes the gain that a proposer receives when playing PF and being accepted; C denotes the gain for RF being accepted. Clearly, this extensive-form game can be represented in a 2x2 matrix game which omits the responder's strategy. The gain matrix of the symmetrical game between the two proposers is given below.

	PF	**RF**
PF	$\frac{1}{2}A, \frac{1}{2}A$	$0, C$
RF	$C, 0$	$\frac{1}{2}C, \frac{1}{2}C$

Since the game between the two proposers is a 2x2 matrix game, it is straightforward to analyze. The game depends on the relationship between A and C, as follows. For $A < 2C$, the game is a Prisoners' Dilemma, with one Nash Equilibrium at (RF, RF) and a Pareto-optimal outcome at (PF, PF). For $A \geq 2C$, we obtain a Stag Hunt game, with two Nash Equilibria; the RF-equilibrium (shorthand notation) is risk-dominant, while the PF-equilibrium is payoff-dominant. Both the Prisoners' Dilemma and the Stag Hunt game are well-known *social dilemmas* [13].

The strategic qualities of the original CT game are preserved in the 2x2 matrix game played between metastrategies. In the original CT game, the RF metastrategy corresponds to offering the best possible offer to the responder. RF is therefore also the risk-dominant proposal in the original game, because it guarantees a positive gain to the proposer. Even if both proposers play RF, the expected gain for each proposer will be positive. In contrast, the PF metastrategy is payoff-dominant but risky, because it provides a low (or even zero) gain to the responder. It will yield the most positive possible gain (payoff) for the proposer if the other proposer also plays PF, but will yield no gain at all otherwise. The proposers' dilemma in the original game (favoring themselves or the responder) is thus accurately reflected in the reduced 2x2 matrix game.

In the example CT instance shown in Figure 1(a) and (b), we find that the PF strategy yields a gain to the proposer of 60 and a gain of 0 to the responder if accepted, while the RF strategy yields a gain of 20 to the proposer and 55 to the responder. Hence, $A = 60$ and $C = 20$ here, and $A > 2C$; the game played between the two proposers is thus a Stag Hunt. In a similar manner, we can conclude that the CT game of Figure 1(c) and (d) is a Prisoners' Dilemma, because $A = 25$ and $C = 15$ yields $A < 2C$.

Introducing a third metastrategy. While we distinguish only two metastrategies thus far, a proposer's actual strategy s may be mixed, yielding (in theory) infinitely many possible (mixed) strategies s based on the two metastrategies.

We now demonstrate that a proposer can benefit from employing a metastrategy other than such a mixed strategy s. This is illustrated in Figure 3 (left and right). In the gain graph, all mixed strategies of PF and RF are located on the straight line connecting PF and RF. From the proposer's perspective, any mixed strategy s is strictly dominated by a strategy s^* for which $G_P(s^*) > G_P(s)$. This constraint is met by all points to the right of s in the plot. Given that the responder behaves rationally, we say that s^* strictly dominates s iff $G_R(s^*) > G_R(s)$. All points above s in the plot meet this constraint. Thus, strategies that lie in the white area of the graphs in Figure 3 strictly dominate the mixed strategy s.

[4]When calculating these expected gains, we assume that metastrategy pairs (e.g., PF_{P1} and PF_{P2}) yield the same gain for the responder, so the responder is indifferent to the two metastrategies. In other words, we assume the CT game instance is *symmetrical*. A CT game instance may generally not be (fully) symmetrical, unless we explicitly generate only symmetrical games. As we will discuss in our experimental section, using our set of criteria that lead to strategically interesting games, we find games that are symmetrical in expectation, even though symmetry is not a criterium.

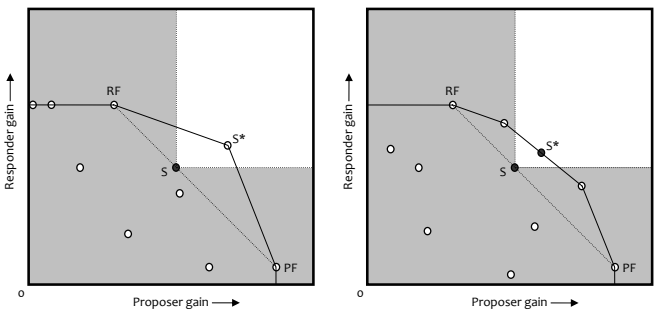


Figure 3: Pure or mixed strategies on the convex hull may strictly dominate a mixed strategy of PF and RF. In the example on the left, we find a pure strategy s^* on the convex hull that strictly dominates a mixed strategy s of PF and RF. On the right, a mixed strategy s^* strictly dominates s.

In some cases, the convex hull may lie on the straight line between PF and RF, as for instance in the second example game (Figure 1(d)); then, there is no strategy that strictly dominates s. In other cases however, as in Figures 3 and 1(b), the convex hull may be located above the line PF-RF. In these cases, we can *always* find a strategy s^* that strictly dominates s, except if s is a pure strategy itself, i.e., if s assigns a probability of 1 to a certain metastrategy. For instance, in Figure 3 (left), we find a pure strategy s^* on the convex hull that dominates s. In Figure 3 (right), a mixed strategy s^* on the convex hull dominates s. In Figure 1(b), both proposers have three pure strategies (and an infinite number of mixed strategies involving one or more of these pure strategies) that dominate a mixed strategy of PF and RF.

Thus, proposers indeed may benefit from employing additional metastrategies, since these additional metastrategies may dominate (mixed strategies of) the two metastrategies we already defined. We therefore introduce a third metastrategy, named QF (where Q is chosen simply because it is between P and R), which is to play the median proposal on the convex hull. Note that the median may be defined for any number of proposals on the convex hull; for an even number, we probabilistically select a proposal from the two median ones. Thus, proposals corresponding to the third metastrategy may again be found in any CT game.

Figures 1(b) and 1(d) show the mixed metastrategy QF for the two example CT instances. We see that the first instance has a pure QF metastrategy which dominates a mixed strategy of PF and RF. The gain graph shows that QF yields a proposer (responder) gain of 40 (35) here. The second instance has no strategies on the convex hull that dominate a mixed strategy of PF and RF; still, QF may be defined by choosing probabilistically from PF and RF. The (expected) gain for QF is then the average of the gains for PF and RF, i.e., 20 for the proposers and 17.5 for the responder.

Interactions between three metastrategies. With three metastrategies, the game between the two proposers becomes a 3x3 matrix game as follows.

	PF	**QF**	**RF**
PF	$\frac{1}{2}A, \frac{1}{2}A$	$0, B$	$0, C$
QF	$B, 0$	$\frac{1}{2}B, \frac{1}{2}B$	$0, C$
RF	$C, 0$	$C, 0$	$\frac{1}{2}C, \frac{1}{2}C$

Here, $A \geq B \geq C$. As with the two-strategy game, we can find different types of game depending on the relation between A, B, and C. It is easy to see that potential equilibria are located on the diagonal of the matrix. Moreover, as in the two-strategy game, (RF, RF) is an equilibrium. Depending on the values of A, B, and C, we may distinguish four different games. For all games in which $A < 2B < 4C$, (RF, RF) is the sole equilibrium. For $A \geq 2B \geq 4C$, all three diagonal strategies are equilibria. For $A < 2B$ and $B \geq 2C$, the equilibria are (RF, RF) and (QF, QF). For $A \geq 2B$ and $B < 2C$, we find equilibria at (RF, RF) and (PF, PF).

In the example of Figure 1(a), we find $B = 40$ ($A = 60$ and $C = 20$ still holds); thus, $A < 2B$ and $B = 2C$, meaning the 3x3 matrix game has two equilibria, i.e., the RF- and the QF-equilibrium. In Figure 1(c), we find $A = 25$, $B = 20$ and $C = 15$, so $A < 2B$ and $B < 2C$, yielding a single equilibrium at (RF, RF).

Adding social factors to the responder model. Thus far we have assumed the responder to be rational. Empirical evidence (in Ultimatum game settings) suggests that human responders are actually not fully rational [5]. One of the most well-known alternative models for Ultimatum-game responder behavior is *inequity aversion* [1]. The responder does not act directly on its gain $G_R(s)$, but instead on a utility function $U_R(s)$, which depends on its own gain, but also on how it compares to the gain of the proposer, $G_P(s)$. The original model distinguishes two components in the utility function, namely greed and compassion, both of which decrease the responder's utility in comparison to the actual gain. The greed component is generally far stronger (in humans); we do not consider the compassion component here. Translated to our settings, the responder's utility function may be then defined as follows.

$$U_R(s) = \begin{cases} G_R(s) & G_R(s) \geq G_P(s) \\ G_R(s) - \alpha_R \left(G_P(s) - G_R(s)\right) & \text{otherwise} \end{cases}$$

There is one parameter, α_R, which determines how strongly the responder dislikes a proposal which gives a proposer more gain than the responder.

To illustrate the effect of inequity aversion, we apply it to the gain graph of proposer 1 in the first example game (Figure 1(a) and (b)). The effect for $\alpha_R = 0.5$ is visualized in Figure 4. For proposals that give the proposer more gain than the responder (i.e., below the diagonal), the utility (perceived gain) for the responder is lower than the actual gain. As a result, some proposals that may be accepted by a rational responder are not accepted by an inequity-averse responder. As is visible from the figure, the convex hull changes, as does the location of the PF metastrategy. If the re-

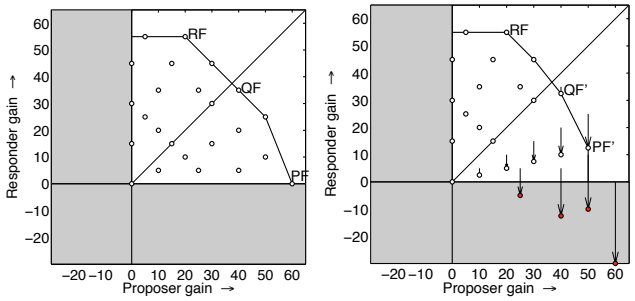


Figure 4: Effect of inequity aversion on the utility (perceived gain) of the responder: the convex hull and the metastrategies change. PF will no longer be accepted by the responder, which means proposers need to offer PF' instead.

sponder is inequity-averse and the proposers do not take this into account, they may coordinate (without communication) to offer the PF proposal, expecting that one of them will be accepted, while in reality, the responder will reject both proposals.

Instead of offering PF, proposers should offer PF', yielding a gain of 50 instead of 60 in the example. They are able to find PF' if they are aware of the inequity aversion present in the responder (i.e., the value of α_R), which implies they can calculate the modified gain graph. We deal with both unaware as well as aware proposers in our empirical evaluation.

4.3 Generalizing the analysis

In this section we show that our analysis may be generalized to more than three metastrategies and other variants of the CT game.

An arbitrary number of metastrategies. We may introduce additional metastrategies in a similar manner as the third one; e.g., we could have five metastrategies, corresponding to the minimal, first quartile, median, third quartile, and maximal proposal on the convex hull. Generally speaking, for n metastrategies, we obtain an nxn matrix game. The n diagonal strategies may each be equilibria or not, except the ever-present RF-equilibrium. Depending on the gains for each metastrategy pair, we may thus distinguish 2^{n-1} different possibilities for equilibria in the nxn game.

Generalizing to other CT variants. The analysis above is specifically performed on the three-player negotiation variant of CT. However, results generalize to other games within the CT framework, since chip exchanges are a vital part of the framework [7]. We will provide a few pointers here. A common different variant is a two-player negotiation game (i.e., one proposer and one responder), potentially with multiple phases and/or alternating roles [8]. The one-shot two-player game also allows us to construct the gain graph and identify the metastrategies. Since the dilemma (and the associated competition) between the proposers is missing, the single proposer may get away with offering PF every time.[5]

More generally, the concept of the gain graph naturally extends to negotiation games with any number of proposers and responders. For instance, a game with three proposers and one responder leads to a three-player social dilemma between the proposers, which may be modeled for instance as a Public Goods game, and interactions similar to the Ultimatum game between each proposer and the responder. In a game with multiple responders we can still construct gain graphs between pairs of proposer(s) and responders, with an Ultimatum game with responder competition [6] taking place between these pairs. Analytical and experimental studies in the Ultimatum game clearly indicate that players benefit from an increased number of opponents in the opposite role (e.g., responders fare well with more proposers) [6, 9].

5. EMPIRICAL EVALUATION

In this section, we outline how strategically interesting instances of the CT game may be generated. We then discuss how players that perform actions according to the metastrategies may be heuristically implemented. Finally, we generate a large number of games, have our heuristic players play them, and evaluate the empirical payoff tables, which can be compared with analytical results.

5.1 Strategically interesting games

This section outlines three criteria that ensure strategically interesting games, i.e., games that require strategic thinking from their players, and thus facilitate researchers to study this strategic thinking. Basically, strategically interesting games are fair games that require negotiation and mutual dependence.

Baseline scores. The initial board state (positions and chip sets) should yield baseline scores that are comparable for all three players. We generate games that limit the difference in baseline scores to be less than a certain ϵ.

$$\max\left\{G_{P1}\left(\emptyset\right), G_{P2}\left(\emptyset\right), G_{R}\left(\emptyset\right)\right\} - \min\left\{G_{P1}\left(\emptyset\right), G_{P2}\left(\emptyset\right), G_{R}\left(\emptyset\right)\right\} < \epsilon$$

Negotiation requirement. No player should be able to reach the goal location on its own without engaging in a chip trade. We define $isSolution(P, C) = true$ iff player P can reach the goal given a chip set C. The initial chip set of a player P is given by $chips(P)$.

$$\neg isSolution(P1, chips(P1)) \wedge \neg isSolution(P2, chips(P2)) \wedge \neg isSolution(R, chips(R))$$

Mutual dependence. Due to the negotiation requirement, both proposers depend on a *subset* of the responder's chip set. In turn, the responder relies on a subset of either proposer 1 or proposer 2. A one-sided proposal (i.e. asking for all chips or dispensing of all chips) may not lead to a chip set allowing both the proposer and the responder to reach the goal.

$$\exists C_{P1}, C_R \in chips(P1) \cap chips(R)\ s.t.\ C_{P1} \cap C_R = \emptyset \wedge isSolution(P1, C_{P1}) \wedge isSolution(R, C_R)$$
$$\exists C_{P2}, C_R \in chips(P2) \cap chips(R)\ s.t.\ C_{P2} \cap C_R = \emptyset \wedge isSolution(P2, C_{P1}) \wedge isSolution(R, C_R)$$

We implement these three criteria by generating many pseudo-random games and checking them against the criteria, keeping only those games that match. In a similar manner, we may introduce additional criteria, such as symmetry (see the discussion following).

5.2 Experimental setup

For the empirical evaluation of the proposed metastrategies we generate a database of $10K$ games that adhere to the criteria listed above (we chose $\epsilon = 20$). Below, we discuss how the metastrategies are implemented in heuristic players and how empirical payoffs are computed from games played between these players.

Heuristic players. We implemented three heuristic players, each following one of the three metastrategies, i.e. PF, QF and RF. All three heuristic players start by enumerating all possible chip exchange proposals. Proposals that yield negative gains for either the proposer or the responder are neglected. Heuristic players following metastrategies PF and RF are a straightforward implementation of the definitions given earlier. Metastrategy QF requires to compute the PF and RF strategy points in the gain graph, as well as the convex hull connecting both.[6] The median proposal on the convex hull is then selected. For an even number, the heuristic player probabilistically selects a proposal from the two median ones.

Computing empirical payoffs. A single entry of the empirical payoff matrix is computed as follows. The row determines the metastrategy played by P1, while the column determines the metastrategy for P2. For each game in the database, chip exchanges proposed by the players are evaluated by the responder and if a proposal is excepted, chips are exchanged and scores evaluated. The resulting payoff is the difference between final and baseline scores (i.e. gain) averaged over all $10K$ games. This process leads to a full empirical payoff table for the game as a whole.

[5]Repeated games fall outside the scope of this paper.

[6]While any convex hull algorithm is adequate, our implementation is based on the time-efficient Graham scam algorithm.

5.3 Results

In this section, empirical payoff tables obtained by the metastrategies are presented and compared to the predicted payoff tables.

Two-strategy game. With two metastrategies PF and RF, we obtain an empirical payoff table as follows.

	PF	RF
PF	21.0, 20.6	2.9, 11.7
RF	11.8, 2.6	6.5, 6.2

The empirical payoffs yield a Stag Hunt game, with $A \approx 42$ and $C \approx 12$. When we compare the empirical payoff table to the analytical one, we notice two things.

First, the game is nearly, but not completely symmetrical. This may be explained by the relatively small size of the board, which leads to relatively large differences (i.e. possible disbalances) between the two proposers. On the small board we use, symmetry arises from repeated play. The game is guaranteed to be symmetrical in expectation, since proposers' positions are randomized.

Second, there are (small) positive values where we expected values of 0. In some instances, a certain proposer's PF proposal is preferred by the responder over the other proposer's RF proposal. Once again, this issue may be dealt with by using larger boards, which would reduce the probability that PF 'wins' from RF. However, larger boards are (even) more difficult for human players.

Three-strategy game. The empirical payoff table for the three-strategy game is given below. The values in the corners of the table are identical to those in the two-strategy game.

	PF	QF	RF
PF	21.0, 20.6	5.7, 24.3	2.9, 11.7
QF	24.5, 5.6	14.8, 14.2	6.0, 9.8
RF	11.8, 2.6	10.0, 5.7	6.5, 6.2

We see that $B \approx 25$. It is interesting to consider the interactions between the 'neighboring' metastrategies. Looking at the interaction between PF and QF, we find a Prisoners' Dilemma. The QF metastrategy is very strong against PF, giving proposers a strong incentive to defect. Between QF and RF, we find a Stag Hunt. The payoff table thus yields a game with two equilibria, namely the QF- and the RF-equilibrium.

Inequity aversion (unaware proposers). In our next experiment, we determine the effect of introducing social considerations (inequity aversion) in the responder's decision-making, without the proposers being aware of this. We provide the empirical payoff matrices for two reasonable values of α_R, restricting ourselves to the two-strategy game.

$\alpha_R = 0.5$

	PF	RF
PF	9.6, 9.9	1.2, 12.0
RF	12.0, 1.2	6.4, 6.3

$\alpha_R = 1.0$

	PF	RF
PF	5.3, 5.3	0.7, 11.9
RF	12.0, 0.7	6.5, 6.1

The second equilibrium (PF, PF) disappears, because proposers expect their PF proposal to be accepted more than it actually is. The game thus turns into a Prisoners' Dilemma with one Pareto-dominated equilibrium at (RF, RF). The higher the value of α_R, the stronger this effect.

Inequity aversion (aware proposers). We also investigate what happens if the proposers *do* know that the responder is inequity-averse. The payoff matrices for the same values of α_R are:

$\alpha_R = 0.5$

	PF	RF
PF	17.0, 17.2	3.4, 11.3
RF	11.3, 3.2	6.5, 6.1

$\alpha_R = 1.0$

	PF	RF
PF	15.3, 15.1	4.6, 10.4
RF	10.5, 4.4	6.4, 6.1

The second equilibrium is back again; proposers appropriately adjust their PF proposals to the expectations of the responder. The payoff for PF is (sensibly) lower against itself than in the original game with a rational responder. PF does increasingly well against RF, simply because PF is (slightly) more similar to RF when proposers take into account the responder's expectation.

6. DISCUSSION

The previous sections have shown how CT games can be decomposed into a set of multiple normal-form games that are characterized by social dilemmas (i.e., Prisoners' Dilemma, Stag Hunt and Ultimatum games), as visualized in Figure 5, using a number of metastrategies defined on th chip exchange proposals in the game.

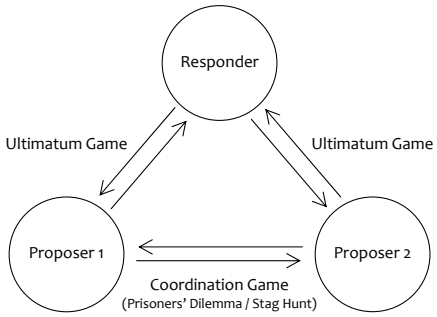


Figure 5: Decomposing the three-player negotiation variant of the Colored Trails Game.

We show that the metastrategy that favors the responder (RF) is always an equilibrium in the reduced normal form game, regardless of the number of metastrategies. This is because the responder has an advantage in the CT setting we consider, in that no player receives a gain if it does not accept an offer. An empirical analysis of a large set of game instances illustrates that, in expectation, two metastrategies yield two equilibria, reducing it to a Stag Hunt game. We may also find game instances that are Prisoners' Dilemmas, i.e., with only the equilibrium that favors the responder. Using three metastrategies in the same set of game instances also yields two equilibria in expectation, namely those two metastrategies that are most favorable for the responder (QF and RF).

Adding social factors to the responder allows this player to enforce a higher payoff—essentially, the proposers are driven to defection in the Stag Hunt or Prisoners' Dilemma game they play, because the responder is better at exploiting the proposer competition in the Ultimatum game component. This increased power for the responder may be countered by introducing multiple responders, as in an Ultimatum game with responder competition [6].

As noted, our analysis of a single game instance assumes that responders are indifferent between the gains from the two proposals resulting from any pair of metastrategies, i.e., for the metastrategies PF_{P1} and PF_{P2}, we have that $G_R(PF_{P1}) = G_R(PF_{P2})$ (and similar for QF and RF). If this condition does not hold, the responder will favor one of the metastrategy proposals over the other, which means the actual game instance does not reduce to a Stag Hunt or Prisoners' Dilemma. We observed that approximately 25% of the $10K$ games we generated (and that met our

three criteria) were actually games in which the responder is indifferent between metastrategy pairs. Of these 25%, approximately one-fifth are Prisoners' Dilemmas, and four-fifth are Stag Hunts. The remaining games (i.e., 75%) were not symmetrical, meaning that one proposer has a strategic advantage over the other proposer. Even though our symmetry assumption thus does not hold for a majority of generated game instances, our empirical results show that, even for games in which the assumption does not hold, the *expected* gains to proposers from playing metastrategies do in fact correspond to Stag Hunt and Prisoners' Dilemma games.

In case a certain experiment requires all games to be Stag Hunts or Prisoners' Dilemmas (i.e., not only in expectation), the assumption of responder indifference can be enforced during game generation. We note that, for the case in which responders are assumed to be rational, we do not need to make assumptions about the gains to proposers from pairs of metastrategies (a rational responder does not consider those gains), while for inequity-averse responders, the gains to proposers for every metastrategy pair must also be equal, i.e., $G_{P1}(PF_{P1}) = G_{P2}(PF_{P2})$ (and similar for QF and RF).

7. CONCLUSION AND FUTURE WORK

In this paper, we show how to reduce a large multi-agent task-setting, i.e., a game in the often-used Colored Trails (CT) framework, to a set of smaller, bilateral normal-form games, while still preserving most of the strategic charachteristics of the original setting. We show how to define representative heuristic metastrategies in the CT setting that make minimal assumptions about the other players. The games taking place between metastrategies are shown to correspond to Prisoners' Dilemma, Stag Hunt and Ultimatum games. We demonstrate that the metastrategies' analytical payoff tables, which we generated on the basis of assumptions that are not always met, nonetheless correspond to empirical payoff tables by sampling from thousands of CT game instances and showing that the outcome to players from using the metastrategies corresponds *on average* to the outcomes from the social dilemma games.

Although our analysis and examples are based on a particular CT scenario (a three-player take-it-or-leave-it game), they demonstrate the possibility of using metastrategies to reduce other CT games (e.g., games with a different number of players in each role, or even games that are further removed from the game under consideration here), and multi-agent interactions in general, to (social dilemma) normal-form games. More precisely, the techniques we present apply to general multi-agent interactions in which optimal actions can be computed, given that other players are using specified strategies.

We are currently extending our approach in two ways. First, we are developing metastrategies that consider other social factors that affect people's behavior in task settings, such as altruism and generosity, also from the perspective of the proposers. Second, we are considering more complex CT scenarios that include repeated negotiation, in which metastrategies will need to account for players' trust and reciprocity relationships.

8. REFERENCES

[1] E. Fehr and K. Schmidt. A Theory of Fairness, Competition and Cooperation. *Quart. J. of Economics*, 114:817–868, 1999.

[2] S. G. Ficici and A. Pfeffer. Modeling how Humans Reason about Others with Partial Information. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2008.

[3] Y. Gal, B. Grosz, S. Kraus, A. Pfeffer, and S. Shieber. Agent decision-making in open-mixed networks. *Artificial Intelligence*, 2010.

[4] Y. Gal and A. Pfeffer. Modeling reciprocity in human bilateral negotiation. In *National Conference on Artificial Intelligence (AAAI)*, 2007.

[5] H. Gintis. *Game theory evolving*. Princeton University Press Princeton:, 2000.

[6] B. Grosskopf. Reinforcement and directional learning in the ultimatum game with responder competition. *Experimental Economics*, 6(2):141–158, 2003.

[7] B. J. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The Influence of Social Dependencies on Decision-Making. Initial investigations with a new game. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2004.

[8] G. Haim, Y. Gal, S. Kraus, and Y. Blumberg. Learning Human Negotiation Behavior Across Cultures. In *HuCom10 - Second International Working Conference on Human Factors and Computational Models in Negotiation*, 2010.

[9] C. Hauert. Ultimatum game with proposer competition. GameLab experiments, consulted 10 August 2010.

[10] D. Hennes, K. Tuyls, M. Neerincx, and M. Rauterberg. Micro-scale social network analysis for ultra-long space flights. In *The IJCAI-09 Workshop on Artificial Intelligence in Space*, 2009.

[11] M. Kaisers, K. Tuyls, F. Thuijsman, and S. Parsons. Auction analysis by normal form game approximation. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 2, pages 447–450. IEEE, 2009.

[12] E. Kamar, Y. Gal, and B. Grosz. Modeling user perception of interaction opportunities for effective teamwork. In *Symposium on Social Intelligence and Networking, IEEE Conference on Social Computing*, 2009.

[13] D. Messick and M. Brewer. Solving social dilemmas: A review. *Review of personality and social psychology*, 4:11–44, 1983.

[14] S. Phelps, K. Cai, P. McBurney, J. Niu, S. Parsons, and E. Sklar. Auctions, evolution, and multi-agent learning. In *Proceedings of the Symposium on Adaptive Learning Agents and Multi-Agent Systems*, 2007.

[15] S. Phelps, M. Marcinkiewicz, and S. Parsons. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 705–712. Springer, 2001.

[16] M. Ponsen, K. Tuyls, M. Kaisers, and J. Ramon. An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, 1:39–45, 2009.

[17] P. Vytelingum, D. Cliff, and N. Jennings. Evolutionary stability of behavioural types in the continuous double auction. In *Agent-Mediated Electronic Commerce. Automated Negotiation and Strategy Design for Electronic Markets*, pages 103–117. Springer, 2007.

[18] W. Walsh, R. Das, G. Tesauro, and J. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, pages 109–118, 2002.

[19] W. Walsh, D. Parkes, and R. Das. Choosing samples to compute heuristic-strategy Nash equilibrium. *Agent-Mediated Electronic Commerce*, V:109–123, 2004.

Computing stable outcomes in hedonic games with voting-based deviations

Martin Gairing
Department of Computer Science
University of Liverpool
m.gairing@liverpool.ac.uk

Rahul Savani
Department of Computer Science
University of Liverpool
rahul.savani@liverpool.ac.uk

ABSTRACT

We study the computational complexity of finding stable outcomes in hedonic games, which are a class of coalition formation games. We restrict our attention to a nontrivial subclass of such games, which are guaranteed to possess stable outcomes, i.e., the set of symmetric additively-separable hedonic games. These games are specified by an undirected edge-weighted graph: nodes are players, an outcome of the game is a partition of the nodes into coalitions, and the utility of a node is the sum of incident edge weights in the same coalition. We consider several stability requirements defined in the literature. These are based on restricting feasible player deviations, for example, by giving existing coalition members veto power. We extend these restrictions by considering more general forms of preference aggregation for coalition members. In particular, we consider voting schemes to decide if coalition members will allow a player to enter or leave their coalition. For all of the stability requirements we consider, the existence of a stable outcome is guaranteed by a potential function argument, and local improvements will converge to a stable outcome. We provide an almost complete characterization of these games in terms of the tractability of computing such stable outcomes. Our findings comprise positive results in the form of polynomial-time algorithms, and negative (PLS-completeness) results. The negative results extend to more general hedonic games.

Keywords

Hedonic games, coalition formation, voting, local search, PLS-completeness.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Algorithms, Economics, Theory

Cite as: Computing stable outcomes in hedonic games with voting-based deviations, M. Gairing and R. Savani, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 559-566.

1. INTRODUCTION

Hedonic games were introduced in the economics literature as a model of coalition formation where each player cares only about those within the same coalition [12]. Such games can be used to model a variety of settings ranging from multi-agent coordination to group formation in social networks. This paper studies the computational complexity of finding stable outcomes in hedonic games. We consider and extend the stability requirements introduced in the work of Bogomolnaia and Jackson [6], which includes a detailed discussion of real-life situations in which purely hedonic models are reasonable.

An outcome is called *Nash-stable* if no player prefers to be in a different coalition. Here a deviation depends only on the preferences of the deviating player. Less stringent stability requirements restrict feasible deviations: a coalition may try to hold on to an attractive player or block the entry of an unattractive player. In [6], deviations are restricted by allowing members of a coalition to "veto" the entry or exit of a player. They introduce *individual stability*, where there is a veto for entering - a player can deviate to another coalition only if *everyone* in this coalition is happy to have her. They also introduce *contractual individual stability*, where, in addition to a veto for entering, coalition members have a veto to prevent a player from leaving the coalition - a player can deviate only if everyone in her coalition is happy for her to leave.

The case where every member of a coalition has a veto on allowing players to enter and/or leave the coalition can be seen as an extreme form of *voting*. This motivates the study of more general voting mechanisms for allowing players to enter and leave coalitions. In this paper, we consider general voting schemes, for example, where a player is allowed to join a coalition if the majority of existing members would like the player to join. We also consider other methods of *preference aggregation* for coalition members. For example, a player is allowed to join a coalition only if the aggregate utility (i.e., the sum of utilities) existing members have for the entrant is non-negative. These preference aggregation methods are also considered in the context of preventing a player from leaving a coalition. We study the computational complexity of finding stable outcomes under stability requirements with various restrictions on deviations.

The model.

In this paper, we study hedonic games with *symmetric additively-separable* utilities, which allow a succinct representation of the game as an *undirected edge-weighted graph*

$G = (V, E, w)$. For clarity of our voting definitions, we assume w.l.o.g. that $w_e \neq 0$ for all $e \in E$. Every node $i \in V$ represents a player. An outcome is a partition p of V into coalitions. Denote by $p(i)$ the coalition to which $i \in V$ belongs under p, and by $E(p(i))$ the set of edges $\{\{i,j\} \in E \mid j \in p(i)\}$.

The utility of $i \in V$ under p is the sum of edges to others in the same coalition, i.e., $\sum_{e \in E(p(i))} w(e)$. Each player wants to maximize her utility, so a player *wants to deviate* if there exists a (possibly empty) coalition c where

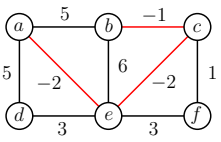
$$\sum_{e \in E(p(i))} w(e) < \sum_{\{\{i,j\} \in E \mid j \in c\}} w(\{i,j\}).$$

We consider different restrictions on player deviations. Those restrict when players are allowed to join and/or leave coalitions. A deviation of player i to coalition c is called

- *Nash feasible* if player i wants to deviate to c.
- *vote-in feasible* with threshold T_{in} if it is Nash feasible and either at least a T_{in} fraction of i's edges to c are positive or i has no edge to c.
- *vote-out feasible* with threshold T_{out} if it is Nash-feasible and either at least a T_{out} fraction of i's edges to $p(i)$ are negative or i has no edges within $p(i)$.
- *sum-in feasible* if it is Nash feasible and
$$\sum_{\{\{i,j\} \in E \mid j \in c\}} w(\{i,j\}) \geq 0.$$
- *sum-out feasible* if it is Nash feasible and
$$\sum_{e \in E(p(i))} w(e) \leq 0.$$

Outcomes where no corresponding feasible deviation is possible are called *Nash stable, vote-in stable, vote-out stable, sum-in stable*, and *sum-out stable*, respectively. Outcomes which are vote-in (resp. vote-out) stable with $T_{in} = 1$ (resp. $T_{out} = 1$) are also called *veto-in* (resp. *veto-out*) *stable*. Note that a veto-in stable outcome is an *individual stable* outcome i.e., any player can veto a player joining a coalition; an outcome that is veto-in and veto-out stable is a *contractual individual stable* outcome.

An example.



The above figure gives an example of a hedonic game. Consider the outcome $\{\{a,b,d\},\{c,e,f\}\}$. The utilities of the players a, b, c, d, e, f are $10, 5, -1, 5, 1, 4$, respectively. Players a, b, d, f have no Nash-feasible deviations, c has a Nash-feasible deviation to go alone and start a singleton coalition, and e has a Nash-feasible deviation to join the other coalition. The deviation of c is not veto-out feasible, since f prefers c to stay, however it is vote-out feasible for any $T_{out} \leq 0.5$. It is also sum-out feasible. The deviation of e is not veto-in feasible, but is vote-in feasible for any $T_{in} \leq 2/3$. Since there are no deviations that are both veto-in and veto-out feasible, this is a contractual individual stable outcome. The outcome $\{\{a,b,d\},\{c\},\{e,f\}\}$ is an individual stable outcome, and $\{\{a,b,d,e,f\},\{c\}\}$ is Nash stable.

Justification of the model.

With the goal of understanding how difficult it is for agents to *find* stable outcomes, we focus on a model in which they are *guaranteed* to exist. The computational complexity of a problem is measured in terms of the size of its input and therefore depends on the representation of the problem instance. For games, we desire that the size of the input is polynomial in the number of players, as this is the natural parameter with which to measure the size of the game. We consider only such *succinct representations*, since otherwise we can find solutions using trivial algorithms (enumeration of strategy profiles) that are polynomial in the input size. Our focus on additively-separable games is motivated by the hardness of even deciding the existence of stable outcomes and other solution concepts for more general (universal) succinct representations, such as hedonic nets [14]. A *non-symmetric* additively-separable game, which is represented by a edge-weighted *directed* graph, may not have a Nash-stable outcome [6, 4], and deciding existence is NP-complete. We study a more restrictive model where stable outcomes (for all of the stability requirements we consider) are guaranteed to exist, noting that our hardness results extend to all more general models where existence of stable outcomes is either guaranteed or promised, i.e., instances are restricted to those possessing stable outcomes.

In a *symmetric* additively-separable hedonic game, for each of the stability requirements we consider, a stable outcome *always exists* by a simple potential function argument: the potential function is the total happiness of an outcome, i.e., the sum of players' utilities. Unilateral player deviations improve the potential. So for all our considered stability requirements, local improvements will find a stable outcome, and all the problems we consider are in the complexity class PLS (polynomial local search) [20], which we introduce next.

Local search and the complexity class PLS.

Local search is one of few general and successful approaches to difficult combinatorial optimisation problems. A local search algorithm tries to find an improved solution in the *neighborhood* of the current solution. A solution is *locally optimal* if there is no better solution in its neighborhood. Johnson et al. [20] introduced the complexity class PLS (polynomial local search) to capture those local search problems for which a better neighboring solution can be found in polynomial time if one exists, and a local optimum can be verified in polynomial time.

They also introduced the notion of *PLS-reduction*. Suppose A and B are problems in PLS. Then A is PLS-reducible to B if there exist polynomial time computable functions f and g such that f maps instances of A to instances of B and g maps the local optima of B to local optima of A. A problem is PLS-complete if all problems in PLS are PLS-reducible to it. Prominent PLS-complete problems are those of finding a local max-cut in a graph (LocalMaxCut) [24], a stable solution in a Hopfield network [20], or a pure Nash equilibrium in a congestion game [16]. PLS captures the problem of finding pure Nash equilibria for many classes of

games where pure equilibria are guaranteed to exist.

On the one hand, finding a locally optimal solution is presumably easier than finding a global optimum; in fact, it is very unlikely that a PLS problem is NP-hard since this would imply NP=coNP [20]. On the other hand, a polynomial-time algorithm for a PLS-complete problem would immediately imply such an algorithm for all problems in PLS and thus solve a number of long open problems including the *simple stochastic game problem* [29]. PLS-complete problems are believed not to have polynomial-time algorithms.

Computational problems.

We define the search problems, NashStable, IS (individual stable), CIS (contractual individual stable), VoteIn, and VoteOut of finding a stable outcome for the respective stability requirement. We introduce VoteInOut as the search problem of finding an outcome which is vote-in and vote-out stable. All voting problems are parametrized by T_{in} and/or T_{out}. We also introduce sumCIS as the problem of finding an outcome which is sum-in and sum-out stable.

Symmetric additively-separable hedonic games are closely related to party affiliation games, which are also specified by an undirected edge-weighted graph. In a party affiliation game each player must choose between one of two "parties"; a player's happiness is the sum of her edges to nodes in the same party; in a stable outcome no player would prefer to be in the other party. The problem PartyAffiliation is to find a stable outcome in such a game. If such an instance has only negative edges then it is equivalent to the problem LocalMaxCut, which is to find a stable outcome of a local max-cut game. In party affiliation games there are at most two coalitions, while in hedonic games any number of coalitions is allowed. Thus, whereas PartyAffiliation for instances with only negative edges is PLS-complete [24], NashStable is trivial in this case, as the outcome where all players are in singleton coalitions is Nash-stable. Both problems are trivial when all edges are non-negative, in which case the grand coalition of all players is Nash-stable. Thus, interesting hedonic games contain both positive and negative edges.

The problem OneEnemyPartyAffiliation is to find a stable outcome of a party affiliation game where each node is incident to at most one negative edge. This problem was introduced in [17]. In this paper, we use a variant of this problem as a starting point for some of our reductions:

Definition 1. *Define the problem* OneEnemyPartyAffiliation* *as a restricted version of* OneEnemyPartyAffiliation *which is restricted to instances where no player is ever indifferent between the two coalitions.*

Gairing and Savani [17, Corollary 1] showed that OneEnemyPartyAffiliation* is PLS-complete.

Our results.

In this paper, we examine the complexity of computing stable outcomes in symmetric additively-separable hedonic games. In [17], it was shown that NashStable is PLS-complete while CIS is solvable in polynomial time. We make explicit two conditions, both met in the case of CIS, that (individually) guarantee that local improvements converge in polynomial time. The complexity of IS (i.e., of finding a veto-in stable outcome) was left open in [17]. Here we resolve that question, showing that IS is PLS-complete. Perhaps surprisingly, given the apparantly restrictive nature of the stability requirement, we show that sumCIS is PLS-complete, in contrast to CIS.

We also study the complexity of finding vote-in and vote-out stable outcomes. Using a different argument to the polynomial-time cases mentioned previously, we show that local improvements converge in polynomial time in the case of vote-in- and vote-out- stability with $T_{in}, T_{out} \geq 0.5$ and $T_{in} + T_{out} > 1$. We show that if we require vote-in-stability alone, we get a PLS-complete search problem. The problem of finding a vote-out stable outcome is conceptually different, and we can find a veto-out-stable outcome in polynomial time (whereas it is PLS-complete to find a veto-in-stable outcome). The technical difficulty in proving a hardness result for VoteOut is restricting the number of coalitions. Ultimately, we leave the complexity of VoteOut open, but do show that k-VoteOut, which is the problem of computing a vote-out stable outcome when at most k coalitions are allowed, is PLS-complete (Theorem 2). Our results are summarized in Figure 1, which gives an almost complete characterization of tractability.

Related work.

Hedonic coalition formation games were first considered by Dreze and Greenberg [12]. Greenberg [18] later surveyed coalition structures in game theory and economics. Based on [12], Bogomolnaia and Jackson [6] formulated different stability concepts in the context of hedonic games - see also the survey [26]. These stability concepts were our motivation to introduce definitions of stability based on voting and aggregation.

The general focus in the game theory community has been on characterizing the conditions for which stable outcomes exist. Burani and Zwicker [8] showed that additively-separable and symmetric preferences guarantee the existence of a Nash-stable outcome. They also showed that under certain different conditions on the preferences, the set of Nash-stable outcomes can be empty but the set of individually-stable partitions is always non-empty.

Cechlárová [9] surveys algorithmic problems related to stable outcomes. Ballester [4] showed that for hedonic games represented by an *individually rational list of coalitions*, the complexity of checking whether core-stable, Nash-stable or individual-stable outcomes exist is NP-complete, and that every hedonic game has a contractually-individually-stable solution. Recently, Sung and Dimitrov [27] showed that for additively-separable hedonic games checking whether a core-stable, strict-core-stable, Nash-stable or individually-stable outcome exists is NP-hard. For core-stable and strict-core-stable outcomes those NP-hardness results have been extended by Aziz et al. [2] to the case of symmetric player preferences. Brânzei and Larson [7] studied the tradeoff between stability and social welfare in additively-separable hedonic games. Elkind and Wooldridge [14] characterize the complexity of problems related to coalitional stability for hedonic games represented by hedonic nets, a succinct, rule-based representation based on marginal contribution nets (introduced by Ieong and Shoham [19]).

This work extends the model and results in Gairing and Savani [17]. The definition of party affiliation games we use appears in Balcan et al. [3]. Recent work on local max cut and party affiliation games has focused on approximation [5, 10]; see also [23]. For surveys on the computational

Leave \ Enter	1: no restr.	2: sum-in	3: veto-in	4: vote-in
A: no restr.	NashStable PLS-complete [17]	PLS-complete [17]	IS PLS-complete Theorem 4	VoteIn PLS-complete Theorem 1
B: sum-out	PLS-complete Theorem 5	sumCIS PLS-complete Theorem 5	P Proposition 1	?
C: veto-out	P Proposition 2	P Proposition 2	CIS P [17]	P Proposition 2
D: vote-out	VoteOut ? (see Theorem 2)	? (see Theorem 2)	P Proposition 1	VoteInOut P $(T_{in}, T_{out} > 0.5)$ Theorem 3

Figure 1: Table showing the computational complexity of the search problems for different entering and leaving deviation restrictions. Note that columns 1 and 2 are essentially equivalent, since if a player has a Nash-feasible deviation that results in a negative payoff, she also has a sum-in feasible (and hence also Nash-feasible) deviation, namely to form a singleton coalition.

complexity of local search, see [22, 1]. We use the PLS-completeness of LocalMaxCut which was shown in Schäffer and Yannakakis [24].

There is an extensive literature on weighted voting games, which are formally simple coalitional games. For such a game, a "solution" is typically a vector (or set of vectors) of payoffs for the players, rather than a coalition structure as in our setting; for recent work on computational problems associated with weighted voting games see [13, 15]. Deng and Papadimitriou [11] examined the computational complexity of computing solutions for coalitional games for a model similar to additively-separable hedonic games, where the game is given by an edge-weighted graph, and the value of a coalition of nodes is the sum of weights of edges in the corresponding subgraph. Here, we study the complexity of finding a stable set of coalitions.

2. COMPUTATIONAL COMPLEXITY OF FINDING STABLE OUTCOMES

In this section we study the complexity of computing stable outcomes under various stability requirements. We start by showing PLS-hardness for the case that a deviating player needs a T_{in} majority in the target coalition but there is no restriction on leaving coalitions.

Theorem 1. VoteIn *is PLS-complete for any voting threshold* $0 \leq T_{in} < 1$.

Proof. We reduce from OneEnemyPartyAffiliation* represented by an edge-weighted graph $G = (V, E, w)$. Let $\Delta(G)$ be the maximum degree of a node in G. Recall that no player is ever indifferent between the two coalitions.

First observe that the case $T_{in} > \frac{\Delta(G)-1}{\Delta(G)}$ is exactly the same as IS (for which we show hardness in Theorem 4), since in this case one negative edge is enough to veto a player joining a coalition. In the following we assume $T_{in} \leq \frac{\Delta(G)-1}{\Delta(G)}$.

We augment G as follows:

For every negative edge (a, b) in G we introduce $2\Delta(G)-2$ new nodes, called *followers*, and connect them with a and b as shown in the Figure 2. Both, a and b, get $\Delta(G) - 1$ followers and have a δ edge to each of them. Moreover, the followers have also an edge of weight ε to the other node.

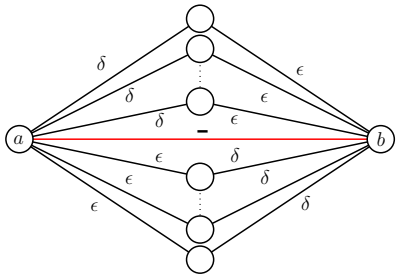


Figure 2: Gadget used for showing that VoteIn is PLS-complete. The gadget augments negative edges with followers that ensure that there is always a T_{in}-majority when a player enters a coalition.

Here $0 < \varepsilon < \delta$ and δ is small enough so that the player preferences of the original players (a and b) are still determined only by the original edges. In a stable outcome the followers will be in the same coalition as their "leader", i.e., the node to which they have a δ edge. The followers make sure that their is always a T_{in}-majority for entering a coalition. In other words, in a stable outcome of the VoteIn instance, the voting doesn't impose any restrictions.

To ensure that any stable outcome for the VoteIn instance has only two coalitions we further augment G by introducing two new players, called *supernodes*. Every player $i \in V$ has an edge of weight $W > \sum_{e \in E} |w_e|$ to each of the supernodes. The two supernodes are connected by an edge of weight $-M$, where $M > |V| \cdot W$. This enforces that the two supernodes are in a different coalition in any stable outcome. Moreover, by the choice of W, each player in V will be in a coalition with one of the supernodes. The fact that edges to supernodes have all the same weight directly implies that a stable outcome for the VoteIn instance is also a stable outcome for the OneEnemyPartyAffiliation* instance. The claim follows. □

In contrast to VoteIn, VoteOut is conceptually different. In VoteOut a coalition of two players connected by a positive edge is vote-out stable. This makes it hard to

restrict the number of coaltions. Doing this is probably the key for proving PLS-hardness also for VoteOut. For the following theorem we consider a version of VoteOut where the number of coalitions are restricted by the problem. Let k-VoteOut be the problem of computing a vote-out stable outcome when at most k coalitions are allowed. Observe that for any $k \geq 2$ such a vote-out stable outcome exists and that local improvements starting from any k-partition converge to such a stable outcome.

Theorem 2. *k-VoteOut is PLS-complete for any voting threshold $0 \leq T_{out} < 1$ and any $k \geq 2$.*

Proof. Our reduction is from OneEnemyPartyAffiliation, but we first reduce to the intermediate problem OneEnemyNashStable, which is a restricted version of NashStable where each player is only incident to at most one negative edge. Consider an instance of OneEnemyPartyAffiliation which is represented as an edge-weighted graph $G = (V, E, w)$. We augment G with two supernodes in exactly the same way as in Theorem 1. This ensures that any stable outcome of the OneEnemyNashStable instance uses only two coalitions and thus is also a stable outcome for the OneEnemyPartyAffiliation instance. Hence, OneEnemyNashStable is PLS-complete.

We now reduce from OneEnemyNashStable to k-VoteOut. Let G be the graph corresponding to an instance of OneEnemyNashStable. Let $\Delta(G)$ be the maximum degree of a node in G. We augment G as follows: We introduce $s \cdot k \cdot \Delta(G)$ new nodes where s is an integer satisfying $s \geq \frac{T_{out}}{1-T_{out}}$. Those nodes are organized in $s \cdot \Delta(G)$ complete graphs of k nodes each. All the edges in the complete graphs have weight $-M$ where M is sufficiently large ($M > |V| \cdot \Delta(G) \cdot \varepsilon$ will do). Moreover, we connect every original node $u \in V$ to every new node with an edge of weight $-\varepsilon$, where $\varepsilon > 0$.

By the choice of M and since at most k coalitions are allowed, in any stable solution there will be one node from each complete graph in each of the k coalitions. This shifts the utility of each player $i \in V$ with respect to each coalition by $-s \cdot \Delta(G) \cdot \varepsilon$. Moreover, every original node has at least $s \cdot \Delta(G)$ negative edges to each coalition. Since each node is incident to at most $\Delta(G)$ positive edges, it follows that the fraction of negative edges to each coalition is at least $\frac{s}{s+1} \geq T_{out}$. Thus, in every stable outcome all nodes $u \in V$ have a T_{out}-majority for leaving their coalition. This implies that in the corresponding outcome of the OneEnemyNashStable instance, no player can improve her utility by joining one of the k coalitions used in k-VoteOut. Moreover, in every stable outcome the utility of each node $u \in V$ with respect to the set of original nodes V is non-negative, since u has at most one negative incident edge in the OneEnemyNashStable instance and $k \geq 2$. It follows that a stable outcome for the k-VoteOut instance is also a stable outcome for the OneEnemyNashStable instance. The claim follows. □

It is an interesting open problem whether PLS-completeness also holds if the restriction on the number of allowed coalitions is dropped. Can we construct a gadget that imposes this restriction without restricting the problem a priori?

Since VoteIn and a restricted version of VoteOut are PLS-complete it's interesting to study the combination of both problems. What happens if we require vote-in stability *and* vote-out stability? With a mild assumption on the voting thresholds T_{in}, T_{out}, we establish:

Theorem 3. *For any instance of VoteInOut with voting thresholds $T_{in}, T_{out} \geq \frac{1}{2}$ and $T_{in} + T_{out} > 1$, local improvements converge in $\mathcal{O}(|E|)$ steps.*

Proof. For any outcome p define a potential function $\Phi(p) = \Phi^+(p) - \Phi^-(p)$, where $\Phi^+(p)$ (resp. $\Phi^-(p)$) is the number of positive (resp. negative) internal edges, i.e. edges not crossing coalition boundaries. Consider a local improvement of some player i from coalition $p(i)$ to $p'(i)$. Since $T_{out} \geq \frac{1}{2}$, player i has at least as many negative as positive edges to $p(i)$. Likewise since $T_{in} \geq \frac{1}{2}$, player i has at least as many positive as negative edges to $p'(i)$. So $\Phi(p)$ cannot decrease by a local improvement. Moreover, since $T_{in} + T_{out} > 1$, one of the threshold inequalities must be strict, which implies $\Phi(p') > \Phi(p)$. The claim follows since $-|E| \leq \Phi(p) \leq |E|$ and $\Phi(p)$ is integer. □

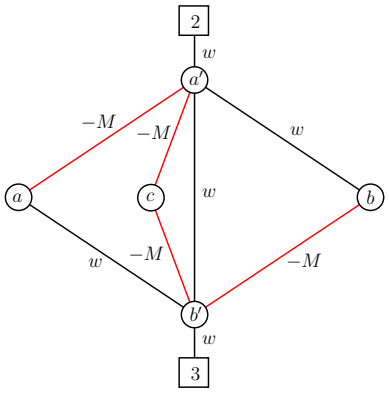
Without the assumption on the voting thresholds, the complexity of computing stable outcomes remains an interesting open problem. In particular the case $T_{in} = T_{out} = 1/2$ is very tantalizing.

We proceed by studying the complexity of finding stable outcomes if a single player in the target coalition can prevent (veto) a player from joining it. Observe, that the proof of Theorem 1 does not go through for this case. In [17] it was shown that a restricted version of IS (where in addition to normal IS deviations, two players connected by an negative edge are allowed to swap coalitions) is PLS-complete. Here, we show that allowing swaps is not necessary for PLS-hardness.

Theorem 4. IS *is PLS-complete.*

Proof. We start with an instance of OneEnemyPartyAffiliation*. The instance has the property that no player is ever indifferent between the two coalitions that make up stable outcomes. We add four supernodes which are connected by a complete graph of sufficiently large negative edges. This enforces that in any stable outcome the supernodes are in different coalitions, say 0, 1, 2, 3. The supernodes are used to restrict which coalition a node can be in in a stable outcome. This is achieved by having large positive edges of equal weight to the corresponding supernodes. All original nodes of the OneEnemyPartyAffiliation* instance are restricted to be 0 or 1.

We now show how to simulate a negative edge of OneEnemyPartyAffiliation* by an IS-gadget. To do so, we replace a negative edge (a, b) of weight $-w$ with the gadget in Figure 3. Nodes a and b are original nodes and restricted to $\{0, 1\}$, node a' is restricted to $\{0, 1, 2\}$, node b' is restricted to $\{0, 1, 3\}$, and node c is restricted to $\{2, 3\}$. As depicted in the gadget, nodes a' and b' have an additional offset to 2 and 3, respectively. Coalitions 2 and 3 are only used locally within the gadget. The pseudocode next to the gadget describes how the internal nodes of the gadget are biased. Here, checking whether a node can improve is w.r.t. her *original* neighborhood. We use "look at" and "bias" as defined in the following lemma and definition, which are analogous to those in [28, 21]. In particular, we check if a node can improve by looking at all nodes in her original neighborhood.



Bias internal nodes

if a can improve **then**
  bias c to 3
  bias a' to 2
else
  bias a' to $\{0,1\}$
  bias c to 2
end if
if b can improve **then**
  bias b' to 3
else
  bias b' to $\{0,1\}$
end if

Figure 3: Gadget to replace negative edges

LEMMA 1. *For any polynomial-time computable function $f : \{0,1\}^k \mapsto \{0,1,2,3\}^m$ one can construct a graph $G_f = (V_f, E_f, w)$ having the following properties: (i) there exist $s_1, \ldots, s_k, t_1, \ldots, t_m \in V_f$, (ii) all edges $e \in E_f$ are positive, (iii) $f(s_1, \ldots, s_k) = (t_1, \ldots, t_m)$ in any stable solution of the hedonic game defined by G_f.*

DEFINITION 2. *For a polynomial-time computable function $f : \{0,1\}^k \mapsto \{0,1,2,3\}^m$ we say that G_f as constructed in Lemma 1 is a graph that* looks *at $s_1, \ldots, s_k \in V_f$ and* biases *$t_1, \ldots, t_m \in V_f$ according to the function f.*

Recall that the instance of ONEENEMYPARTYAFFILIATION* has the property that no player is ever indifferent between the two coalitions that make up stable outcomes. By scaling edge weights we can implement the "look at" required to bias the internal nodes of the gadget without affecting their original preferences.

We say that node a is *locked* by the gadget if $a = 1$ and $a' = 0$ or $a = 0$ and $a' = 1$. Node b is said to be locked accordingly. The following two lemmas describe the operation of the gadget. Both lemmas should be read with the implicit clause: *If the internal nodes (a', b', c) are stable.* Let $\neg u$ denote the complement of u over $\{0,1\}$.

LEMMA 2. *If neither a nor b can improve then a and b are locked by the gadget.*

LEMMA 3. *If a or b (or both) can improve then one improving node is not locked while the other node is locked by the gadget. Moreover, if a (resp. b) is not locked by the gadget then $b' = \neg b$ (resp. $a' = \neg a$).*

To complete the proof we show that a stable outcome of the IS instance is also a stable outcome for the ONEENEMYPARTYAFFILIATION* instance. Suppose the contrary. Then there must exist an original node which is stable for IS but not for ONEENEMYPARTYAFFILIATION*. Clearly such a node must be the node a or b for some gadget. So either a or b (or both) can improve. But then by the first statement in Lemma 3 one of the improving nodes is unlocked, say a. Since a was only incident to one negative edge in the ONEENEMYPARTYAFFILIATION* instance, a cannot be locked by any other gadget. Moreover, by the second statement in Lemma 3, a is now connected in the gadget by a positive edge to the node b' and $b' = \neg b$. On the one hand, if $a = b$ then the original edge (a, b) contributes $-w$ to a's utility while now a receives 0 from the edge (a, b'). On the other hand, if $a \neq b$ then the corresponding utility contributions are 0 and w. So if a changes strategy then the difference in her utility w.r.t. b is the same in both problems, since we just shifted the utility of node a w.r.t. b by w. So a is also not stable for IS, a contradiction. This finishes the proof of Theorem 4. □

In IS a single player can veto against others joining her coalition but there is no restriction on leaving a coalition. The following proposition shows that adding certain leaving conditions yields polynomial-time convergence from the all-singleton partition.

PROPOSITION 1. *Any problem in column 3 of Figure 1 can be solved in polynomial time provided that the leaving condition requires that the leaving node has at least one negative edge within the coalition. In particular this hold for the problems in cells 3B, 3C, and 3D.*

PROOF. We use local improvements starting from the set of singleton coalitions. Then a player can make at most one improving step, since all edges in resulting non-singleton coalitions will be positive, and so no player can leave such a coalition. Hence we arrive at a stable outcome in at most $|V|$ improving steps. □

Interestingly, requiring veto-feasablity is already enough for polynomial-time convergence even if we have no restriction on the entering condition. This stands in contrast to Theorem 4.

PROPOSITION 2. *All problems in row C of Figure 1 can be solved in polynomial time by local improvements using at most $2|V|$ improving steps.*

PROOF. To get a running time of $2|V|$ (rather than $O(|V|^2)$) we restrict players from joining a non-empty coalition to which they have no positive edge. This ensures that whenever a player joins a non-empty coalition then this player (and all players to which she is connected by a positive edge in the coalition) will never move again. Moreover, a player can only start a new coalition once. It follows that each player can make at most two strategy changes. In total we have at most $2|V|$ local improvements. □

We close this paper with a result for SUMCIS. Even though deviations are very restricted here, it is PLS-complete to compute a stable outcome.

THEOREM 5. sumCIS *is PLS-complete.*

PROOF. We reduce from LOCALMAXCUT. Consider an arbitrary instance of LOCALMAXCUT with only integer edge weights. Recall that such an instance can be cast as an instance of PARTYAFFILIATION by negating the weights of the edges. Let $G = (V, E, w)$ represent the PARTYAFFILIATION instance. For each player $i \in V$ let σ_i be the total weight of edges incident to player i, i.e. $\sigma_i = \sum_{(i,j)\in E} w_{(i,j)}$. Observe that σ_i is a negative integer. We augment G by introducing two new players, called *supernodes*. Every player $i \in V$ has an edge of weight $\frac{-\sigma_i}{2} + \frac{1}{4}$ to each supernode. The two supernodes are connected by an edge of weight $-M$ where M is sufficiently large (i.e., $M > \sum_{i\in V}(\frac{-\sigma_i}{2} + \frac{1}{4})$). The resulting graph G' represents our sumCIS instance.

Consider a stable outcome of the sumCIS instance G'. By the choice of M the two supernodes will be in different coalitions. Now consider any player $i \in V$. If i is not in a coalition with one of the supernodes, then i's payoff is negative. On the other hand joining the coalition of one of the supernodes yields positive payoff, since $2(\frac{-\sigma_i}{2}+\frac{1}{4})+\sigma_i > 0$. Thus, each player $i \in V$ will be in a coalition with one of the supernodes. So our outcome partitions V into two partitions, say V_1, V_2.

It remains to show that any stable outcome for the sumCIS instance is also a local optimum for the PARTYAFFILIATION instance. Assume that the outcome of the sumCIS instance is stable but in the corresponding outcome of PARTYAFFILIATION instance there exists a player i which can improve by joining the other coalition. W.l.o.g. assume $i \in V_1$. Then, $\sum_{s\in V_1} w_{(i,s)} < \sum_{s\in V_2} w_{(i,s)}$. With $\sigma_i = \sum_{s\in V} w_{(i,s)}$ and since σ_i is integer, we get

$$\sum_{s\in V_1} w_{(i,s)} \leq \frac{\sigma_i}{2} - \frac{1}{2} < \frac{\sigma_i}{2} < \frac{\sigma_i}{2} + \frac{1}{2} \leq \sum_{s\in V_2} w_{(i,s)}.$$

It follows that in the sumCIS instance, player i's payoff is negative in her current coalition V_1 whereas joining V_2 would yield positive payoff. This contradicts our assumption that we are in a stable outcome of the sumCIS instance. The claim follows. □

3. CONCLUSIONS AND OPEN PROBLEMS

Our findings comprise both positive and negative results, some of which are somewhat surprising. There is an asymmetry between the case of vote-in and vote-out stability. We show that VOTEIN is PLS-complete for all voting thresholds, including $T_{in} = 1$. The case for $T_{in} = 1$, which corresponds to the search problem IS for finding a veto-in stable outcome, has to be treated separately from the case $T_{in} < 1$. In contrast, we show that the case of finding a veto-out stable outcome is polynomial-time solvable. This suggests that VOTEOUT is conceptually different from VOTEIN. Indeed, it seems difficult to restrict the coalitions in this case. We do show that k-VOTEOUT, where we restrict the outcome to have at most k coalitions, is PLS-complete for $0 \leq T_{out} < 1$, but we leave the complexity of VOTEOUT as an interesting open problem.

We show that even though requiring both sum-in and sum-out stability is apparantly quite restrictive, the resulting search problem sumCIS is PLS-complete.

In terms of positive results, we show that local improvements converge in polynomial time in the case of requiring *both* vote-in- and vote-out- stability with $T_{in}, T_{out} \geq 0.5$ and $T_{in} + T_{out} > 1$. We leave open the interesting case of VOTEINOUT with voting thresholds that do not satisfy $T_{in}, T_{out} \geq \frac{1}{2}$ and $T_{in} + T_{out} > 1$. We also leave open the case of finding an outcome that is vote-in and sum-out stable.

References

[1] E. H. L. Aarts and J. K. Lenstra. *Local Search in Combinatorial Optimization.* Wiley-Interscience, 1997.

[2] H. Aziz, F. Brandt, H. G. Seedig. Stable partitions in additively separable hedonic games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

[3] M.-F. Balcan, A. Blum, and Y. Mansour. Improved equilibria via public service advertising. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pp. 728–737, 2009.

[4] C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.

[5] A. Bhalgat, T. Chakraborty, and S. Khanna. Approximating pure Nash equilibrium in cut, party affiliation and satisfiability games. In *ACM Conference in Electronic Commerce (EC)*, pp. 132–146, 2010.

[6] A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[7] S. Brânzei and K. Larson. Coalitional affinity games and the stability gap. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 79–84, 2009.

[8] N. Burani and W. S. Zwicker. Coalition formation games with separable preferences. *Mathematical Social Sciences*, 45(1):27–52, 2003.

[9] K. Cechlárová. Stable partition problem. In *Encyclopedia of Algorithms.* Springer, 2008.

[10] G. Christodoulou, V. S. Mirrokni, and A. Sidiropoulos. Convergence and approximation in potential games. In *Symp. on Theoretical Aspects of Computer Science (STACS)*, pp. 349–360, 2006.

[11] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 12(2):257–266, 1994.

[12] J. H. Dreze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–1003, 1980.

[13] E. Elkind and D. Pasechnik. Computing the nucleolus of weighted voting games. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pp. 327–335, 2009.

[14] E. Elkind and M. Wooldridge. Hedonic coalition nets. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 417–424, 2009.

[15] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. On the computational complexity of weighted voting games. *Ann. Math. Artif. Intell.*, 56 (2):109–131, 2009.

[16] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The Complexity of Pure Nash Equilibria. In *ACM Symp. on Theory of Computing (STOC)*, pp. 604–612, 2004.

[17] M. Gairing and R. Savani. Computing stable outcomes in hedonic games. In *International Symp. on Algorithmic Game Theory (SAGT)*, pp. 174–185, 2010.

[18] J. Greenberg. Coalition structures. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume II. Elsevier, 1994.

[19] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *ACM Conference on Electronic Commerce (EC), 2005*, pp. 193–202, 2005.

[20] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37:79–100, 1988.

[21] B. Monien and T. Tscheuschner. On the power of nodes of degree four in the local max-cut problem. In *International Conference on Algorithms and Complexity (CIAC)*, pp. 264–275, 2010.

[22] B. Monien, D. Dumrauf, and T. Tscheuschner. Local search: Simple, successful, but sometimes sluggish. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 1–17 , 2010.

[23] J. B. Orlin, A. P. Punnen, and A. S. Schulz. Approximate local search in combinatorial optimization. *SIAM Journal on Computing*, 33(5):1201–1214, 2004.

[24] A. A. Schäffer and M. Yannakakis. Simple Local Search Problems that are Hard to Solve. *SIAM Journal of Computing*, 20(1):56–87, 1991.

[25] M. Sipser. *Introduction to the Theory of Computation*. Thomson, 2006.

[26] S. C. Sung and D. Dimitrov. On myopic stability concepts for hedonic games. *Theory and Decision*, 62, 2007.

[27] S. C. Sung and D. Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, 2010.

[28] T. Tscheuschner. The local max-cut problem is PLS-complete even on graphs with maximum degree five. *http://arxiv.org/abs/1004.5329*, 2010.

[29] M. Yannakakis. Equilibria, fixed points, and complexity classes. In *International Symp. on Theoretical Aspects of Computer Science (STACS)*, pp. 19–38, 2008.

APPENDIX

Proof of Lemma 1

PROOF. It is well known that for any polynomial computable function $f : \{0,1\}^k \mapsto \{0,1\}^m$ one can construct a circuit C with polynomial many gates that implements this function [25, Theorem 9.30]. Clearly, we can also restrict C to NOR gates with fan-in and fan-out at most 2. Organize the gates in levels according to their distance to C's output; output gates are at level 1.

We replace each gate g_i at level ℓ with the gadget in Figure 4. Nodes a, b are inputs and e is the output of the gate.

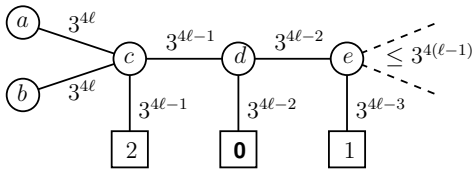


Figure 4: NOR gate

Nodes a, b and e are restricted (by supernodes) to $\{0,1\}$, node c is restricted to $\{1,2\}$, and node d is restricted to $\{0,2\}$. If a (or b) is an input of the circuit then we connect a to the corresponding input s-node by an edge of weight $3^{4\ell+1}$. If $\ell = 1$, i.e. g_i is an output gate, then we connect e to the corresponding output t-node with an edge of weight 1. Otherwise ($\ell > 1$), d is also the input to at most 2 lower level gates. The corresponding edges have weight at most $3^{4(\ell-1)}$. In any Nash-stable solution, $e = 1$ if and only if $a = b = 0$. In other words $e = NOR(a, b)$. The claim follows since our construction fulfils properties (i), (ii) and (iii). If a component of the function output has to be 2 or 3 we slightly adjust the corresponding output NOR gate. □

Proof of Lemma 2

PROOF. Since neither a nor b can improve, a' and b' are biased to $\{0,1\}$ and c is biased to 2. If $c = 2$ then the bias on a' assures $a' = \neg a$. So b' has an edge of weight w to both 0 and 1. Together with the bias this implies $b' = \neg b$. If $c = 3$ then the bias on b' assures $b' = \neg b$. So a' has an edge of weight w to both 0 and 1. Together with the bias this implies $a' = \neg a$. So in both cases $a' = \neg a$ and $b' = \neg b$. The claim follows. □

Proof of Lemma 3

PROOF. We consider three cases: (i) only a can improve, (ii) only b can improve, (iii) a and b can improve.

Case (i) (only a): Here c is biased to 3, a' is biased to 2, and b' is biased to $\{0,1\}$. First assume $c = 2$. This enforces $a' = \neg a$ which together with the bias implies $b' = \neg b$. But then the bias on c gives $c = 3$, a contradiction. Thus $c = 3$, which enforces $b' = \neg b$ and with the bias implies $a' = 2$. So a is not locked and b is locked.

Case (ii) (only b): Here c is biased to 2, a' is biased to $\{0,1\}$, and b' is biased to 3. First assume $c = 3$. This enforces $b' = \neg b$ which together with the bias implies $a' = \neg a$. But then the bias on c gives $c = 2$, a contradiction. Thus $c = 2$, which enforces $a' = \neg a$ and with the bias implies $b' = 3$. So a is locked and b is not locked.

Case (iii) (a and b): Here c is biased to 3, a' is biased to 2, and b' is biased to 3. If $c = 2$ then this enforces $a' = \neg a$, which together with the bias implies $b' = 3$. So in this case a is locked and b is not locked. If $c = 3$ then this enforces $b' = \neg b$, which together with the bias implies $a' = 2$. So in this case a is not locked and b is locked.

In every case both claims of the lemma are fulfilled. □

Empirical Evaluation of Ad Hoc Teamwork in the Pursuit Domain

Samuel Barrett
Dept. of Computer Science
The Univ. of Texas at Austin
Austin, TX 78712 USA
sbarrett@cs.utexas.edu

Peter Stone
Dept. of Computer Science
The Univ. of Texas at Austin
Austin, TX 78712 USA
pstone@cs.utexas.edu

Sarit Kraus
Dept. of Computer Science
Bar-Ilan University
Ramat Gan, 52900 Israel
sarit@cs.biu.ac.il

ABSTRACT

The concept of creating autonomous agents capable of exhibiting *ad hoc teamwork* was recently introduced as a challenge to the AI, and specifically to the multiagent systems community. An agent capable of ad hoc teamwork is one that can effectively cooperate with multiple potential teammates on a set of collaborative tasks. Previous research has investigated theoretically optimal ad hoc teamwork strategies in restrictive settings. This paper presents the first empirical study of ad hoc teamwork in a more open, complex teamwork domain. Specifically, we evaluate a range of effective algorithms for on-line behavior generation on the part of a single ad hoc team agent that must collaborate with a range of possible teammates in the pursuit domain.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]

General Terms

Algorithms, Experimentation

Keywords

Ad Hoc Teams, Agent Cooperation: Teamwork, coalition formation, coordination, Agent Reasoning: Planning (single and multiagent), Agent Cooperation: Implicit Cooperation

1. INTRODUCTION

Autonomous agents, both of the software and robotic varieties, are becoming increasingly common and accepted as a part of day to day life. More often than not, these agents are deployed in settings in which they are aware ahead of time of what other agents they will encounter. In multiagent team settings, the teammates are usually deployed at the same time and by the same developers or users.

However, as agents become more robust and therefore more relied upon, they are likely to be deployed for longer periods of time and in less controlled teamwork settings. When that happens, these agents will need to be prepared to cooperate with many different types of teammates. For example, in a software setting, an agent may need to create travel plans for a client by interacting with other agents that it has not encountered before.

Cite as: Empirical Evaluation of Ad Hoc Teamwork in the Pursuit Domain, Samuel Barrett, Peter Stone, and Sarit Kraus, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 567-574.

In a recent AAAI challenge paper, Stone et al. defined an *ad hoc team setting* as a problem in which team coordination strategies cannot be developed a priori [15]. They presented an evaluation framework for measuring the ad hoc teamwork capabilities of an agent and summarized previous theoretical results. Although they emphasized that the ad hoc teamwork challenge is "ultimately an empirical challenge," to the best of our knowledge, there have not yet been any empirical evaluations of strategies for ad hoc teamwork.

This paper fills that gap. Specifically, using the evaluation framework from the aforementioned challenge paper, we evaluate and compare strategies for ad hoc teamwork in the popular pursuit domain from the multiagent systems literature [1]. In this domain, four predators must collaborate to capture a prey. In the usual setting, strategies for a full team of predators are evaluated together. In contrast, we study the effectiveness of an *individual* ad hoc team agent's strategy when combined with various sets of teammates.

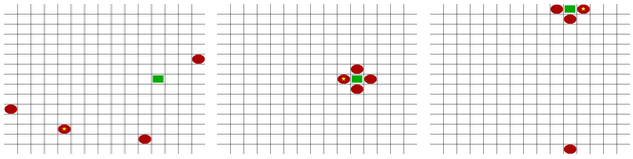
We begin with the simplest case in which the agent's teammates are homogeneous and behave deterministically, and the agent has a full model of their behavior (though it only learns of this behavior at the last minute: it still needs to determine its own behavior online). We then consider progressively more difficult scenarios in which the teammates are stochastic, heterogeneous, unknown but drawn from a distribution of types, and eventually completely unknown a priori. In so doing, we compare several different successful methods for generating teamwork behavior online. These methods range from optimal, but computationally complex, solutions to efficient, approximate sampling-based methods that incorporate Bayesian updates over the space of possible teammate behaviors.

The primary contribution of this paper is the initial empirical evaluation of ad hoc teamwork strategies. We present detailed analyses of extensive controlled empirical tests comparing generally applicable and effective algorithms for ad hoc teamwork.

The remainder of the paper is organized as follows. Section 2 describes our problem setting, including the testbed domain and evaluation framework. Section 3 introduces the space of teammates with which we test our ad hoc team agent, and Section 4 fully specifies the on-line behavior planning algorithms that we evaluate for the purposes of ad hoc teamwork. Section 5 presents the main contribution, namely detailed empirical results and analysis. Section 6 situates our contribution in the literature, and Section 7 concludes.

2. PROBLEM DESCRIPTION

The focus of this paper is an empirical evaluation of ad hoc teams. To this end, we present a well defined testing domain that requires the cooperation of a team, but still relies on each team member performing intelligently. Also, we specify a framework for evaluating and comparing the performance of ad hoc team agents.



(a) Random starting position (b) One capture position (c) Another capture position

Figure 1: Start and capture positions in the pursuit domain. The green rectangle is the prey, the red ovals are predators, and the red oval with the star is the ad hoc predator (the one under our control that is being evaluated).

2.1 Pursuit Domain

The pursuit domain was introduced by Benda et al. [1] and has been used frequently in the multiagent systems literature [18]. This problem is well suited for ad hoc team research as it requires cooperation between the agents; no agent can accomplish the task by itself regardless of its abilities. There are many variations of the pursuit domain, but they all involve a set of predators whose aim is to "capture" a prey, though the mechanics of the world and the definition of "capture" vary. A common formulation that we adopt is that the world is a toroidal grid and the predators must block all possible moves of the prey. For this work, we use a single prey and four predators, with only left, right, up, down, and no-op movements. We use a simple prey behavior that moves randomly.

Note that the world is a torus, so moving off one side of the world brings the agent back on the opposite side. This means that all four predators are required to capture the prey; it is not possible to trap the prey against the side of the board. Each agent can observe the positions of all other agents, but the agents are not capable of explicit communication. Agents start in random positions and select their actions simultaneously at each time step. Collisions are handled by ordering the agents, including the prey, randomly each time step, and performing moves in this order. If an agent's desired destination is occupied, the agent stays in its current location. Excluding collisions, all action effects are deterministic. Examples of the starting positions and capture positions can be found in Figure 1.

2.2 Ad Hoc Team Agent

For the purpose of comparing potential ad hoc team agents, we adopt the evaluation framework introduced by Stone et al. [15] and reproduced in Algorithm 1. According to this framework, the quality of an ad hoc team player depends on both the domain D and the set of possible agents A that the ad hoc agent will interact with. The algorithm compares agents a_0 and a_1 as potential ad hoc teammates of agents drawn from the set A collaborating on tasks drawn from domain D. Note that $s(B, d)$ is a scalar score resulting from the team B executing the problem d, where higher scores indicate better team performance and s_{min} is a minimum acceptable reward.

Throughout this paper, the domain D is the pursuit domain as described in Section 2.1. We consider each task $d \in D$ to be defined by the starting positions of the agents, the sequence of moves to be made by the prey, and the agent orderings for collisions. Therefore, if two different ad hoc agents perform the same actions on the same task, they will end with the same reward. The possible teammates comprising the set A are described next in Section 3.

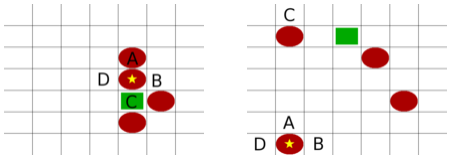
3. AGENT DESCRIPTIONS

In order to meaningfully test our proposed ad hoc teamwork algorithms, we implemented four different predator algorithms with varying and representative properties. The deterministic *greedy predator* mostly ignores its teammates' actions while the deterministic *teammate-aware predator* tries to move out of the way of its teammates, but it also assumes that they will move out of its way when needed. We expect these differences to require the ad hoc agent to adapt and reason about how its actions will interact with its teammates' actions. In addition to these two deterministic agents, we created two stochastic agents that select an action distribution at each time step. We expect it to be fairly trivial for the ad hoc agent to differentiate the deterministic agents, but harder to differentiate the stochastic agents. Finally, we tested our agent's ability to cooperate with a number of other agents for which it had no model.

Algorithm 1 Ad hoc agent evaluation

Evaluate(a_0 , a_1 , A, D):

- Initialize performance (reward) counters r_0 and r_1 for agents a_0 and a_1 respectively to $r_0 = r_1 = 0$.
- Repeat:
  - Sample a task d from D.
  - Randomly draw a subset of agents B, $|B| = 4$, from A such that $E[s(B, d)] \geq s_{min}$.
  - Randomly select one agent $b \in B$ to remove from the team to create the team B^-.
  - Increment r_0 by $s(\{a_0\} \cup B^-, d)$
  - Increment r_1 by $s(\{a_1\} \cup B^-, d)$
- If $r_0 > r_1$ then we conclude that a_0 is a better ad hoc team player than a_1 in domain D over the set of possible teammates A. Similarly, if $r_1 > r_0$ then a_1 is better.



(a) Configuration 1 (b) Configuration 2

Figure 2: World configurations that differentiate the predators' behaviors.

We will now introduce some notation to simplify the predator descriptions. Assume that a predator is at position (x, y) and is trying to move to a destination (x', y') on a world of size (w, h).

$$\Delta_x = (x' - x) \bmod w \qquad \Delta_y = (y' - y) \bmod h$$
$$\dim_{\min} = \operatorname{argmin}(\Delta_x, \Delta_y) \qquad \dim_{\max} = \operatorname{argmax}(\Delta_x, \Delta_y)$$
$$m_i = \operatorname*{argmin}_{\text{moves}} \Delta_i$$

Thus, m_i is the move that minimizes the difference to the destination for dimension i, and $\overline{m_i}$ is the move in the opposite direction. The stochastic agents use the softmax activation function, which assigns probabilities to a set of values, favoring the higher values. The temperature, τ, controls the amount of this bias, with values closer to 0 resulting in higher probabilities of the maximum value. If $v(i)$ is the value of option i, the probability of option a is

$$p(a) = \frac{\exp(v(a)/\tau)}{\sum_{i=1}^{n} \exp(v(i)/\tau)}$$

To clarify the predators' behaviors, we will show examples of their action selection on the cases shown in Figure 2, looking at the actions taken by the starred agent. The letters in the figure indicate the destination of the agent after taking one step. Note that none of the predators we created ever choose to stay still, so we do not label that action here.

3.1 Greedy Predator

The greedy predator selects the nearest unoccupied cell neighboring the prey, and tries to move towards it while avoiding immediate obstacles. It follows the following rules in order.

- If already neighboring the prey, try to move onto the prey so that if it moves, the predator will follow.
- Choose the nearest unoccupied cell neighboring the prey as the destination.
- Let $d = \dim_{\max}$. If m_d is not blocked, take it.
- Let $d = \dim_{\min}$. If m_d is not blocked, take it.
- Otherwise, move randomly.

For example, using the configurations shown in Figure 2 and taking actions as the starred agent, if the starred agent were a Greedy predator, it chooses the move taking it to cell C in configuration 1, and B in configuration 2. On average, a team of all Greedy predators captures the prey in 7.74 steps on a 5x5 world.

3.2 Teammate-aware Predator

The teammate-aware predator considers its teammates' distances from the prey when selecting its destination and uses A* path planning (an optimal heuristic search algorithm) [10] to avoid other agents, treating them as static obstacles. In contrast to the greedy predator, a teammate-aware predator that is already neighboring the prey may move towards another neighboring cell to give its spot to a farther away teammate. It is implemented as follows.

- Calculate the distance from each predator to each cell neighboring the prey.
- Order the predators based on worst shortest distance to a cell neighboring the prey.
- In order, the predators are assigned the unchosen destination that is closest to them (without communication), breaking ties by a mutually known ordering of the predators.
- If the predator is already at the destination, try to move onto the prey so that if it moves, the predator will follow.
- Otherwise, use A* path planning to select a path, treating other agents as static obstacles.

For the configurations shown in Figure 2, a Teammate-aware predator in the position of the starred predator chooses the move taking it to cell D in configuration 1, and C in configuration 2 (note that since the world is a torus, this is a single move). A team of Teammate-aware predators captures the prey in 7.41 steps on a 5x5 world.

3.3 Greedy Probabilistic Predator

The greedy probabilistic predator moves towards the nearest cell neighboring the prey, but does not always take a direct path there. The predator favors minimizing $\dim_{\max}$ and prefers $m_{\dim}$ over $\overline{m_{\dim}}$.

- If already neighboring the prey, try to move onto the prey so that if it moves, the predator will follow.
- Choose the nearest unoccupied cell neighboring the prey as the destination.
- Given a destination, choose a dimension, d, to minimize using the softmax function with temperature 0.5 using the distance as v.
- Choose either m_d or $\overline{m_d}$ using the softmax function with temperature -0.5, using the distance after the move as v, but penalizing moves that are currently blocked.

On configuration 1 from Figure 2, the predator is deterministic, choosing the action taking it to position C. On configuration 2, it selects a distribution of actions, specifically the moves taking it to cells A, B, C, and D with probabilities 0.000, 0.879, 0.119, and 0.002. On a 5x5 world, a team of Greedy Probabilistic predators captures the prey in 12.88 steps.

3.4 Probabilistic Destinations Predator

The probabilistic destinations predator attempts to tighten a circle around the prey. It favors destinations that are both nearer to the prey and to itself, but may choose farther destinations to prevent getting stuck on other predators and dealing with a moving prey.

- If already neighboring the prey, try to move onto the prey so that if it moves, the predator will follow.
- Select a desired distance from the prey using the softmax function with temperature -1 using the distance as v.
- Select a destination at the chosen distance using the softmax function with temperature -1 weighted by the distance of the destination to the predator's current position.
- Let $d = \dim_{\max}$, and select m_d.
- If the destination or the next position is occupied, repeat.

For the configurations in Figure 2, a Probabilistic Destinations predator would select the move ending in C in configuration 1. On configuration 2, it would select actions taking it to cells A, B, C, and D with probabilities 0.007, 0.596, 0.388, and 0.009. A team of predators following the Probabilistic Destinations behavior capture in 9.19 steps on a 5x5 world.

3.5 Student-created Predator

In some situations, an ad hoc team agent may be aware of the space of possible behaviors from which its teammates are drawn. However in other cases, it may not know anything about them. To fairly test the latter scenario, we incorporated into our testing a number of agents that we did not create. Specifically, we used a set of agents created by undergraduate and graduate Computer Science students for an assignment in a workshop on agents. These students were initially provided with a skeleton agent and then iteratively improved their agent.

As one might expect from a class, there was a wide variety in the quality of the agents that were submitted. In order to ensure a base level of competence, we only considered agents that were able to capture the prey within 15 steps on average on a 5x5 world (i.e. $s_{min} = 15$ in Algorithm 1). Out of the 41 agents submitted, 12 of the agents met this threshold.

Due to space constraints, we cannot fully describe all of the student agents used, but here we highlight some interesting cases. One student focused on avoiding collisions at cells neighboring the prey. Therefore, this student assigned the predators an arbitrary ordering and had each predator only consider blocking a specific direction chosen based on the assigned ordering. This strategy works if all the predators have mutually complementary assignments, but can create inefficiencies when the predators start far from their desired blocking directions.

The highest performing agent from the class performed better than any of our agents on the 5x5 world, capturing in only 4.05 steps on average. This agent considers all the cells neighboring the prey, and then considers all possible assignments of these destinations to the predators. For each possible assignment, it calculates the distance from each predator to its destination. Then, it chooses the assignment that minimizes the sum of these distances. Finally, each predator chooses the move that minimizes its distance to the selected destination. This agent performs quite well, although it does not seek to avoid collisions among the predators.

4. PLANNER DESCRIPTIONS

As is clear from the evaluation framework described in Section 2.2, the main thing that distinguishes one ad hoc team agent from another is its strategy for planning and selecting actions as a function of the current task d and current set of teammates B^-. In this section we describe the ad hoc teamwork planning algorithms that we test in this paper.

One might think that the most appropriate thing for an ad hoc team agent to do is to "fit into" the team by following the same behavior as its teammates. However, in some cases, it is possible for

the ad hoc team agent to improve on this or even solve for the optimal behavior, if the agent has a full model of its teammates' behaviors. Even without such a model, the ad hoc agent can approximate the optimal behavior. Indeed, in our tests, we found situations in which model-based planning even with an imperfect model outperforms the ad hoc agent mimicking its teammates' behaviors.

In some cases, the ad hoc team agent may "recognize" its teammates and be able to use its stored knowledge of their behaviors to plan its own actions. This situation is still an ad hoc team setting because the agent must generate its strategy on-line: it does not know in advance whom its teammates will be.

4.1 Value Iteration

When there is a fully known model of the environment and each agent, the ad hoc team agent can treat the domain as a Markov Decision Process (MDP) and can solve for the optimal behavior using Value Iteration (VI) [19]. Value iteration relies on dynamic programming to solve the optimal state-action values for all state-action pairs. VI initializes the state-action values arbitrarily, and then improves these estimates using an update version of the Bellman optimality equation:

$$Q(s,a) = \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \max_{a'} Q(s',a') \right]$$

where $Q(s,a)$ is the long term expected reward of taking action a from state s, $P^a_{ss'}$ is the probability of transitioning from state s to state s' after taking action a, $R^a_{ss'}$ is the corresponding reward, and γ is the discount factor. These updates are repeated iteratively until convergence. The state-action values calculated by VI are guaranteed to be correct.

However, the problem space is exponential in the size of the world, with a power proportional to the number of agents. The use of symmetries can reduce the size of this space, but in our tests, VI on a 5x5 world took approximately 12 hours on the [[removed for blind review]] computing cluster. Due to the exponential blowup of the state space, there are 100^5 states in a 10x10 world (without using symmetries) as opposed to 25^5 in a 5x5 world, so running VI on larger worlds was unfeasible.

4.2 Monte Carlo Tree Search

When the state space is large and only small sections of it are relevant to the agent, it can be advantageous to use a sample-based approach to approximating the values of actions, such as Monte Carlo Tree Search (MCTS). Specifically, we use the MCTS algorithm called Upper Confidence bounds for Trees (UCT) as a starting point for creating our algorithm [13].

MCTS does not require the complete model of the environment; it only needs a way of sampling the effects of selected actions. Furthermore, rather than treating all of the state-actions as equally likely, UCT focuses on calculating only the values for relevant state-actions. UCT does so by performing a number of playouts at each step, starting at the current state and sampling actions and the environment until the end of the episode. It then uses these playouts to estimate the values of the sampled state-action pairs. Also, it maintains a count of its visits to various state actions, and estimates the upper confidence bound of the values to balance exploration and exploitation. UCT has been shown to be effective in games with a high branching factor, such as Go [7], so it should be able to handle the branching factor caused by the number of agents.

We modify UCT to use eligibility traces and remove the depth index to help speed learning in the pursuit domain. The pseudocode of the algorithm can be seen in Algorithm 2, with s being the current state. Similar modifications were made by Silver et al. with good success in Go [14].

Algorithm 2 The Monte Carlo Tree Search algorithm used by our ad hoc agent.

```
function Select(s):
  for i = 1 to NumPlayouts do
    Search(s)
  return a = argmax_a Q(s,a)
function Search(s):
  a = bestAction(s)
  while s is not terminal do
    (s', r) = simulateAction(s, a)
    a' = bestAction(s')
    e(s,a) = 1
    δ = r + γQ(s',a') − Q(s,a)
    for all s*, a* do
      Q(s*,a*) = Q(s*,a*) + e(s*,a*) * δ/visits(s*,a*)
      e(s*,a*) = λe(s*,a*)
    s = s'; a = a';
```

4.3 Planning for uncertainty

Both of the planners described above assume that some kind of a model of the environment is known. However, it is likely that the ad hoc team agent has some uncertainty about the behavior of its teammates. One possibility is that the agent has a prior probability distribution over a set of possible behaviors, representing its belief of the likelihood of its teammates following this behavior. As the ad hoc agent gets more information about the agents from their actions, it should update its belief using Bayes theorem:

$$P(\text{model}|\text{actions}) = \frac{P(\text{actions}|\text{model}) * P(\text{model})}{P(\text{actions})}$$

If the agent has a complete model of each of the teammate types and a prior belief $P(\text{model})$, it can calculate $P(\text{actions}|\text{model})$. Finally, $P(\text{actions})$ can just be treated as a normalizing factor, to make the probabilities of the various models sum to 1.

Using this method, the MCTS-based agent can keep track of the probabilities of the different behaviors, and sample the environment accordingly. For Value Iteration, the exact solution requires recalculating the correct Q-values for each new set of probabilities, but this was not feasible for our tests. Therefore, we approximate the VI solution using a linear combination of the Q-values learned for each set of teammates:

$$Q(s,a) = p_1 * Q_1(s,a) + p_2 * Q_2(s,a) + \ldots + p_n * Q_n(s,a)$$

where $p_i = P(\text{model}_i|\text{actions})$ and $Q_i(s,a)$ are the Q-values calculated for model_i. Note that this is not guaranteed to be correct, but it works well in practice and gives us a baseline of how well an ad hoc agent can do. However, this approach still requires the ad hoc agent to know that the teammates are using one of several known behaviors.

4.4 Learning to model teammates

Sometimes the ad hoc agent will encounter teammates that do not come from its set of known behaviors, so it may need to learn a model of these teammates. To learn a model of the teammates, we used an implementation of the C4.5 algorithm for generating decision trees, provided in WEKA [9]. The decision tree was given the absolute x and y coordinates of each agent in the world, and attempted to predict the action that the agent would take in that world state. However, the ad hoc agent does not directly observe its teammates' actions; it only observes the results of the actions. For the training data, these actions were approximated by observing

the agents' movements. If the agent did not move, it may have chosen not to move or it collided, so the decision tree is given both possibilities, weighted by the probability that each occurred.

In this case, the ad hoc agent starts with no information about these agents and has only a single episode to learn about its teammates, so it must adapt over a short period of time. In this paper, we assume that all the teammates are running the same algorithm, so we use a single decision tree to learn about the behaviors of all the teammates. Thus the decision tree is given three additional observations of the agents' actions at each time step. The ad hoc agent continues to use its set of known models to plan as it builds the model, tracking the probabilities of these models and the learned model using the Bayesian updates as in Section 4.3. The idea is that the agent will use the known models for its initial planning until it encounters actions that these models would not predict, at which time it will increase its reliance on the learned model.

Due to the extreme paucity of data available to the learner, it would be difficult to learn a useful model over the course of a single episode. Surprisingly, our empirical results in Section 5.6 indicate that this form of model learning is beneficial.

5. RESULTS

In this section, we evaluate and thoroughly analyze the planning algorithms in Section 4 in a series of increasingly open-ended ad hoc team scenarios. These results constitute the main contribution of this paper.

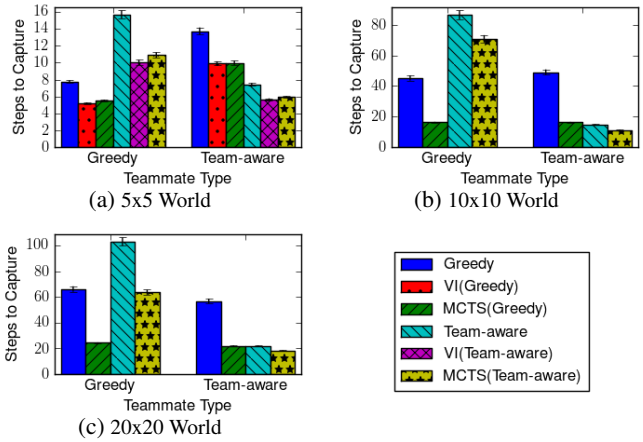
In all of our experiments, we use the evaluation framework discussed in Section 2.2. For each test, D is the set of all valid starting positions of the agents. For each episode, the starting position is randomly selected, but these positions are held constant across evaluations of the different agents. Similarly, other random factors such as the prey's action selection and collision tie-breakers are fixed with the starting positions. Therefore, if two ad hoc agents execute the same sequence of actions on the same problem, their results will be exactly the same. This approach controls for randomness in the environment and makes differences in the ad hoc agent's behavior the only cause for differences in the results.

For the first set of experiments, we assume that the behavior of the teammates is known at the start of the episode (though not before), and that this behavior is deterministic (Section 5.1). Even though the ad hoc team agent has a full model of its teammates, this scenario is still an ad hoc teamwork setting because there is no opportunity for the team to coordinate prior to starting the task: the agent must determine its strategy online. In the second set of experiments, we relax the constraint that the teammates' behavior is deterministic and investigate stochastic agents (Section 5.2). Next, we explore the performance of the ad hoc agents when the teammates' behavior is not exactly known, but is instead known to be drawn from a set of known behaviors (Section 5.3). Then, we mix the teammate types to see how the ad hoc agent could cope with unplanned teams and select the correct model from a large set of possible models (Section 5.4). In Section 5.5, we test against a set of agents that we did not create and that do not fit the models given to the ad hoc agent. Finally, we enable the agent to learn models on the fly to deal with these agents (Section 5.6).

For these tests, we compare against value iteration on the 5x5 worlds as it defines the optimal policy for the ad hoc agent, but we were unable to practically run VI on larger worlds as discussed in Section 4. In the graphs, we use VI(Greedy) and MCTS(Greedy) to indicate that the planning was performed treating the teammates as Greedy Predators, and we do likewise for VI(Teammate-aware), etc. Note that all of the predators on a team follow the same behavior for Sections 5.1–5.3, but not necessarily for Sections 5.4–5.6. In all cases, a lower number on the graphs is better, as it means that it took fewer steps to capture the prey. All results are averaged for 1,000 runs, but note that the ad hoc agent does not keep any knowledge between runs. All statistical tests are performed as paired Student-T tests to control for the randomness caused by the starting positions of the tests. The error bars shown are given as $\frac{\sigma}{n}$, where σ is the standard deviation of the lengths of the runs and n is the number of runs (1,000).

5.1 Deterministic known teammates

For our initial tests, we consider the simple case in which the ad hoc agent has an exact model of its teammates and its teammates are deterministic (either Greedy or Teammate-aware). These tests are designed to determine whether the ad hoc agent can do better than just mimicking its teammates, and how effective MCTS is at approximating the optimal behavior found by VI.



(a) 5x5 World (b) 10x10 World

(c) 20x20 World

Figure 3: Results with known deterministic teammates.

The results in Figure 3 show that the ad hoc agent can do much better than just copying the behavior of its teammates. Following the optimal behavior found by VI achieves capture in 5.19 and 5.92 steps respectively when cooperating with Greedy and Teammate-aware teammates as opposed to 7.74 and 7.41 steps when mimicking their behavior. These differences are statistically significant with $p < 0.001$. The improvements of planning over mimicking the teammates increase as the worlds get larger, although we use MCTS to approximate the optimal behavior for these worlds.

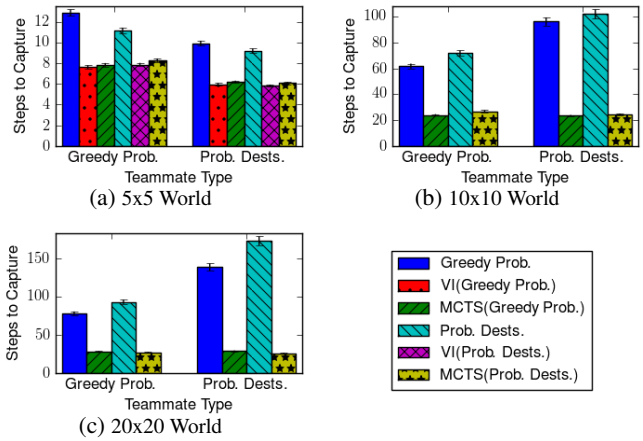
We verify that this use of MCTS is not much of a compromise, since it performs nearly as well as VI despite using much less computation time. In the 5x5 world, it takes 5.50 and 5.92 steps to capture with Greedy and Teammate-aware agents, as opposed to VI's 5.19 and 5.66 steps. The difference with the Greedy teammates is statistically significant ($p = 0.0244$), but the difference with the Teammate-aware teammates is not significant. The difference in performance could be lowered by using more playouts in the MCTS at the cost of more computation time. Given the close approximation to optimal that MCTS provides, the most important difference between the methods is the time it takes to plan. On the 5x5 world, MCTS episodes take on average less than a minute compared to VI's 12 hour computation (although VI only needs to run once, rather than for each episode). Furthermore, MCTS is an anytime algorithm, so it can be used to handle variable time constraints and can modify its plan online as the models change.

The results also show that having an incorrect model of your teammates can be costly. Even simply playing the Teammate-aware behavior when your teammates play Greedy hurts the team by a large amount. Intuitively, this seems odd as the ad hoc agent play-

ing smarter should help the team, but the Teammate-aware behavior relies on its teammates also moving out of its way, which will not happen with Greedy teammates. Planning as if your teammates are Greedy, when in fact they are Teammate-aware is costly when mimicking their behavior or when planning using VI or MCTS. On the 5x5 world, using the wrong model results in taking 13.75, 9.92, and 9.97 steps to capture for the mimic, VI, and MCTS cases, respectively, when the teammates are in fact Teammate-aware as opposed to 7.41, 5.66, and 5.92 steps when using the correct models.

5.2 Known stochastic teammates

We now consider the case where the ad hoc agent once again has an exact model of its teammates, but this time its teammates' behavior is stochastic (either Greedy Probabilistic or Probabilistic Destinations). The goal was to test whether the ad hoc agent could plan for agents choosing from several possible actions at any time step. VI uses the entire probability distribution of possible outcomes to update its values, while MCTS samples from this distribution to approximate the values.



(a) 5x5 World (b) 10x10 World (c) 20x20 World

Figure 4: Results with known stochastic teammates.

Figure 4 shows that the MCTS-based and VI-based agents are capable of planning with this uncertainty, and still significantly outperform mimicking the behavior of its teammates. Similar to the deterministic results, MCTS performs nearly as well as VI taking 7.84 and 6.07 steps versus 7.63 and 5.82 steps with Greedy Probabilistic and Probabilistic Destinations teammates respectively on the 5x5 world. These differences are significant, but MCTS still does a good job of approximating the optimal behavior.

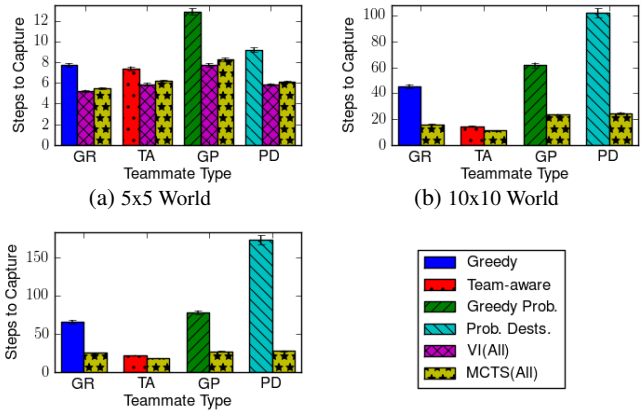
On the larger worlds, the performance of MCTS is much better than copying its teammates. For example, on the 20x20 world, the MCTS-based agent takes 24.00 steps to capture when cooperating with Greedy Probabilistic teammates compared to 78.48 steps when mimicking the teammates' behavior. Similarly, the MCTS-based agent takes 24.39 steps rather than 173.46 steps when paired with Probabilistic Destinations teammates.

Unlike the deterministic case, using an incorrect model for the teammates is not a large penalty with these agents. We believe that this is due to the overlap in the possible actions taken, and that the plans must be fairly robust to unexpected actions due to the stochasticity of the teammates.

5.3 Unknown stochastic teammates

Expanding the problem once again, all four predators used in the previous tests were used for this test, i.e. the Greedy, Teammate-aware, Greedy Probabilistic, and Probabilistic Destinations behaviors were used. Furthermore, the ad hoc agent did not know which of the four types of behavior its teammates were using. This setting gets us closer to the general ad hoc teamwork scenario, because it shows how well an ad hoc agent can do if it only knows that its teammates are drawn from a larger set A of possible teammates.

If it has a set of possible models for its teammates, ideally the ad hoc agent should be able to determine which model is correct and plan with that model appropriately. The VI and MCTS agents use the algorithm described in Section 4.3 to calculate the probabilities of each model. For both the MCTS and VI based ad hoc agents, we used a uniform prior over the teammate types, but assume that its teammates are homogeneous; i.e. there were no teams with some agents following the Greedy behavior and others following the Teammate-aware behavior. It is fairly trivial to differentiate the deterministic agents because as soon as they take one action that does not match the deterministic behavior, that incorrect model can be removed. However, the stochastic teammates are more difficult to differentiate, as there is significant overlap in the actions that are possible for them to take.



(a) 5x5 World (b) 10x10 World (c) 20x20 World

Figure 5: Results with unknown stochastic teammates. MCTS(All) means that the MCTS-based agent planned considering all homogeneous teams of the known predator models according to the current probabilities of the models.

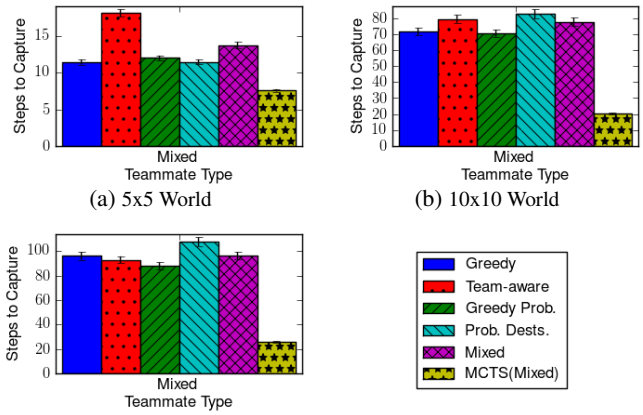
The results in Figure 5 show that both the VI and MCTS agents perform well despite this uncertainty and determine which model its teammates are following when given the set of possible models. These results are not quite as good as if the agent had the correct models to start with, but still perform quite well. For example, on the 20x20 world, if the ad hoc agent knew its teammates were using the Probabilistic Destinations behavior, it took 26.14 steps to capture, while if it needed to select the correct model, it took 27.83 steps. On the other hand, if it mimicked its teammates, the team would have taken 173.46 steps to capture the prey.

5.4 Mixed stochastic teammates

To this point, all of the teammates have used the same behavior, a fact which was known to the ad hoc team agent. In this section, we remove that restriction, thus significantly increasing the size of the possible set of agents A, specifically from 4 to $4^3 = 64$ possible teams. Doing so again moves us towards the general ad hoc team problem, where teammates may be following a variety of behaviors and may not be coordinating with one another.

Note that the teammate types were fixed for each problem, so this variance does not affect the different ad hoc agents' evaluation. As shown in Figure 6, following any of the fixed predator types achieved fairly poor results. However, the MCTS-based agent with the knowledge that any mix of the teammates was pos-

sible performs quite well. It learns which behaviors its teammates are likely to be following and adapts appropriately. For example, on the 20x20 world, the MCTS agent takes 20.19 steps to capture rather than 96.32 if it randomly chooses a model to mimic (labeled "Mixed" in the graphs). We were unable to run VI on this case, as it would have had to learn the optimal behavior for each combination of agents ($4^3 = 64$ possible teammate combinations).



(a) 5x5 World (b) 10x10 World (c) 20x20 World

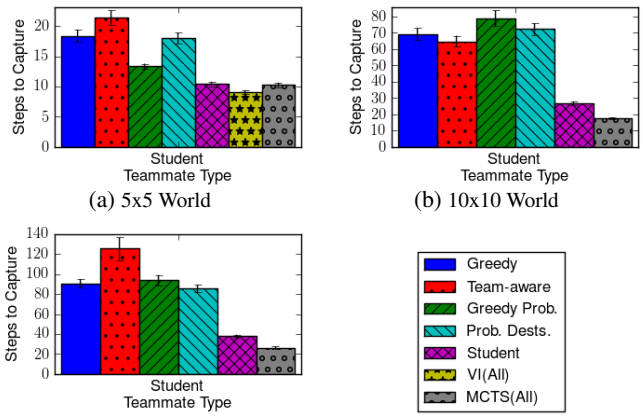
Figure 6: Results with mixed teams of stochastic teammates. The Mixed results are from the ad hoc agent randomly choosing a known predator model to mimic. MCTS(Mixed) refers to a MCTS-based agent that plans considering all heterogeneous teams of the known predator models, sampled according to the current probabilities of the models.

5.5 Unmodeled teammates

To this point, the ad hoc team agent has always had the benefit of a full model of all the teammates in A, even when it has not known a priori which types its teammates were. We now consider the case where there are agents in A for which the ad hoc team agent does not have a prior behavior model. Instead, we give the agent the same four models from before, and see how well it handles agents not following those models. To make sure we have not biased the creation of these agents, and that they truly are unknown, we used the student agents described in Section 3.5. Note that all the agents on each team used here are produced by the same student: we did not mix and match agents from different students. However, on some of the students' teams, not all of the agents use the same behavior. As before, the ad hoc agent does not store information between trials, so any learning happens during a single episode.

In this section, the ad hoc agent maintains the probabilities of the four known models as before and samples from this distribution. It does not actively consider the possibility that the teammates are unknown. Also, it assumes that all its teammates are using the same model, so it does not consider heterogeneous teams of the known models. Note that it is possible for the probability of all models to drop to 0 after a move if no known model would select that move. In this case, the agent just maintains the previous probabilities.

The results in Figure 7 show that the ad hoc agents do quite well despite the incorrect models. For example, on the 20x20 world, the MCTS agent captures in 26.47 steps rather than in 37.83 steps if it followed the student's behavior for the fourth agent. This is surprising because one would assume that planning using an incorrect model would perform worse than playing the behavior of the student's agent that the ad hoc agent replaced. Also, if the ad hoc agent follows any single one of its known models, it performs much worse than this baseline. So the ability to adapt and select the best known model at that time helps the ad hoc agent. This experiment shows that it is possible for an agent to cooperate with unknown teammates by using a set of known, representative models.



(a) 5x5 World (b) 10x10 World (c) 20x20 World

Figure 7: Results with student teams.

5.6 Learning to model teammates

In Section 5.5, the ad hoc agent tried to deal with the unknown agents as if they were one of the known models. Although this works fairly well in practice, the other agents may differ significantly from the known models, so it is desirable for the ad hoc agent to learn to model these agents as in Section 4.4. However, the ad hoc agent is given a very short time to learn, so we still use the set of known models and add an extra model that will be learned on the fly.

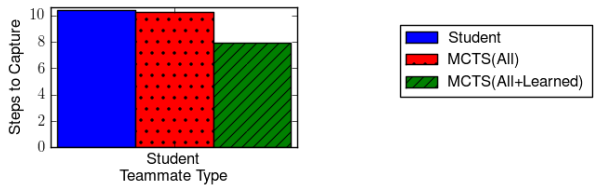
The results in Figure 8 show that learning a model of the teammates can improve performance over pretending that the teammates are following a known algorithm. Specifically, the MCTS-based agent using the learned model captured the prey in an average of 7.98 steps, as opposed to 10.26 when the agent only considered the known models. Furthermore, this is also an improvement over using the student's fourth agent, which captured in 10.40 steps on average.

This positive result is surprising due to the small number of training examples given to the agent. The episodes only lasted about nine steps on average, so the decision tree was being trained on only 27 training examples by the end of an episode on average. However, the learned model does not need to represent the entire action model of the teammates: only the states which occur. The visited number of states is likely to be small, and teammates are likely to act similarly in the visited states. Therefore, this model is much simpler to learn than a complete model. Also, we believe that a main advantage of learning the model was to prevent situations in which the agents became stuck due to collisions and incorrect predictions of the ad hoc agent.

The known models also provide a good starting point for the ad hoc agent, and it may not need to rely on the learned model too much. In our tests, for the final step of each episode, the ad hoc agent put 0.67 weight on the learned model on average, and that model was correct only 0.26 of the time. So the agent relied on the model, despite its inaccuracies. We theorize that the good performance of the system is due to the fact that it increases the options that the ad hoc agent considers, preventing it from being restricted to the actions of the known models. By itself, it is unlikely that this model would be sufficient for the agent, but when the agent uses both the learned and known models, it performs quite well.

6. RELATED WORK

The ad hoc team formulation and evaluation framework was proposed by Stone et al. [15], and there have been a few theoretical analyses of specific applications of ad hoc teams [16, 17]. Other



(a) 5x5 World

Figure 8: Results of learning models with student teams.

work in this area (though prior to the introduction of the ad hoc teamwork challenge) includes Brafman and Tennenholtz's work in which one agent teaches another while engaging in a repeated joint activity [2]. Knudson and Tumer have also investigated ad hoc teams, but in a significantly different framework [12]. Unlike our work, all of their agents adapt, and each agent is given a clear metric of its effect on the team's performance in the form of a difference objective. Furthermore, they learn over 2,000 episodes rather than our single episode. On the other hand, most prior work on coordinating teams of agents relies on explicit protocols for coordinating such as SharedPlans [8], STEAM [20], and GPGP [6]. Our work does not require these shared protocols, and does not even require the teammates to know of the ad hoc agent's existence.

The ad hoc team framework is similar to the existing opponent modeling problem. The ad hoc agent needs to model and understand its teammates, just from observing their actions, similar to opponent modeling. However, the ad hoc agent does not need to assume the worst case scenario; its teammates are not rational adversaries. In the area of opponent modeling, Conitzer and Sandholm created AWESOME, an algorithm that achieves convergence and rationality in repeated games [5]. Furthermore, Chakraborty and Stone have developed an algorithm for repeated games that handles arbitrary opponents safely and exploits memory bounded opponents [4]. This work makes weak assumptions about the adversaries, but it requires long learning times and assumes that all agents can calculate the same Nash equilibrium. Our work makes stronger assumptions about our teammates, but learns faster and makes no requirements about calculating Nash equilibria.

The pursuit domain is a well studied problem in multiagent research [18], but most research has focused on developing a coordinated team. However, some work has been done on learning to adapt to teammates. Chakraborty and Sen focus on having teammates teach novice predators, but they assume that the novices are trying to learn and share a known training protocol [3]. On the other hand, we assume that there is no shared protocol for training agents, and that there is only a single episode in which to adapt. Other work in the pursuit domain includes MAPS [21], which considers partially observable environments and more sophisticated prey behaviors, but require shared coordination algorithms. Alternatively, some approaches consider partial observability in continuous worlds [11]. However, these approaches focus on creating an entire team to solve the pursuit problem, rather than considering the case where some teammates are already following fixed behaviors.

7. CONCLUSIONS AND FUTURE WORK

This work presents the first empirical investigation of ad hoc teams, and establishes the pursuit domain as a useful domain for testing ad hoc teams. We show that an ad hoc team agent can do better than mimicking its teammates, and that efficient planning is possible using MCTS. Additionally, the ad hoc agent can differentiate its teammates on the fly when given a set of known starting models. We show that even if these models are incorrect or incomplete, as long as they are representative, they can be used to provide good performance. Finally, we show that it is possible to quickly learn models for previously unseen teammates, using known models until an accurate model is learned.

As the initial empirical investigation of ad hoc teams, this paper opens up several possible avenues for future research. In this work, we only considered teammates following fixed behaviors, and we assume that there is no explicit communication between teammates. Our ongoing research agenda includes extending to teammates that themselves learn, as well as considering the effects of a capability for partial communication among the agents. For example, the agents may then be able to communicate their desired destinations. Finally, all of the results in this paper were reported in the pursuit domain. Testing whether the same algorithms exhibit the same properties in a variety of other domains is also an important direction for future empirical research.

Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (IIS-0917122), ONR (N00014-09-1-0658), and the Federal Highway Administration (DTFH61-07-H-00030). Samuel Barrett is supported by a NDSEG fellowship. Sarit Kraus as also affiliated with UMIACS and her research is supported by NSF grant 0705587 and ISF Grant #1685.

8. REFERENCES

[1] M. Benda, V. Jagannathan, and R. Dodhiawala. On optimal cooperation of knowledge sources - an empirical investigation. Technical Report BCS–G2010–28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, Washington, July 1986.

[2] R. I. Brafman and M. Tennenholtz. On partially controlled multi-agent systems. *JAIR*, 4:477–507, 1996.

[3] D. Chakraborty and S. Sen. Teaching new teammates. In *AAMAS '06*, pages 691–693, 2006.

[4] D. Chakraborty and P. Stone. Convergence, targeted optimality and safety in multiagent learning. In *ICML '10*, June 2010.

[5] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Mach. Learn.*, 67, May 2007.

[6] K. S. Decker and V. R. Lesser. Designing a family of coordination algorithms. In *ICMAS '95*, pages 73–80, June 1995.

[7] S. Gelly and Y. Wang. Exploration exploitation in Go: UCT for Monte-Carlo Go. In *NIPS-2006*, December 2006.

[8] B. Grosz and S. Kraus. Collaborative plans for complex group actions. *Artificial Intelligence*, 86:269–368, 1996.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[10] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100 –107, July 1968.

[11] Y. Ishiwaka, T. Sato, and Y. Kakazu. An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning. *Robotics and Autonomous Systems*, 43(4):245 – 256, 2003.

[12] M. Knudson and K. Tumer. Robot coordination with ad-hoc team formation. In *AAMAS '10*, pages 1441–1442, 2010.

[13] L. Kocsis and C. Szepesvari. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer Berlin / Heidelberg, 2006.

[14] D. Silver, R. S. Sutton, and M. Müller. Sample-based learning and search with permanent and transient memories. In *ICML '08*, 2008.

[15] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI '10*, July 2010.

[16] P. Stone, G. A. Kaminka, and J. S. Rosenschein. Leading a best-response teammate in an ad hoc team. In *Agent-Mediated Electronic Commerce: Designing Trading Strategies and Mechanisms for Electronic Markets*. November 2010.

[17] P. Stone and S. Kraus. To teach or not to teach? Decision making under uncertainty in ad hoc teams. In *AAMAS '10*, May 2010.

[18] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, July 2000.

[19] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

[20] M. Tambe. Towards flexible teamwork. *JAIR*, 7:81–124, 1997.

[21] C. Undeger and F. Polat. Multi-agent real-time pursuit. *AAMAS '10*, 21:69–107, July 2010.

Decision Theoretic Behavior Composition

Nitin Yadav
School of Computer Science and IT
RMIT University
Melbourne, Australia
nitin.yadav@rmit.edu.au

Sebastian Sardina
School of Computer Science and IT
RMIT University
Melbourne, Australia
sebastian.sardina@rmit.edu.au

ABSTRACT

The behavior composition problem involves realizing a virtual target behavior (i.e., the desired module) by suitably coordinating the execution of a set of partially controllable available components (e.g., agents, devices, processes, etc.) running in a shared partially predictable environment. All existing approaches to such problem have been framed within *strict* uncertainty settings. In this work, we propose a framework for automatic behavior composition which allows the seamless integration of classical behavior composition with decision-theoretic reasoning. Specifically, we consider the problem of *maximizing the "expected realizability"* of the target behavior in settings where the uncertainty can be quantified. Unlike previous proposals, the approach developed here is able to (better) deal with instances that do not accept "exact" solutions, thus yielding a more practical account for real domains. Moreover, it is provably strictly more general than the classical composition framework. Besides formally defining the problem and what counts as a solution, we show how a decision-theoretic composition problem can be solved by reducing it to the problem of finding an optimal policy in a Markov decision process.

Categories and Subject Descriptors

I.12.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods.

General Terms

Theory, Verification, Algorithms.

Keywords

Behavior composition, decision theory, synthesis.

1. INTRODUCTION

In this work, we develop a decision theoretic account for behavior composition, that is, the problem of synthesizing a smart controller that is able to realize a virtual (i.e., non-available) target behavior module by suitably coordinating a set of available behaviors acting in a shared environment. Such problem has been extensively studied in the web-service composition literature (e.g., [1, 2]), where behaviors are deterministic and represent services, and more recently within the AI literature in more general settings (e.g., [5, 10, 11]), where nondeterministic behaviors may stand for the logic of various artifacts, such as agents or agents' high-level plans, physical devices, business processes, or software modules. Nonetheless, all approaches to behavior composition have assumed a setting of *strict uncertainty* [6], in that the incomplete information, for example, on the dynamics of the environment or on that of the available behaviors, cannot be quantified in any way. Hence, classical behavior composition problems may only accommodate *exact* solutions, that is, controllers that will guarantee the realization of the given target module no matter what. In many settings, however, while exact solutions may not exist, the ability to obtain a controller realizing the target module to the highest degree—the "optimal" controller—is desirable.

In order to better deal with non-solvable behavior composition instances, a framework in which different non-exact controllers can be compared is required. To that end, we propose an extension of the classical composition problem that goes beyond strict uncertainty, by accommodating ways of quantifying the different uncertainties in the model. Following the literature in behavior composition, we abstract the actual behaviors and environment as finite state *transition systems*. More precisely, each available module is represented as a nondeterministic transition system (to model partial controllability); the target behavior is represented as a deterministic transition system (to model full controllability); and the environment, which is fully accessible by all behaviors, is represented as a nondeterministic transition system (to model partial predictability).

As one can observe, in the above model, there are three sources of uncertainty stemming from the potential nondeterminism in both the environment and the available behaviors as well as in the potential different transitions in the target behavior. In the extended behavior composition framework to be developed here, all three uncertainties can be quantified. Note this is a reasonable assumption in many realistic settings, in which such information is readily available to the modeller. Consider a domain in which different bots are meant to maintain a garden, by performing various gardening activities such as cleaning, watering, and plucking flowers. Some bots may be equipped with buckets that may, nondeterministically, get filled after using it, and such nondeterminism can be quantified depending on various aspects of the domain (e.g., size of the bucket, average amount of dirt collected in a single action, etc.) Similarly, execution of actions in the garden environment—where the bots are meant to operate—can also be represented stochastically: a single clean operation may not always successfully clean the whole garden; the probability of a successful clean depends, for example, on the size of the garden and the season. More interestingly, given that the desired target behavior for maintaining the garden may involve more than one action from a given state, probabilities can be assigned to these depending on their expected

Cite as: Decision Theoretic Behavior Composition, Yadav, N. and Sardina, S., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 575-582.

frequency. For instance, in some state, the gardening target system is expected to request the plucking action 30% of the time only, most times it will just request watering the garden.

The contributions of this paper are threefold. First, building on [10, 5, 11], a decision theoretic framework for behavior composition is developed. In doing so, we define the notion of *optimal composition controllers* using the "expected realizability" of the target, as well as the notion of *exact* compositions, that is, controllers that will solve the composition problem robustly. Unlike previous frameworks for behavior composition, the proposed one is able to deal with problems that do not accept exact solutions. Second, we provide a translation of a decision theoretic behavior composition problem into a Markov decision process (MDP) [8, 6], and show that finding an optimal policy for such MDP amounts to finding an optimal composition. This problem reduction provides a readily available technique for solving the new composition framework using the established MDP paradigm. Third, we show that the decision theoretic framework developed here is a *strict* extension of the classical behavior composition frameworks in the literature.

2. THE PROBABILISTIC FRAMEWORK

Classical behavior composition problems are stated on an abstract framework based on a sort of finite state transition systems (see, e.g., [1, 5, 10, 11]). Specifically, the so-called (available) *system* includes a set of available behaviors representing those artifacts or devices at disposal that are meant to run within a shared environment. A *target* behavior then stands for such module that is desired but not directly available and is therefore meant to be "realized" by suitably composing the available behaviors in the system.

In a classical composition problem, incomplete information on any component is modeled by means of nondeterminism in the transition systems (in the available behaviors or in the environment) or different action transitions per state (in the target). However, all the work so far on the problem of behavior composition has assumed a setting of *strict uncertainty* [6] in that the space of possibilities—possible effects of actions, evolution of behaviors, and future action requests—is known, but the probabilities of these potential alternatives is not quantified.

In this section, we extend the framework used in [11, 10] to accommodate stochastic measures in the different components, thus yielding a framework for behavior composition under (non-strict) uncertainty. In particular, we use probabilities to model the uncertainty of the dynamics of the environment and of the available behaviors, as well as of the preferences on actions in the target module. Such probabilities are provided by a domain expert who is able to state how often a device happens to fail, an action brings about its expected effects, or certain requests arrive to the system.

Environment.

As standard in behavior composition, we assume to have a *shared fully observable environment*, which provides an abstract account of actions' preconditions and effects, and a mean of communication among modules. Since, in general, we have incomplete information about the actual preconditions and effects of actions, we shall use a stochastic model of the environment. Thus, given a state and an action to be executed in such state, different successor states may ensue with different probabilities. Formally, an *environment* is a tuple $\mathcal{E} = \langle \mathcal{A}, E, e_0, \mathcal{P}^{\mathcal{E}}_{next} \rangle$, where:

- $\mathcal{A}$ is a finite set of shared actions;
- E is the finite set of environment's states;
- $e_0 \in E$ is the initial state of the environment;
- $\mathcal{P}^{\mathcal{E}}_{next} : E \times \mathcal{A} \times E \mapsto [0, 1]$ is the probabilistic transition function among states: $\mathcal{P}^{\mathcal{E}}_{next}(e, a, e') = p$, or just $e \xrightarrow{a:p} e'$ in $\mathcal{E}$, states that action a when performed in state e leads the environment to a successor state e' with probability p. Furthermore, we require that for every $e \in E$ and $a \in \mathcal{A}$, $\sum_{e' \in E} \mathcal{P}^{\mathcal{E}}_{next}(e, a, e') \in \{0, 1\}$, that is, the action is not executable (the sum is 0) or all possible evolutions of the environment are accounted (the sum is 1).

EXAMPLE 1. A scenario wherein a garden is maintained by several bots is depicted in Figure 1. To keep the garden healthy one needs to regularly water the plants, pluck the ripe fruits and flowers, clean the garden by picking fallen leaves and removing dirt, and emptying the various waste bins. Whereas cleaning and emptying the bins is a regular activity, plucking and watering are done as required. The environment $\mathcal{E}$ models the states the garden can be in. The environment allows plucking and cleaning activities to be done in any order, and plants can be watered in any state. The pluck action results in the flowers and fruits been fully plucked 75% of the time (i.e., 25% of the time the garden still remains to be plucked), whereas the clean action results in the garden being totally cleaned 20% of the time (i.e., dirt still remains 80% of the time). A pluck action from the initial state (e_0) results in the garden been plucked but dirty (e_2) with a probability of .75, a subsequent clean action results in the garden being both plucked and clean (e_3), with a probability of .2. Similarly, a clean action from the initial state results in the garden being fully clean but not plucked (e_1) 20% of the time, and a subsequent pluck action causes the garden being cleaned and plucked (e_3) 75% of the time. For simplicity, we assume that emptying the bins always results in the environment evolving to its initial state. ■

Behaviors.

A behavior stands, essentially, for the logic of some available component (e.g., device, agent, plan, workflow), which provides, step by step, its user with a set of actions that can be performed. At each step, the user selects one action among those provided and executes it. Then, a new set of actions is provided, and so on. As behaviors are intended to interact with the environment (cf. above), their dynamics may depend on conditions in the environment. Formally, a *behavior* over an environment $\mathcal{E} = \langle \mathcal{A}, E, e_0, \mathcal{P}^{\mathcal{E}}_{next} \rangle$ is a tuple $\mathcal{B} = \langle B, b_0, \mathcal{P}^{\mathcal{B}}_{next} \rangle$, where:

- B is the finite set of behavior's states;
- $b_0 \in B$ is the initial state of the behavior;
- $\mathcal{P}^{\mathcal{B}}_{next} : B \times E \times \mathcal{A} \times B \mapsto [0, 1]$ is the probabilistic transition function of the behavior: $\mathcal{P}^{\mathcal{B}}_{next}(b, a, e, b') = p$, or $e \xrightarrow{a,e:p} e'$ in $\mathcal{B}$, denotes that action a executed in behavior state b when the environment is in state e will result in the behavior evolving to state b' with probability p. Since *all* potential transitions are accounted for in the model, we require that for every $b \in B$, $a \in \mathcal{A}$, and $e \in E$, $\sum_{b' \in B} \mathcal{P}^{\mathcal{B}}_{next}(b, a, e, b') \in \{0, 1\}$.

Behaviors are, in general, *nondeterministic*, that is, given a state and an action, there may be several transitions enabled by the environment. Hence, when choosing the action to execute next, one cannot be certain of the resulting state and of which actions will be available later on, since this depends on what particular transition happens to take place. In other words, nondeterministic behaviors are only *partially controllable*.

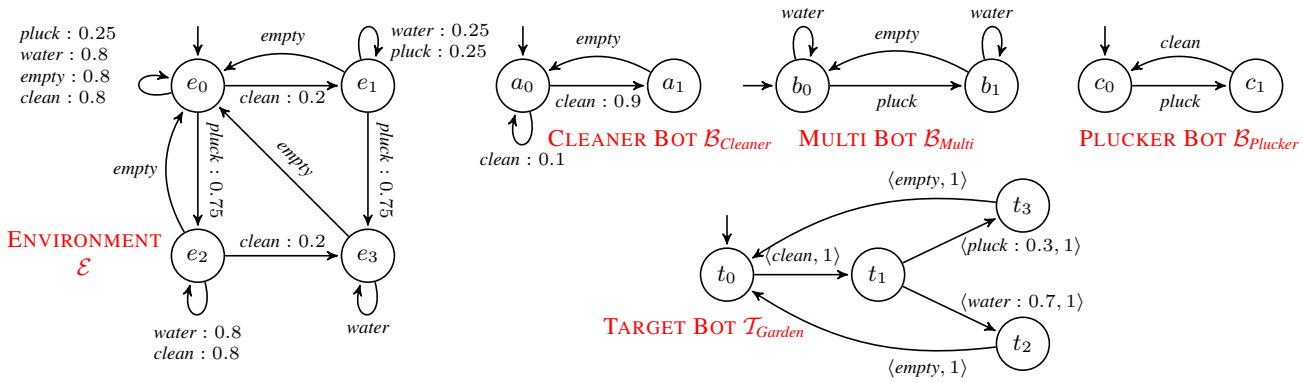


Figure 1: The garden bots system $\mathcal{S}_{Garden} = \langle \mathcal{B}_{Cleaner}, \mathcal{B}_{Multi}, \mathcal{B}_{Plucker}, \mathcal{E}\rangle$ and the target behavior $\mathcal{T}_{Garden}$.

EXAMPLE 2. In the gardening scenario, we assume there are three available garden bots; see Figure 1. The cleaner bot $\mathcal{B}_{Cleaner}$ cleans the garden by collecting the fallen leaves, dirt, waste, etc., into its own bucket. Most generally—90% of the time—its bucket gets filled up with a cleaning session, and the bot has to empty it to be able to start cleaning again. We assume the *empty* action involves emptying all garden bins as well as the bots' buckets. The plucker bot $\mathcal{B}_{Plucker}$ can pluck and clean the garden; since it is not equipped with a bucket, it plucks and collects from the ground directly. Finally, the multi-bot $\mathcal{B}_{Multi}$ has the capability to water the plants and pluck. It has a small bucket, and so it needs to empty it after every plucking session. ■

A behavior is deterministic if given a state and a legal action in that state, we always know exactly *the* next behavior state—the behavior is *fully controllable* through the selection of the next action to perform. Formally, a behavior $\mathcal{B} = \langle B, b_0, \mathcal{P}^{\mathcal{B}}_{next}\rangle$ over an environment $\mathcal{E} = \langle \mathcal{A}, E, e_0, \mathcal{P}^{\mathcal{E}}_{next}\rangle$ is *deterministic* iff for every $b, b' \in B$, $e \in E$, and $a \in \mathcal{A}$, it is the case that $\mathcal{P}^{\mathcal{B}}_{next}(b, e, a, b') \in \{0, 1\}$. In such case, the dynamics of the behavior can be represented using a transition relation $\delta_{\mathcal{B}} \subseteq B \times E \times A \times B$, where $\delta_{\mathcal{B}}(b, e, a, b')$ holds iff $\mathcal{P}^{\mathcal{B}}_{next}(b, e, a, b') = 1$.

Target behavior.

A target behavior is basically a *deterministic* behavior over $\mathcal{E}$ that represents the fully controllable desired behavior. A target behavior is virtual, in the sense that it does not exist in reality and, hence, is meant to be "realized" through the available behaviors.

Formally, a *target behavior* over an environment $\mathcal{E} = \langle \mathcal{A}, E, e_0, \mathcal{P}^{\mathcal{E}}_{next}\rangle$ is a tuple $\mathcal{T} = \langle T, t_0, \delta, R, \mathcal{P}_{req}\rangle$, where:

- T is the finite set of target's states;
- $t_0 \in T$ is the initial state of the target;
- $\delta \subseteq T \times E \times \mathcal{A} \times T$ is the target's deterministic transition relation: $\langle t, e, a, t'\rangle \in \delta$, or $t \xrightarrow{e:a} t'$ in $\mathcal{T}$, states that action a executed in the target state t, when the environment is in a state e, results in the target evolving to (unique) state t';
- $R : T \times A \mapsto \mathbb{R}^+$ is the reward function of the target: $R(t, a)$ denotes the reward obtained when the action a is successfully executed in target state t;
- $\mathcal{P}_{req} : T \times E \times \mathcal{A} \mapsto [0, 1]$ is the probabilistic action request function: $\mathcal{P}_{req}(t, e, a)$ denotes the probability of the target requesting the execution of action a when it is in state t and the environment is in state e. For consistency, we require that $\sum_{a \in \mathcal{A}} \mathcal{P}_{req}(t, e, a) \in \{0, 1\}$, for every $t \in T$, $e \in E$ (i.e., all possible requests are accounted for), and moreover, for all $a \in \mathcal{A}$, we have $\mathcal{P}_{req}(t, e, a) = 0$ whenever there is no state $t' \in T$ such that $\langle t, e, a, t'\rangle \in \delta$.

A *uniform-reward* target behavior is one where all actions have the same reward, that is, there exists $\alpha \in \mathbb{R}^+$ such that for all $a \in \mathcal{A}$ and $t \in T$, we have $R(t, a) = \alpha$.

This concludes the definition of the basic components for a decision-theoretic behavior composition problem. As the reader can easily note, this framework is essentially that of [5, 10, 11], except that *stochastic probabilistic transitions* are used instead of transition relations, a probability distribution over the potential action requests is used in the specification of the target, and a reward function is used in the target to state how "important" a particular request is. Note also that the probability function $\mathcal{P}_{req}$ in the target is very different to the ones used in the available behaviors and the environment. In the former, it denotes the probability of the target executing (i.e., requesting) an action from a given state, whereas in the latter the corresponding function simply denotes the stochastic evolutions of the entity.

EXAMPLE 3. The desired behavior required to maintain the garden in a particular season is not directly represented by any of the existing bots in the garden, and is modeled by the deterministic uniform-reward target bot $\mathcal{T}_{Garden}$ shown in Figure 1. Intuitively, the garden should always be cleaned first to remove any fallen leaves and dirt, followed by either plucking or watering the garden. Since flowers and fruits do not grow everyday, the plucking is required only 30% of the time; 70% of the time a request for watering the garden will be issued. Finally, the bins are to be emptied, and the whole process can repeat again. All requests are of equal value, namely, 1 unit (second component in each transition label). ■

Enacted system.

A *system* $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{E}\rangle$ is built form n, possibly nondeterministic, *available behaviors* $\mathcal{B}_i$, with $i \in \{1, \ldots n\}$, acting in a shared *environment* $\mathcal{E}$. Since, in the simplest case, one action can be executed at a given time, available behaviors in a system are meant to act concurrently in an interleaved fashion.

To refer to the behavior that emerges from the behaviors' joint (interleaved) executions, we use the notion of enacted system behavior, the synchronous product of the environment with the asynchronous product of all available be-

haviors. Let $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{E} \rangle$ be a system, where $\mathcal{E} = \langle \mathcal{A}, E, e_0, \mathcal{P}^{\mathcal{E}}_{next} \rangle$ and $\mathcal{B}_i = \langle B_i, b_{i0}, \mathcal{P}^{\mathcal{B}_i}_{next} \rangle$, for $i \in \{1, \ldots, n\}$. The *enacted system behavior* of $\mathcal{S}$ is the tuple $\mathcal{T}_{\mathcal{S}} = \langle S, \mathcal{A}, \{1, \ldots, n\}, s_0, \mathcal{P}_{\mathcal{S}} \rangle$, where:

- $S = B_1 \times \cdots \times B_n \times E$ is the finite set of $\mathcal{T}_{\mathcal{S}}$'s states; when $s = \langle b_1, \ldots, b_n, e \rangle$, we denote b_i by $beh_i(s)$, for $i \in \{1, \ldots, n\}$, and e by $env(s)$;
- $s_0 \in S$, with $env(s_0) = e_0$ and $beh_i(s_0) = b_{i0}$, for each $i \in \{1, \ldots, n\}$, is $\mathcal{T}_{\mathcal{S}}$'s initial state;
- $\mathcal{P}_{\mathcal{S}} : S \times \mathcal{A} \times \{1, \ldots, n\} \times S \mapsto [0, 1]$ is $\mathcal{T}_{\mathcal{S}}$'s probabilistic transition function, defined as follows:

$$\mathcal{P}_{\mathcal{S}}(s, a, k, s') = \mathcal{P}^{\mathcal{E}}_{next}(env(s), a, env(s')) \times \mathcal{P}^{\mathcal{B}_k}_{next}(beh_k(s), a, env(s), beh_k(s')),$$

  if $beh_i(s) = beh_i(s')$, for each $i \in \{1, \ldots, n\} \setminus \{k\}$; and $\mathcal{P}_{\mathcal{S}}(s, a, k, s') = 0$, otherwise.

To distinguish which behavior acts in each enacted transition, we label each stochastic transition in $\mathcal{T}_{\mathcal{S}}$ with the corresponding behavior index—all other behaviors remain still. We observe that the sources of nondeterminism in enacted behaviors stem from two sources, namely, the nondeterminism in the environment and the nondeterminism in the available behaviors.

So, informally, the decision-theoretic (DT) behavior composition task is stated as follows: Given a system $\mathcal{S}$ and a target behavior $\mathcal{T}$, find the "optimal" way of (partially) controlling the available behaviors in $\mathcal{S}$ in a step-by-step manner—by instructing them on which action to execute next and observing, afterwards, the outcome in both the behavior used as well as in the environment—so as to "best realize" a specific deterministic target behavior. In the next section, we make this problem definition precise.

3. DT-COMPOSITION

In order to bring about the desired virtual target behavior in an available system, we assume the existence of a (central) *controller* module that is able to control the available behaviors, in the sense that, at each step, it can observe all behaviors, instruct them to execute an action (within their capabilities), stop, and resume them. In classical behavior composition, one then looks for a controller that *guarantees* that the target will be implemented in the system *always*, that is, no matter how the target happens to requests actions within its logic or how the available behaviors and the environment happen to evolve with actions. Such controller is then deemed an (exact) solution to the problem. From a (generalized) planning perspective, the composition task can be seen as that of planning for a "maintenance" goal, namely, always maintain target realization.

When it comes to realizing a target module in a composition framework as the one described above, though, one should not just look for *exact* solutions, as in general there may be none. Instead, one shall look for *optimal* ways of maximizing the "expected realizability" of the target in the available system.

Controller.

Before formally defining the central module in charge of coordinating the available behaviors, we first need to define the technical notions of traces and histories of a system. A *trace* for a system $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{E} \rangle$ is a, possibly infinite, sequence of states from the enacted system behavior of the form $s^0 \xrightarrow{a^1,k^1} s^1 \xrightarrow{a^2,k^2} \cdots$ such that *(i)* $s^0 = s_0$; and *(ii)* $\mathcal{P}_{\mathcal{S}}(s^j, a^{j+1}, k^{j+1}, s^{j+1}) > 0$, for all $j \geq 0$. Intuitively, a trace represents a possible (legal) evolution of the (enacted) system, where k^j is the index of the behavior which has executed action a^j. A *history* is a just a *finite prefix* $h = s^0 \xrightarrow{a^1,k^1} \cdots \xrightarrow{a^\ell,k^\ell} s^\ell$ of a trace. We denote s^ℓ by $last(h)$, and the length ℓ of the history by $|h|$. The set of all histories for a given system will be denoted by $\mathcal{H}$.

So, formally, a *controller* for an available system $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{E} \rangle$ is a total function $C : \mathcal{H} \times \mathcal{A} \mapsto \{1, \ldots, n, u\}$ such that, given a system history $h \in \mathcal{H}$ and an action $a \in \mathcal{A}$ that ought to be performed, returns the index of the behavior to which the action a is to be delegated for execution. For technical convenience, a special value u ("undefined") may be returned, thus making C a total function which returns a value even for actions that no behavior can perform.[1]

Now, informally, a "dead-end" is reached in a history if the controller in use selects a behavior which is not capable of executing the delegated action. Then, given two controllers, one should prefer the one that reaches a dead-end with lower probability, or put it differently, the one that has the highest probability of honoring the target's requests. In particular, a controller that is guaranteed not to ever reach a dead-end will be an exact, and thus optimal, solution.

We say that a history is reachable by a controller, if starting from the initial state of the enacted system, the behavior executing the action at each state of the history is indeed the one selected by the controller. More formally, a history $h = s^0 \xrightarrow{a^1,k^1} \cdots \xrightarrow{a^\ell,k^\ell} s^\ell$ is *reachable* by a controller C (in a system $\mathcal{S}$) iff $k^i = C(s^0 \xrightarrow{a^1,k^1} \cdots \xrightarrow{a^{i-1},k^{i-1}} s^{i-1}, a^i)$, for each $i \in \{1, \ldots, \ell\}$. We denote with $\mathcal{H}^\ell_C$ the set of all reachable histories of length ℓ and $\mathcal{H}_C = \bigcup_{i \geq 0} \mathcal{H}^i_C$ the set of all histories reachable by C.

Value of a controller and compositions.

In order to evaluate and compare controllers, we define the value of a controller for a given target and system. Roughly speaking, a controller is "rewarded" for every action request from the target that it fulfills by a successful delegation to an available behavior. More specifically, at every point, a controller gets a reward that depends both on the frequency of such request and the value of (fulfilling) it.

From now on, let $\mathcal{T} = \langle T, t_0, \delta, \mathcal{P}_{req} \rangle$ be a target behavior to be realized in a system $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{E} \rangle$. Let C be a controller for system $\mathcal{S}$, and $\mathcal{T}_{\mathcal{S}}$ be the enacted system behavior as defined in the previous section. First, consider the case of evaluating the performance of a controller over a finite number of requests. The *value of C* for $k \geq 1$ requests at system history $h \in \mathcal{H}$ when the target is in state $t \in T$, denoted $\mathcal{Y}^C_k(h, t)$, is defined as follows:

$$\mathcal{Y}^C_k(h, t) = \sum_{a \in \mathcal{A}} [\mathcal{P}_{req}(t, env(last(h)), a) \times IR^C(h, t, a) + \sum_{\substack{s' \in S \\ \langle t, e, a, t' \rangle \in \delta}} \mathcal{P}_{\mathcal{S}}(last(h), a, C(h, a), s') \times \mathcal{Y}^C_{k-1}(h \xrightarrow{a, C(h,a)} s', t')],$$

where $\mathcal{Y}^C_0(h, t) = 0$, for all $h \in \mathcal{H}$ and $t \in T$, and $IR^C(h, t, a)$ stands for the *immediate* reward collected by the controller C when

[1] Although, as we shall see later, under the full observability assumption, it is enough for a controller to depend only on the final state of the enacted system—rather than the whole history–we shall work with the most general definition that could also be used in settings with partial observability.

requested to delegate action a at history h:

$$IR^C(h,t,a) = \begin{cases} R(t,a) & \text{if } \exists s'.\mathcal{P}_{\mathcal{S}}(last(h),a,C(h,a),s') > 0; \\ 0 & \text{if } C(h,a) = u; \\ -R(t,a) & \text{otherwise.} \end{cases}$$

The value of a controller for k steps of target $\mathcal{T}$ in a system $\mathcal{S}$ is defined as $\mathcal{Y}_k^C = \mathcal{Y}_k^C(s_0,t_0)$. We say that a controller C^* is a *k-composition* if for all other controllers C, $\mathcal{Y}_k^{C^*} \geq \mathcal{Y}_k^C$ holds.

Since the target may include infinite traces, we are in general interested in controllers that are optimal for any number of potential requests, that is, for infinite executions of the target behavior. To cope with unbounded executions of the target, we appeal to the use of a *discount factor*, as customary in sequential decision making over infinite episodes [6, 3]. The idea is that the satisfaction of later target-compatible requests are less important than those issued earlier. Formally, the value of a controller C, denoted by $\mathcal{Y}_\gamma^C(h,t)$, relative to a discount factor $0 \leq \gamma < 1$, is defined as follows:

$$\mathcal{Y}_\gamma^C(h,t) = \sum_{a\in\mathcal{A}} [\mathcal{P}_{req}(t, env(last(h)), a) \times IR^C(h,t,a) + \gamma \sum_{\substack{s'\in S \\ \langle t,e,a,t'\rangle\in\delta}} \mathcal{P}_{\mathcal{S}}(last(h),a,C(h,a),s') \times \mathcal{Y}_\gamma^C(h \xrightarrow{a,C(h,a)} s', t')].$$

The use of a discount factor plays the same role as in infinite horizon Markov decision processes, namely, it allows convergence of the value of a controller [3, 8]. Note that the assumption that temporally closer rewards are more important than distant ones is particularly suitable in the context of composition problems, where behaviors may fail, the target and available system may be reset, or the problem may not be fully solvable.

As with the finite case, the value of a controller for a given target $\mathcal{T}$ and system $\mathcal{S}$ is defined as $\mathcal{Y}_\gamma^C = \mathcal{Y}_\gamma^C(s_0,t_0)$. Finally, we say that a controller C^* is a *γ-composition* (of target $\mathcal{T}$ in system $\mathcal{S}$) if for all other controllers C, it is the case that $\mathcal{Y}_\gamma^{C^*} \geq \mathcal{Y}_\gamma^C$.

Put it all together, the decision theoretic behavior composition problem, or simply *DT-composition problem*, amounts to synthesize a γ-composition for a given system $\mathcal{S}$, target behavior $\mathcal{T}$, and discount factor γ.

Exact compositions.

A behavior composition problem has an exact solution when there exists a controller that can fully realize the target, that is, a controller that can *always* honor the target's requests, no matter what. There have recently been various approaches in the literature to synthesize such a controller, called a *composition*, if any exists (see, e.g., [5, 10, 11, 4]). Within our decision theoretic setting, it is important to clearly define what an exact solution is and its relationship with "optimal" controllers.

Since the target behavior is deterministic, its specification can be seen as the set of all possible sequences of actions that can be requested, starting from the initial state. Thus, given any finite run of the target, the most one could expect is that every single action has been successfully realized in the system. This would imply that all possible rewards in the run have indeed been collected. Since one does not know a priori which actual run will ensue, we consider the maximum expected reward when running the target. To make this precise, we define $\mathcal{R}_k^{\max}(t,e)$ as the *maximum* expected reward when running the target from its state t at environment state e for $k \geq 0$ steps as follows (here, $\mathcal{R}_0^{max}(t,e) = 0$):

$$\mathcal{R}_{k\geq 1}^{\max}(t,e) = \sum_{a\in\mathcal{A}} [\mathcal{P}_{req}(t,e,a) \times R(t,a)] + \sum_{\substack{e'\in E \\ \langle t,e,a,t'\rangle\in\delta}} [\mathcal{P}_{\mathcal{E}}(e,a,e') \times \mathcal{R}_{k-1}^{\max}(t',e')].$$

As above, we take $\mathcal{R}_k^{\max} = \mathcal{R}_k^{\max}(t_0,e_0)$, for any $k \geq 0$. Note that this definition is well defined for both cyclic and acyclic targets. Of course, for an acyclic target with a longest path of length ℓ, it is easy to show that $\mathcal{R}_k^{\max} = \mathcal{R}_\ell^{\max}$, for every $k \geq \ell$.

Thus, a controller C is an *exact composition* if $\mathcal{Y}_k^C = \mathcal{R}_k^{\max}$, for all $k \geq 1$, that is, C can *fully* and *always* realize a target behavior in the available system. Note that controllers are meant to have full observability of the current history. A *Markovian* (i.e., memoryless) controller C is one that only looks at the current state of the system to decide the delegation; formally, for all histories $h, h' \in \mathcal{H}$ such that $last(h) = last(h')$ and action $a \in \mathcal{A}$, $C(h,a) = C(h',a)$ applies. When it comes to exact solutions, Markovian controllers are enough under full observability.

THEOREM 1. *Let $\mathcal{S}$ be a system and $\mathcal{T}$ be a target behavior. Then, if there exists an exact solution for realizing $\mathcal{T}$ in $\mathcal{S}$, then there exists a Markovian controller which is also an exact solution.*

PROOF. Let C^* be an exact solution for realizing $\mathcal{T}$ in $\mathcal{S}$. For any $h \in \mathcal{H}$ and $a \in \mathcal{A}$, we define a new controller $\widehat{C}(h,a) = C^*(h',a)$ if $h' \in \mathcal{H}_{C^*}$ is such that $last(h) = last(h')$ and for all $h'' \in \mathcal{H}_{C^*}$ such that $last(h'') = last(h)$, it is the case that $C(h',a) \leq C(h'',a)$ (we assume $u > i$, for any $i \in \{1,\ldots,n\}$). Otherwise, if such history h' does not exist, we take $\widehat{C}(h,a) = u$.

It is easy to check that $\widehat{C}$ is well-defined. In addition, $\widehat{C}$ is Markovian. In fact, consider two histories $h_1, h_2 \in \mathcal{H}$ such that $last(h_1) = last(h_2)$, and suppose that $\widehat{C}(h_1,a) = k_1$. Then, $C^*(h_1',a) = k_1$, for some $h_1' \in \mathcal{H}_{C^*}$ and it is easy to show that $\widehat{C}(h,a) = C^*(h_1',a) = k_1$ as well, the same witness history h_1' can be used for h_2 too. Furthermore, because h_1' is reachable by C^*, together with the fact that C^* is indeed an exact solution, implies that $k_1 \in \{1,\ldots,n\}$ is a correct delegation, in the sense that behavior $\mathcal{B}_{k_1}$ is able to perform a legal step on action a when the environment is in state $env(last(h))$, and since $last(h) = last(h_1')$, such delegation is also legal at history h and $\widehat{C}$ is also exact. □

More importantly, exact solutions are guaranteed to be always optimal controllers under unbounded runs, independently of the discount factor chosen.

THEOREM 2. *If a controller is an exact composition for a decision-theoretic behavior composition problem, then such controller is a γ-composition, for any $0 \leq \gamma < 1$.*

PROOF (SKETCH). Let C^* be an exact solution to a DT-composition problem, and assume, wlog, a target with a uniform reward α. Then, at each step, C^* collects the maximum possible reward of α. If a discount factor γ is used, then C^* will collect a reward of $\alpha \times \sum_{n=1}^{\ell} \gamma^{n-1}$ over ℓ steps, which is indeed the maximum possible reward for a γ-composition after ℓ steps. Hence, C^* is also a γ-composition for the given composition problem. □

4. SOLVING DT-COMPOSITIONS

Various techniques have been used to actually solve classical behavior composition problems, including PDL satisfiability [5],

search-based approaches [11], LTL/ATL synthesis [9, 4], and computation of special kind of simulation relations [10, 2]. Unfortunately, in the context of the decision theoretic framework from Section 2, none of these techniques can be applied. In this section, we show how to solve a decision theoretic composition problem, by reducing it to a Markov decision problem in a natural manner. We also demonstrate the reduction with a proof of concept implementation using an off-the-shelf existing MDP solver.

Markov decision processes.

A Markov decision process (MDP) is a discrete time stochastic control process [8, 3]. At each step, the process is in a state q, the decision maker chooses an action a, the process evolves to a successor state q' with some probability, and the decision maker receives certain reward r. The "best" decision maker is one that collects maximum (potential) rewards over time.

Formally, a *Markov decision process* (MDP) is a tuple $\mathcal{M} = \langle Q, A, p, r\rangle$, where:

- Q is a (finite) set of states;
- A is a (finite) set of actions;
- $p : Q \times A \times Q \mapsto [0, 1]$ is the probabilistic state transition function: $p(q, a, q')$ denotes probability of the process evolving to state q' when action a is executed in state q;
- $r : Q \times A \mapsto \mathbb{R}$ is the reward function: $r(q, a)$ denotes the immediate reward obtained when action a in executed state q.

A policy—the decision maker—is a collection of state-action mappings stating what action to take in each state of the process. Formally, a (stationary) *policy* is a function $\pi : Q \mapsto A$; $\pi(q)$ denotes the action to be taken in state q. Solving an MDP involves then computing a policy that accumulates maximum reward over time. In doing so, one can be interested in finite horizon problems, where the decision maker is meant to perform a fixed number of sequential decisions, or infinite horizon problems, where rewards over infinite runs of the MDP are considered.

So, the value of an optimal policy in a state q for a *finite* horizon k is given by the following Bellman's *principle of optimality* [8, 3]:

$$V_k^*(q) = \max_{a\in A}\{r(q, a) + \sum_{q'\in Q} p(q, a, q') \times V_{k-1}^*(q')\}.$$

Similarly, the value of an optimal policy in a state q for *infinite* horizon relative to a discount factor of $0 \leq \gamma < 1$ is as follows [7]:

$$V^*(q) = \max_{a\in A}\{r(q, a) + \gamma \sum_{q'\in Q} p(q, a, q') \times V^*(q')\}.$$

Howard [7] showed that there always exists an optimal *stationary* policy for infinite horizon problems, that is, one that does not depend on which stage a decision is taken.

From behavior composition to MDPs.

With the notion of MDPs at hand, we show next how to reduce a DT-composition problem, as described in Sections 2 and 3, to the problem of solving an MDP.

Informally, in our setting, the decision maker is the controller, and thus, the possible actions that can be taken are those of behavior delegation. Consider then a system $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{E}\rangle$, with $\mathcal{T}_\mathcal{S}$ denoting the corresponding enacted system behavior, and a target $\mathcal{T} = \langle T, t_0, \delta, R, \mathcal{P}_{req}\rangle$. We define the corresponding MDP encoding $\mathcal{M}_{\mathcal{S},\mathcal{T}} = \langle Q, A, p, r\rangle$ as follows:

- $Q = \mathcal{S} \times T \times \mathcal{A} \cup \{q_{init}\}$, where for all $\langle s, t, a\rangle \in Q$, $\mathcal{P}_{req}(t, env(s), a) > 0$. Given an MDP state $q = \langle s, t, a\rangle \in Q$, we define $sys(q) = s$, $tgt(q) = t$, and $req(q) = a$. A special, domain independent, state q_{init} is used as a "dummy" initial state of the process.
- $A = Index = \{1, \ldots, n, u\}$, that is, an action in the encoded MDP stands for a behavior selection (or no selection at all).
- The state transition function is defined as follows:

$$p(q, i, q') = \begin{cases} \mathcal{P}_{req}(tgt(q'), env(sys(q')), req(q')), \text{ if} \\ \quad q = q_{init}, sys(q') = s_0, tgt(q') = t_0; \\ \\ \mathcal{P}_\mathcal{S}(sys(q), req(q), i, sys(q')) \times \\ \quad \mathcal{P}_{req}(tgt(q'), env(sys(q')), req(q')), \text{ if} \\ \quad q \neq q_{init}; \\ \\ 0, \text{ otherwise.} \end{cases}$$

- The reward function is defined as ($\alpha = R(tgt(q), req(q))$):

$$r(q, i) = \begin{cases} \alpha & \text{if } \mathcal{P}_\mathcal{S}(sys(q), req(q), i, sys(q')) > 0 \\ & \quad \text{for some } q' \in Q \text{ and } q \neq q_{init}; \\ \\ 0 & \text{if } i = u \text{ or } q = q_{init}; \\ \\ -\alpha & \text{otherwise.} \end{cases}$$

In the resulting MDP, a state is built from the state of the enacted system behavior (which includes the states of the environment and those of all available behaviors), the state of target behavior, and an action being requested; in other words, a "snapshot" of the whole composition problem. Each transition in the MDP represents the behavior—through its index—to which the current request is delegated for execution. The dynamics of the MDP encodes both the dynamics of the enacted system behavior and the target behavior, as well as that of the stochastic process (i.e., the user of the target) that is requesting actions. Finally, the reward function in the MDP merely mimics that of the encoded behavior composition problem; no reward is given from the initial dummy state, and an unfeasible delegation (i.e., one where the chosen behavior may not perform the action) receives a penalty (i.e., it is better to prescribe "u").

Given a policy $\pi : Q \mapsto Index$ for the MDP $\mathcal{M}_{\mathcal{S},\mathcal{T}}$, we define the *induced controller* $C_\pi(h, a)$, where $h = s^0 \xrightarrow{a^1,k^1} \cdots \xrightarrow{a^\ell,k^\ell} s^\ell$, with $\ell \geq 0$ and $a \in \mathcal{A}$, as follows:

$$C_\pi(h, a) = \begin{cases} \pi(q) & \text{if } sys(q) = last(h), a = req(q), \text{ and} \\ & \quad t_0 \xrightarrow{env(s^0):a^1} \cdots \xrightarrow{env(s^{\ell-1}):a^\ell} tgt(q) \text{ in } \mathcal{T}; \\ u & \text{otherwise.} \end{cases}$$

Note that the output for histories that do not yield any legal evolution of the target is irrelevant, and hence, we just output value u.

We now state the main result of the section, namely, a solution to the encoded MDP yields an optimal controller for the corresponding DT-composition problem.

THEOREM 3. *Let $\mathcal{S}$ be an available system and $\mathcal{T}$ a target behavior. Let $\mathcal{M}_{\mathcal{S},\mathcal{T}}$ be the corresponding MDP encoding as described above. If π^* is an γ-optimal policy for $\mathcal{M}_{\mathcal{S},\mathcal{T}}$, then its induced controller C_{π^*} is an γ-composition for realizing $\mathcal{T}$ in $\mathcal{S}$.*

PROOF (SKETCH). This is proved by showing that a solution to the optimality equation of the encoded MDP conforms to the solution of the equation for a composition. Specifically, we show, by induction on ℓ, that $\gamma\mathcal{Y}_\ell^{C_\pi}(s_0,t_0) = V_{\ell+1}^\pi(q_{init})$, where γ is a discount factor, $\ell \geq 0$, and π is a policy for $\mathcal{M}_{\mathcal{S},\mathcal{T}}$. To that end, we rely on the fact that the value of a controller and that of a policy in the MDP can be re-written as follows. First, the value of the controller for $k+1$ steps can be re-written as:

$$\mathcal{Y}_{k+1}^{C_\pi}(s_0,t_0) = \mathcal{Y}_k^{C_\pi}(s_0,t_0) + \gamma^k \sum_{h\in\mathcal{H}_k^{C_\pi}} [Pr_{C_\pi}(h) \sum_{a\in\mathcal{A}} \mathcal{P}_{req}(t_h, env(last(h)), a) \times IR^{C_\pi}(h,t_h,a)],$$

where $\mathcal{H}_k^{C_\pi}$ is the set of all histories of length k that may be reached by following controller C_π, $Pr_{C_\pi}(h)$ is probability of such history arising, and t_h stands for the resulting (unique) state of the target after having performed all the actions in h.

Second, the cumulative reward gained by policy π after $k+2$ steps can be re-written as follows:

$$V_{k+2}^\pi(q_{init}) = V_{k+1}^\pi(q_{init}) + \gamma^{k+1} \sum_{\lambda\in\Lambda_{k+1}^\pi} [Pr_\pi(\lambda) \times r(last(\lambda), \pi(last(\lambda)))],$$

where Λ_k^π is the set of all MDP sequence of states of length k that may be traversed when following policy π, and $Pr_\pi(\lambda)$ is the probability of sequence λ arising.

The fact that $\gamma\mathcal{Y}_\ell^{C_\pi}(s_0,t_0) = V_{\ell+1}^\pi(q_{init})$, together with the fact that every controller is always related to some policy in the MDP (even non-Markovian controllers), is enough to prove the thesis. □

This result proves the correctness of the encoding, and provides us with a technique for solving DT-composition problems, by using, for instance, policy-iteration implementations [7].

EXAMPLE 4. We generated the optimal policy for the garden scenario from Figure 1 by using a simple existing MDP solver.[2] The problem does *not* actually have an exact solution. To see that, consider the sequence of action requests *clean · water · empty* compatible with the target $\mathcal{T}_{Garden}$. It is not hard to verify that the first and last actions need to be delegated to bot $\mathcal{B}_{Cleaner}$, whereas the second action *water* ought to be delegated to bot $\mathcal{B}_{Multi}$. However, bot $\mathcal{B}_{Cleaner}$ will be able to perform the last action *empty* only if it has evolved to state a_1 after *clean*'s execution. Otherwise, if $\mathcal{B}_{Cleaner}$ happens to stay in state a_0 instead, action *empty* cannot be realized in the system $\mathcal{S}_{Garden}$ and a dead-end is reached.

Note, though, that the chances of $\mathcal{B}_{Cleaner}$ evolving to state a_0 are indeed low. Hence, an optimal controller—a composition—should still choose $\mathcal{B}_{Cleaner}$ to execute the first *clean* action. This is indeed the controller induced by the optimal policy found when solving the corresponding MDP, which is partially listed below as output by the MDP solver (`BEH0`, `BEH1`, and `BEH2` stand for behaviors $\mathcal{B}_{Cleaner}$, $\mathcal{B}_{Multi}$, and $\mathcal{B}_{Plucker}$, respectively):

```
Beh:0 0 0 | Tgt:0| Env:0|Act:CLEAN ------> BEH0
Beh:0 0 0 | Tgt:1| Env:0|Act:WATER ------> BEH1
Beh:1 0 0 | Tgt:1| Env:0|Act:WATER ------> BEH1
Beh:1 0 0 | Tgt:2| Env:0|Act:EMPTY ------> BEH0
Beh:0 0 0 | Tgt:2| Env:0|Act:EMPTY ------> U
...
```

[2] `http://copa.uniandes.edu.co/software/jmarkov/`

Observe that if after doing a *clean* action, behavior $\mathcal{B}_{Cleaner}$ (`BEH0`) stays in its state a_0, the policy prescribes `U`, thus signaling a dead-end in the composition.

In turn, the following rules in the policy will successfully realize the request sequence *clean · pluck · empty*:

```
Beh:0 0 0 | Tgt:0| Env:0|Act:CLEAN ------> BEH0
Beh:0 0 0 | Tgt:1| Env:0|Act:PLUCK ------> BEH1
Beh:0 1 0 | Tgt:1| Env:0|Act:WATER ------> BEH1
Beh:0 1 0 | Tgt:3| Env:0|Act:EMPTY ------> BEH1
```

Finally, bot $\mathcal{B}_{Plucker}$ (`BEH2`) will be used by the induced controller in cases as the following ones:

```
Beh:0 1 0 | Tgt:1| Env:0|Act:PLUCK ------> BEH2
Beh:0 1 1 | Tgt:0| Env:0|Act:CLEAN ------> BEH2
```

Observe that in the configuration of the second rule, behavior $\mathcal{B}_{Cleaner}$ is also able to perform the cleaning action; however, it is best to use the plucker bot as it will bring it to state c_0, from where it is able to pluck again if needed (see that bot $\mathcal{B}_{Multi}$ is in state b_1 from where it cannot pluck).

All the above rules are only for the cases in which the environment remains in its state e_0, other (similar) rules exist in the policy/controller for other environment states. ■

Exact compositions.

As discussed, in a decision theoretic composition problem, one looks, in general, for the "best" possible controller, since exact compositions may not exist. Nonetheless, the following result states that if one does exist, it is enough to restrict to the finite horizon case in the corresponding MDP (without losing optimality).

THEOREM 4. *If there exists an exact composition for realizing a given target $\mathcal{T}$ in a system $\mathcal{S}$, then the controller induced by any $(|Q|+1)$-optimal policy for MDP $\mathcal{M}_{\mathcal{S},\mathcal{T}}$ is an exact composition.*

PROOF (SKETCH). This follows from the fact that there exists an optimal policy for $\mathcal{M}_{\mathcal{S},\mathcal{T}}$ that is stationary (which can be proven by relying on the fact that there exists a Markovian exact composition due to Theorem 1), and the fact that by optimizing the MDP up to $Q+1$ steps, it is guaranteed that *all* possible configurations of the whole composition framework—which includes both available system and target—are taken into account. □

This result is important in that it provides a way of verifying whether a DT-composition problem accepts an exact solution, namely, find an optimal policy π for horizon $|Q|+1$ and check whether $\mathcal{Y}_{|Q|}^{C_\pi} = \mathcal{R}_{|Q|}^{\max}$ (recall the first step in the MDP involves no action request and attracts no reward). Of course, it is possible to restate the above theorem in terms of an infinite horizon problem:

COROLLARY 1. *If there exists an exact composition for realizing a given target $\mathcal{T}$ in a system $\mathcal{S}$, then there exists a discount factor $\hat{\gamma}$ such that for any γ-optimal policy π for MDP $\mathcal{M}_{\mathcal{S},\mathcal{T}}$, with $\gamma \geq \hat{\gamma}$, the induced controller C_π is an exact composition of $\mathcal{T}$ in $\mathcal{S}$.*

When no exact composition exists, though, all one can do is to settle for the (best) controller induced by an optimal policy in the encoded MDP. Since non-exact compositions will include dead-ends, that is, possible histories where some target-compatible request may not be fulfilled, other mechanisms will be required to bring the overall system to a "healthy" configuration, such as resetting the whole system or even some parts of it.

We close this section by relating our approach to behavior composition with the "classical" approaches to the problem in the literature (e.g., [5, 10, 11, 4]). In such approaches, the task amounts to decide whether an *exact* composition controller exists (and to synthesize one if any) in settings under strict uncertainty. The dynamics of behaviors and that of the environment are represented by means of *transition relations*, rather than probabilistic transition functions. As a result, the designer can only model whether a transition is possible or not. In addition, the target behavior does not include a probabilistic request function $\mathcal{P}_{req}$, but simply a transition relation stating what actions can be legally requested.

As expected, the following result states that our DT-composition framework is at least as expressive as the classical one.

THEOREM 5. *For any instance of a classical behavior composition (as in [10, 11]), there is a decision-theoretic behavior composition instance such that there exists a composition solution for the former iff there exists an exact composition for the latter.*

PROOF (SKETCH). This is shown by building a DT-composition problem instance as follows:

- The environment probabilistic transition function is defined such that $\mathcal{P}^{\mathcal{E}}_{next}(e, a, e') = 1/|\Delta(e, a)|$, whenever $\langle e, a, e' \rangle \in \rho$, where ρ is the transition relation of the original classical environment and $\Delta(e, a) = \{e' \mid \langle e, a, e' \rangle \in \rho\}$.
- The probabilistic transition function for each available behavior $\mathcal{B}_i$ is defined as $\mathcal{P}^{\mathcal{B}_i}_{next}(b, e, a, b') = 1/|\Delta(b, e, a)|$, whenever $\langle b, e, a, b' \rangle \in \delta_i$, where δ_i is the transition relation of the original classical available behavior $\mathcal{B}_i$ and $\Delta(b, e, a) = \{b' \mid \langle b, e, a, b' \rangle \in \delta_i\}$.
- The probabilistic action request function of the target behavior is defined $\mathcal{P}_{req}(t, e, a) = 1/|\Delta(t, e)|$, whenever $\langle t, e, a, t' \rangle \in \delta_T$, where δ_T is the transition relation of the original target and $\Delta(t, e) = \{a \mid \langle t, e, a, t' \rangle \in \delta_T\}$.
- The target reward function is defined as $R(t, a) = 1$ for all $a \in \mathcal{A}$ and $t \in T$ such that $\mathcal{P}_{req}(t, e, a) > 0$ for some $e \in E$.

(In all other cases, the probabilities are assumed to be zero.) It is not hard to show that the resulting DT-composition instance has an exact composition iff the original classical one has a solution. □

Clearly, not every DT-composition problem can be mapped to the classical setting, as it is the case with our gardening scenario. It follows then that the framework developed here is, not surprisingly, a strict extension of the classical ones for behavior composition.

We observe that all previous approaches provide an EXPTIME upper bound to the computational complexity. Fully observable MDPs can be solved in time polynomial in the size of state space and actions [8]. Since the size of $\mathcal{M}_{\mathcal{S},\mathcal{T}}$'s state space is indeed exponential in the number of behaviors, such bound still applies here.

5. CONCLUSIONS

In this paper, we have generalized the classical behavior composition problem (e.g., [11, 5, 10]) to one that is able to account for *quantified uncertainties* in the domain, both in the dynamics of the behaviors and environment, as well as in the preferences over requests from the target user. The task then is to find the "best" controller—a composition—that maximizes the *expected realizability* of the target. Unlike previous approaches, the extended decision theoretic composition framework is able to deal with unsolvable problem instances, that is, those that do not accept exact solutions. In addition, it is provably more expressive than the classical version under strict uncertainty. In order to solve a DT-composition problem, we showed how to reduce it to the problem of finding an optimal policy in a Markov decision process, an established framework for sequential stochastic decision making.

There are many open lines of research in this framework. A natural extension is to accommodate preferences over available behaviors. In many applications, using one component may be more costly than using another one, e.g., it is preferred to transport goods by car than to do by truck. Though catering for this may appear straightforward to achieve by simply encoding, for instance, a ranking over available behaviors in the reward function of the MDP, it is not clear that all the results presented here would generalize. In fact, under such setting, exact solutions may no longer be optimal controllers. Another interesting issue is to combine our framework with that of classical behavior composition in the literature. The idea is that some actions in the target may not be compromised and *must* be met in any composition. For example, once the garden has been plucked, it is mandatory that the collected fruit be adequately stored. Yet another possibility is to generalize the framework to one under partial observability, and possibly using partially-observable MDPs (POMDPs) to tackle those cases. Lastly, if rewards and transitions are not fully known, a reinforcement learning framework could be used to find compositions while learning the domain.

Acknowledgments

We thank the anonymous reviewers and Lawrence Cavedon for their helpful comments. We also acknowledge the support of the Australian Research Council (under grant DP1094627).

6. REFERENCES

[1] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella. Automatic service composition based on behavioural descriptions. *International Journal of Cooperative Information Systems*, 14(4):333–376, 2005.

[2] D. Berardi, F. Cheikh, G. De Giacomo, and F. Patrizi. Automatic service composition via simulation. *International Journal of Foundations of Computer Science*, 19(2):429–452, 2008.

[3] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and compuatational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, July 1999.

[4] G. De Giacomo and P. Felli. Agent composition synthesis based on ATL. In *Proc. of AAMAS*, pages 499–506, 2010.

[5] G. De Giacomo and S. Sardina. Automatic synthesis of new behaviors from a library of available behaviors. In *Proc. of IJCAI*, pages 1866–1871, 2007.

[6] S. French. *Decision Theory: An Introduction to the Mathematics of Rationality*. Ellis Horwood, 1986.

[7] R. Howard. *Dynamic Programming and Markov Process*. The MIT Press, 1960.

[8] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[9] S. Sardina and G. De Giacomo. Realizing multiple autonomous agents through scheduling of shared devices. In *Proc. of ICAPS*, pages 304–312, 2008.

[10] S. Sardina, F. Patrizi, and G. De Giacomo. Behavior composition in the presence of failure. In *Proc. of KR*, pages 640–650, 2008.

[11] T. Stroeder and M. Pagnucco. Realising deterministic behaviour from multiple non-deterministic behaviours. In *Proc. of IJCAI*, pages 936–941, 2009.

Solving Election Manipulation Using Integer Partitioning Problems

Andrew Lin
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
apl8378@cs.rit.edu

ABSTRACT

An interesting problem of multi-agent systems is that of voting, in which the preferences of autonomous agents are to be combined. Applications of voting include modeling social structures, search engine ranking, and choosing a leader among computational agents. In the setting of voting, it is very important that each agent presents truthful information about his or her preferences, and not manipulate. The choice of election system may encourage or discourage voters from manipulating. Because manipulation often results in undesirable consequences, making the determination of such intractable is an important goal.

An interesting metric on the robustness of an election system concerns the frequency in which opportunities of manipulations occur in a given election system. Previous work by Walsh has evaluated the frequency of manipulation in the context of very specific election systems, particularly veto, when the number of candidates is limited to at most three, by showing that manipulation problems in these systems can be directly viewed as problems of (Two-Way) Partition, and then using the best known heuristics of Partition. Walsh also claimed similar results hold for k-candidate veto election by way of problems involving multi-way partitions.

We show that the results for k-candidate veto elections do not follow directly from common versions of partition problems and require non-trivial modifications to Multi-Way Partition. With these modifications, we confirm Walsh's claim that these elections are also vulnerable to manipulation. Our new computational problems also allow one to evaluate manipulation in the general case of k-candidate scoring protocols. We investigate the complexity of manipulating scoring protocols using new algorithms we derive by extending the known algorithms of Multi-Way Partition.

It is our conclusion that the problems of manipulation in more general scoring protocols of four or more candidates are not vulnerable to manipulation using extensions of the current known algorithms of Multi-Way Partition. This may be due to weaknesses in these algorithms or complexity in manipulating general scoring protocols.

Cite as: Solving Election Manipulation Using Integer Partitioning Problems, Andrew Lin, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 583–590.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms, Theory, Experimentation, Performance

Keywords

computational social choice, manipulation, scoring protocols, partition

1. INTRODUCTION

A multi-agent system is composed of multiple interacting intelligent agents, and is used to solve problems that are otherwise difficult or impossible for an individual agent to solve. An interesting problem in multi-agent systems is that of voting [15], in which individual preferences of these agents are to be combined. Voting is used by multi-agent systems in interesting applications such as search engine ranking [17]. To ensure the integrity of the outcome of an election, an important goal in designing election systems is to eliminate or at least limit the opportunity for an agent to have an incentive to report false preferences for personal gain.

An unfortunate result, the Gibbard-Satterthwaite theorem [8, 16], shows that manipulation in inevitable in all reasonable election systems. Manipulation, also known as strategic voting, occurs when one or more voters vote contrary to their true preference ordering. A typical manipulator may bury his or her 2^{nd} true preference to give his first preference a larger relative advantage.

Bartholdi, Tovey, and Trick [2] attack this problem by evaluating the worst-case time complexity of computing such manipulations, and show that in many cases this problem is NP-hard, making NP-hardness the standard for worst-case hardness in manipulation problems. It has since been shown that most non-trivial weighted elections, and particularly scoring protocols, are NP-hard to manipulate [4, 9].

Since NP-hardness only demonstrates worst-case hardness and may not fully reflect the difficulty of finding manipulations in typical settings of interest, more recent work has been done to demonstrate the tradeoffs between fairness of elections and frequency of manipulation, as well as hardness of manipulation in random elections. It is now known that in some cases of random elections, as well as elections involving correlated voters such as single-peaked preferences [1, 6], manipulation is easy either in the worst-case or for the nearly all cases asymptotically.

Walsh [18] furthers the study empirically, to investigate issues such as hidden constants in theoretical asymptotic results, as well as the effects of realistic assumptions such as election systems being bounded in size. It is of interest to discover how the known asymptotic behavior applies in practice.

The weighted veto rule of three candidates was studied in Walsh, particularly because it is most directly related to the NP-hard problem of Partition, for which reasonable heuristics, particularly the Complete Karmarkar-Karp (CKK) Algorithm, are known [12]. The runtime of invoking the CKK algorithm, in particular as measured by the number of search-tree branches generated, is thus utilized throughout the study as a metric to evaluate the hardness of finding a manipulation in an election, or showing that none exists. Particularly of interest is the result that hard instances of manipulation for instances of veto elections of three candidates are "rare" and require highly correlated voters. Walsh also remarks that the Partition problem also applies in other cases of 3-candidate scoring systems, as well as veto systems of multiple candidates, without demonstrating the constructions involved.

In Walsh, the frequency of which manipulations occur for this system is determined for varying election sizes, and is shown to exhibit a smooth transition in probability in relation to the number of manipulators. More specifically, it is shown that the probability that a coalition of m manipulators can influence the outcome in an election of n elections is directly correlated with the ratio $\frac{\sqrt{n}}{m}$, and is independent of the problem size. This is known as a phase transition, and the majority of NP-complete problems exhibit their worst-case complexity in instances defined by this region.

It is then shown that in most cases of uncorrelated voters, including uniform votes, as well as when voter weights are normally distributed, on average one may find a manipulation or prove that none exists with very minimal search, averaging slightly more than one search-tree branch in a CKK search. This is true even in the range of the phase transition, and is in direct contrast with other NP-complete problems, in which hard problems are highly associated with this transition.

In the extreme case of correlated voters, a case in which all non-manipulative voters veto the candidate of interest, and which the manipulators have a total weight of twice the non-manipulative voters, is considered. In this case, the runtime of invoking the CKK algorithm exhibits a phase transition and grows exponentially in relation to the number of manipulators in the election. However, a further case was studied in which the election is highly correlated as before but with one additional agent who votes at random among the three candidates. Surprisingly, the runtime, as measured in search-tree branches of CKK of manipulating such an election rapidly decreases as the distribution of the weight of this non-correlated voter increases, and exhibits a strong phase transition into a constant as this distribution approaches that of the correlated voters. Based on these results, it is concluded that in the case of 3-candidate weighted veto systems, hard manipulation problems only occur when all of the votes are highly correlated, and are thus exceedingly rare in any reasonable distribution of voters. Unlike common NP-complete problems, hard instances in such manipulation problems are also not directly related to the phase transition of frequency of manipulation.

Although it is remarked in [18] that this technique can be applied to that of all scoring systems of three candidates, and that heuristics of the more general k-Way Partition to that of veto systems of more than three candidates, this construction as well as the problem of manipulating other scoring systems of more than three candidates was left open. We propose a solution remedy these open problem. We confirm the results of Walsh for that of k-candidate veto elections, but show that this problem does not directly relate to the problem of $(k-1)$-Way Manipulation, and require non-trivial adjustments. We then show how these adjustments allow one to evaluate general scoring protocols of more than three candidates.

We attack the open problem of Walsh by introducing an analogous partition-like problem for the case of general scoring systems, and extending the known algorithms and heuristics of Partition and k-Way Partition to that of our new problem in a very natural way. We show how problems involving partitions are related to that of manipulation, as scores given by the voters need to be partitioned among the candidates in a way to ensure a certain desirable outcome. In doing so we are able to investigate whether the known algorithms of these problems will give new results to scoring systems of interest.

Based on our analogous testing of the results of Walsh for these cases, we conclude that the algorithms of Korf, the best-known algorithms for some partition problems, support the results of Walsh for k-candidate veto elections of $k > 3$, but require some non-trivial adjustments to the problem instances and algorithms involved.

As our adjustments further allow one to attack the problems of manipulation in more general scoring protocols, we also study the runtime of invoking these new algorithms on such instances. It is our conclusion that the problem of manipulating general scoring protocols, as well as families of scoring protocols, such as veto, are not vulnerable to the best-known algorithms of k-Way Partition and analogous extensions thereof.

This may either support the fact that elections of more than three candidates are inherently more resistant to manipulation, or weaknesses in these partition algorithms. We leave these options as an open problem.

We organize the paper as follows. In the preliminaries section, we formally define relevant problems and known algorithms, as well as the connection between partition problems and election manipulation. In Section 3, we introduce our extensions of the known problem of k-Way Partition, which we will use to solve some instances of manipulation, and introduce analogous extended algorithms. In the next two sections, we define the connection between the problem of manipulation and the new partition problems, and demonstrate the experimental results graphically.

2. PRELIMINARIES

2.1 Election Systems

An **election** $E = (C, V)$ consists of a set of **candidates** $C = \{c_1, \ldots, c_m\}$ and **voters** $V = \{v_1, \ldots, v_n\}$. Each voter $v \in V$ presents a **preference ordering** over the candidates C, in the form of a complete linear ordering $c_{i_1} > \cdots > c_{i_m}$. An **election system** $\mathcal{E} : E \to C^+$ maps an election to winning candidates(s) based on the preferences of the voters, for which it aggregates. An interesting set of election

systems, scoring protocols, which we will evaluate in this paper, encompasses systems such as plurality, veto, and the Borda count where the number of candidates is fixed.

An m-candidate **scoring protocol** is defined as a vector $(\alpha_1, \ldots, \alpha_m)$. In such an election system, each voter contributes α_i points to the i^{th} choice of his or her preference ordering. The candidate with the highest score wins. An election defined by such a scoring protocol can further be weighted, in which each voter v is given a non-negative integer weight $w(v)$. In this case, the vote is counted as $w(v)$ non-weighted votes, and thus the i^{th} choice of candidate c is awarded $w(v)\alpha_i$ points. A prominent example of weighted elections occurs in the U.S. Presidential Elections, in which the Electoral Colleges are given different weights. Weights are also commonly used in computational elections, such as search engine ranking aggregation [17, 5].

A **family of scoring protocols** is an infinite series of scoring protocols $(\alpha^1, \ldots, \alpha^m, \ldots)$ in which $\alpha^m = (\alpha_1^m, \ldots, \alpha_m^m)$ is an m-element scoring protocol.

2.2 Manipulation

A common problem with many election systems is the incentive for some voters to vote contrary to their true preferences, either individually or collectively in a coalition, manipulating the outcome. This can occur when some voters vote for their 2^{nd} preference to avoid "wasting a vote" on their favorite candidate whom is not popular, or bury their 2^{nd} preference to give their first preference a larger relative advantage. Unfortunately, several early results have shown that the existence of such strategies in elections is inevitable [8, 16], and an early compromise was made to make the determination of such results at least NP-hard [2]. We define manipulation as a decision problem as follows.

Name: $\mathcal{E}$-Manipulation

Instance: Candidates C, established voters V, unestablished voters V', and distinguished candidate $p \in C$.

Question: Is there an assignment of preference profiles over C for V' such that p is a winner of the election $(C, V \cup V')$?

Although this problem is indeed NP-hard for some relatively simple election systems, in particular almost all nontrivial weighted scoring protocols [9], more recent papers have focused outside of worst-case complexity. A notable result [18] shows that for some election systems, particularly the veto system and other systems for three candidates, instances where it is easy to find a manipulation, or demonstrate none exists, are very rare. This result was shown using the known algorithms and heuristics for Partition, an NP-hard problem closely related to manipulating weighted three-candidate election systems. We make a definition in the next section.

2.3 Set Partition

A primitive set partition problem is Two-Way Partition, more simply known as Partition, which is NP-complete. We give the decision version of the problem as follows.

Name: Partition[11] (See also [7])

Instance: A multi-set of positive integers $S = \{s_1, \ldots, s_n\}$.

Question: Is there a subset $A \subseteq S$ such that $\sum A = \sum (S - A)$?

In the optimization version of Partition, we wish to minimize the maximum of the two subsets, namely, $\texttt{Max}(\sum A, \sum (S - A))$. Several heuristics exist to approximate this figure.

We give an example of the connection between election manipulation of three-candidate veto election systems and Partition given in Walsh as follows.

Consider a three-candidate veto election over the candidates p, c_1, and c_2, in which we wish for p to win. Suppose that initially, five voters, of weights 10, 8, 6, 4, and 2 veto p, c_1, p, c_2, and c_2 respectively. In addition, our coalition consists of four voters of weights 6, 5, 4, and 3.

Without loss of generality, none of the four manipulators will veto p, and in this example, we must distribute vetoes of weights $\{6, 5, 4, 3\}$ among candidates c_1 and c_2, currently with vetoes of total weight 8 and 6, such that each receives vetoes of weight totaling at least 16.

Since $8 - 6 = 2$, this corresponds to a partitioning of the elements $\{6, 5, 4, 3, 2\}$ such that each side has weight at most 10 (or equally, at least 10).

For this application, as well as many others, Partition has been extended to that of k-Way Partition, in which the goal is to divide the set into k equal subsets. It is noted in [14] that there are at least three optimization functions of interest: we may wish to minimize the maximum subset sum, maximize the minimum subset sum, or minimize the maximum difference between the sums of each two subsets. It is further demonstrated that all three of these optimization functions can produce different optimal partitions. The first two functions are of interest in our problem. The first optimization is interesting because we want the maximum score given to the non-distinguished candidates not to exceed the final score of our distinguished candidate. The second optimization is of interest in cases where vetoes are counted. We define the problem and these two optimizations as follows.

Name: k-Way Partition[11] (See also [7])

Instance: A multi-set of positive integers $S = \{s_1, \ldots, s_n\}$.

Question (decision): Are there disjoint and covering subsets $S = A_1 \cup \cdots \cup A_k$ such that $\sum A_1 = \cdots = \sum A_k$?

Question (optimization #1): Find disjoint and covering subsets $S = A_1 \cup \cdots \cup A_k$ that minimizes

$\texttt{Max}(\sum A_1, \ldots, \sum A_k)$.

Question (optimization #2): Find disjoint and covering subsets $S = A_1 \cup \cdots \cup A_k$ that maximizes

$\texttt{Min}(\sum A_1, \ldots, \sum A_k)$.

In the coming sections, we will describe the algorithms for the first optimization. The algorithms for the second optimization follow symmetrically with some minor adjustments.

Although it is mentioned in [18] that manipulation of veto elections of more than three candidates can be resolved as problems of multi-way partition, some adjustments must be made to this problem. We give an example and demonstrate the adjustments as follows.

Consider a four-candidate veto election over the candidates p, c_1, c_2, and c_3, in which we wish for p to win. Suppose that initially, four voters, of weights 20, 12, 9, and 7 veto p, c_1, c_2, and c_3 respectively. In addition, our coalition consists of six voters of weights 10, 8, 4, 4, 3, and 3.

In this example, we must distribute vetoes of weights $\{10, 8, 4, 4, 3, 3\}$ among candidates c_1, c_2, and c_3, currently with vetoes of total weight 12, 9, and 7, such that each receives vetoes of weight totaling at least 20.

Since $12 - 9 = 3$ and $12 - 7 = 5$, we are interested in partitions of the set $\{10, 8, 4, 4, 3, 3\} \cup \{3, 5\}$. However, in this case, we are interested in partitions in which 3 and 5 are not placed in the same subset, as these two subsets represent the vetoes given to candidates c_2 and c_3, respectively. This requires some adjustments to the partition problem and algorithms of interest, which our work in later sections encompasses.

Furthermore, as k-Way Partition problems partition only individual numbers, the application of this problem and its algorithms in that of manipulation of scoring protocols is limited to cases of plurality and veto, in which each vote is determined by a single candidate. For general scoring protocols such as Borda, we will need to introduce analogous partition problems.

2.4 Algorithms of Partition and k-Way Partition

Because Partition is NP-complete, several heuristics have been developed to approximate the best partition. Two heuristics of common use are the greedy method, which first sorts the elements of S in non-ascending order, and places each element in the set that minimizes the difference iteratively, and the Karmarkar-Karp heuristic [10], also known as the differencing heuristic, which decides that the two largest elements are in different sets, but defers deciding in which set each element is placed. Both of these heuristics can be modified into pruned exhaustive searches [12].

We give an example of the Karmarkar-Karp heuristic as follows. We are given the multi-set $\{6, 4, 3, 3, 2, 2\}$, which we wish to partition. We place 6 and 4 in opposite subsets. By inserting these two elements in opposite subsets, we effectively create a new element equal to the difference of the two largest elements, since we are only concerned about the total difference. We thus are creating the new element of 2, resulting in multi-set $\{3, 3, 2, 2, 2\}$. We then place each 3 in different subsets, and two elements of 2 in different subsets, resulting in $\{2\}$. In the base case of a single element, we must place it in one of the two subsets. In this case, this algorithm gives a partition of difference 2. Note that in this case the optimal partition has a difference of 0, as $6 + 4 = 3 + 3 + 2 + 2$, and this algorithm is not optimal.

Both of these heuristics can be extended to that of a complete algorithm using a depth-first tree search. The construction of the Complete Karmarkar-Karp algorithm is given in [12], which we briefly review. We note that in any given instance of Two-Way Partition, the two largest elements may be in different subsets or the same subset. By the heuristic, we always try the former first, and terminate if the partition is perfect, or within our desired maximum. The search is also pruned if the first element is greater than or equal to the sum of the remaining elements, as the best partition places the first element in one subset and the remaining elements in the other.

An early result by Korf [12] showed how the greedy and Karmarkar-Karp heuristics, as well as the corresponding complete algorithms, can be extended to the case of k-Way Partition. In the case of the greedy algorithm, we search a k-ary tree in which we try inserting elements into each of the k subsets. In the corresponding CKK algorithm, we will need to try each combination of the largest two tuples, of which there are $k!$.

In [13], the runtime of the greedy and CKK heuristics are evaluated k-Way Partition, and two new algorithms of a different type, which utilizes CKK for 2-Way Partition recursively, are introduced. Interestingly, while CKK is still more efficient than the greedy heuristic for 3-Way Partition, the greedy heuristic has a better runtime for k-Way Partition for $k \geq 4$, due to the $k!$-ary search tree.

Two new algorithms for k-Way Partition, Sequential Number Partitioning (SNP) and Recursive Number Partitioning (RNP) were also introduced in [13]. In Sequential Number Partitioning, one complete subset is first chosen, and the remaining unpartitioned numbers are partitioned recursively using this algorithm for $(k-1)$-Way Partition. In the base case, 2-Way Partition is evaluated using the CKK algorithm. The subsets are generated by inclusion-exclusion tree search with various pruning techniques.

There are several pruning techniques for the search of this first complete subset. First, to avoid symmetry, the first subset sum of this subset is restricted to be no more than $\lfloor \frac{t}{k} \rfloor$, where t is the sum of the elements, and the subsets are subsequently chosen in non-descending order by sum. Also, if m is the maximum subset sum of the best partition found thus far, or our target maximum, we restrict the first subset sum to be at least $t - (k-1)m$, as otherwise the best partition found utilizing this first subset sum cannot beat this difference. To induce as much pruning as early as possible in the search tree, the elements are considered in non-ascending order, and we try including first.

In Recursive Number Partitioning, on an instance of k-Way Partition for k even, the set is first divided into two subsets, each of which will be partitioned $\frac{k}{2}$ ways, by a top-level CKK search. In [13, 14], it is shown that both SNP and RNP show a marked improvement over CKK for k-Way Partition when $k \geq 3$, with RNP significantly faster than SNP for $k \geq 4$.

We wish to introduce a further extension of Partition in order to investigate the complexity of manipulating a general scoring system of weighted voters. In this paper, we demonstrate a natural extension of the above algorithms to this problem and also how to apply this new problem to the problem of manipulation in scoring systems. In doing so, we wish to gain a better understanding of the complexity of such manipulation problems in relation to the current best known algorithms of such partition-type problems.

3. EXTENSIONS OF k-WAY PARTITION

In our new problem of k-Way Permutation Partition, we are given a multiset of tuples, each of cardinality k. We wish to find permutations of each tuple such that, if we take the sum of each of the k positions among the tuples, the k sums are equal. As in the case of k-Way Partition, there are a few interesting optimization functions, of which two are of interest to manipulation problems. We give a formal definition as follows.

Name: k-Way Permutation Partition

Instance: A multi-set of k-tuples, $S = \{x_1, \ldots, x_n\}$, where $x_i = \{x_i^1, \ldots, x_i^k\}$. It can be assumed without loss of generality that $x_i^k = 0$ for all $1 \leq i \leq n$, as one may normalize the tuple, noting that only the difference among tuple elements is crucial.

Question (decision): Is there a mapping $P : \{1, \ldots, n\} \rightarrow S_{\{1,\ldots,k\}}$[1] such that

$$\sum_{1 \leq r \leq n} x_r^{P(r)_1} = \cdots = \sum_{1 \leq r \leq n} x_r^{P(r)_k}?$$

Question (optimization #1): Find the mapping $P : \{1, \ldots, n\} \rightarrow S_{\{1,\ldots,k\}}$ that minimizes

$$\texttt{Max}_{1 \leq i \leq k} \sum_{1 \leq r \leq n} x_r^{P(r)_i}.$$

Question (optimization #2): Find the mapping $P : \{1, \ldots, n\} \rightarrow S_{\{1,\ldots,k\}}$ that maximizes

$$\texttt{Min}_{1 \leq i \leq k} \sum_{1 \leq r \leq n} x_r^{P(r)_i}.$$

Note that k-Way Partition is a special case of this problem, in which each k-tuple has the form $(x_i^1, 0, \ldots, 0)$.

In this paper, we will examine extensions of algorithms introduced by Korf in [12, 13, 14] from Partition and k-Way Partition to that of k-Way Permutation Partition, their relationship to that of more general scoring systems, the distribution of the complexity of instances in evaluating such using these algorithms, and thus the frequency of hard instances of this problem in a system of uniform voters. Using these algorithms, we wish to show whether and when the results of [18] can be extended to that of other scoring systems using these heuristics.

In the rest of this section, we examine how each of the known algorithms for k-Way Partition may be extended to that of k-Way Permutation Partition. Such algorithms allow us to examine the complexity of manipulating some scoring protocols.

4. K-WAY PERMUTATION PARTITION AS A RESTRICTED K-WAY PARTITION PROBLEM

Our extensions of the SNP and RNP algorithms to that of k-Way Permutation Partition stem from the observation that k-Way Permutation Partition is a restricted version of the k-Way Partition problem. More specifically, given tuples $S = \{x_1 = \{x_1^1, \ldots, x_1^k\}, \ldots, x_n = \{x_n^1, \ldots, x_n^k\}\}$, we wish to find a k-Way Partition of the multiset of the union of all elements, $\{x_1^1, \ldots, x_1^k, \ldots, x_n^1, \ldots, x_n^k\}$, excluding zeros, with the additional constraint that each of the k subsets contains at most one element from each of the n tuples. As both the SNP and RNP algorithm work in a divide-and-conquer manner in which subpartitions are generated in sequence, this constraint can be implemented by restrictions in branching in each node of the search tree. We give a brief description of how these restrictions are implemented.

[1] S_A is the set of all permutations over A.

In the SNP algorithm, we wish to choose a set of elements for our first subset. As in k-Way Partition, we consider the elements in non-ascending order in an inclusion-exclusion search tree. In the algorithm for k-Way Permutation Partition, each element is labeled with its corresponding tuple, and in the top-level inclusion-exclusion search, if an element from this tuple has already been chosen, we must exclude this element from the set. On the other hand, if this is the last element from its tuple in the elements to be considered, we must include it. We then partition the remaining elements $k-1$ ways, normalizing the tuples if necessary. Note that since zero elements are excluded, this algorithm is also consistent with the SNP algorithm given by Korf.

There also exists a similarly well-defined extension of the RNP algorithm proposed by Korf. In our extension of RNP on k-Way Permutation Partition, we perform the CKK algorithm on the elements, which are labeled with their corresponding tuples, under the constraint that each of the two subsets receive no more than $\frac{k}{2}$ elements from each tuple. As each of the two branches in a CKK search node entail combining two subpartitions, we achieve this by refraining from combining subpartitions in which one or both subsets exceeds $\frac{k}{2}$ elements from any tuple. There also exists similar simple pruning techniques, that exclude search nodes in which we cannot attain a subset minimum we used in pruning, or that cannot beat the best partition found thus far.

In both algorithms, the base case of 2-Way Permutation Partition can be solved as a case of 2-Way Partition, as we may normalize so that there is only one non-zero element in each 2-tuple.

We consider an example as follows: partition the tuples $\{\{8,0,0,0\},\{6,2,0,0\},\{5,5,2,0\},\{5,5,0,0\},\{5,4,3,0\}\}$ optimally. In this algorithm we sort the union of the elements of each tuple, excluding zero, arriving at the multiset $\{8_1, 6_2, 5_3, 5_3, 5_4, 5_4, 5_5, 4_5, 3_5, 2_2, 2_3\}$. We note the tuple each element originates from as a subscript. The first branch of the top-level CKK search combines elements 8_1 and 6_2, adding a new element of net weight 2 to this set. Since we are interested in the further partitioning of the 2-Way Partition of this set, we must keep track of how we arrive at our net elements. We could represent this resulting multiset as follows: $\{5_3, 5_3, 5_4, 5_4, 5_5, 4_5, 3_5, 2 = 8_1 - 6_2, 2_2, 2_3\}$. Skipping a few steps, eventually this top-level CKK algorithm produces a partition of sets $\{6_2, 5_3, 5_4, 4_5, 3_5, 2_3\}$ and $\{8_1, 5_3, 5_4, 5_5, 2_2\}$ at its first left-hand-side leaf. This corresponds to two instances of 2-Way Permutation Partition: $\{\{0,0\},\{6,0\},\{5,2\},\{5,0\},\{4,3\}\}$ and $\{\{8,0\},\{2,0\},\{5,0\},\{5,0\},\{5,0\}\}$. In this case, since we are interested in 4-Way Permutation Partition, we evaluate two instances of 2-Way Permutation Partition for each leaf of the top-level CKK instance.

We give some pseudocode for an implementation of such an algorithm. In our code below, the $\texttt{RNP}_{\texttt{inner}}$ function takes a multi-set of weights, each associated with a difference between the sums of two sets of elements. In each of these two sets of elements, each element is associated with the tuple it originated from. For example, in the example above, in the first node, we formed an element 2, which was derived from the difference of element 8 and 6 from tuples 1 and 2 respectively. In the code below, recall that we are minimizing the maximum subset sum. We store this figure in the variable `best`.

$\mathbf{RNP_{inner}}(S = \{(s_1, A_1, B_1), (s_2, A_2, B_2), \ldots, (s_n, A_n, B_n)\}, k)$, where $s_1 \geq \cdots \geq s_n$, $s_i = \sum A_i - \sum B_i$. A_i and B_i are multisets of elements, each element labeled with the tuple it originated from.

if $n = 1$ [Base case.]
  [Recursively partition the tuples corresponding to sets A_1 and B_1 $\frac{k}{2}$ ways, after normalizing.]
  return `Max(RNP(Normalize(`A_1`)),RNP(Normalize(`B_1`)))`

else
  [Try difference if possible.]
  if each of $A_1 \cup B_2$ and $B_1 \cup A_2$ sum to at most $k($`best`$-1)$ and contain at most $\frac{k}{2}$ elements from each tuple
    `best` $\leftarrow$ `Min(best,RNP`$_{\texttt{inner}}$`({`
      $(s_1 - s_2, A_1 \cup B_2, B_1 \cup A_2), \ldots,$
      $(s_n, A_n, B_n)\}, k))$

  [Try sum if possible.]
  if each of $A_1 \cup A_2$ and $B_1 \cup B_2$ sum to at most $k($`best`$-1)$ and contain at most $\frac{k}{2}$ elements from each tuple
    `best` $\leftarrow$ `Min(best,RNP`$_{\texttt{inner}}$`({`
      $(s_1 + s_2, A_1 \cup A_2, B_1 \cup B_2), \ldots,$
      $(s_n, A_n, B_n)\}, k))$

  return best

At the top level, we break the tuples into individual elements, with corresponding trivial element sets.

$\mathbf{RNP}(S = \{(s_1^1, \ldots, s_k^1), \ldots, (s_1^m, \ldots, s_k^m)\})$

  `best` $\leftarrow \infty$
  [Each element by itself, labeled with its originating tuple.]
  return `RNP`$_{\texttt{inner}}(\{(s_1^1, \{(s_1^1, 1)\}, \emptyset), \ldots,$
    $(s_k^m, \{(s_k^m, m)\}, \emptyset)\}, k)$

The main difference between the implementation of RNP for this restricted RNP problem and that of Korf's implementation for k-Way Partition is the extra restriction for combining weights, as the subsets involved must not exceed $\frac{k}{2}$ elements from any tuple. The SNP algorithm can be implemented.

5. MANIPULATION AS A PARTITION PROBLEM

We resolve manipulation in scoring systems into instances of k-Way Permutation Partition. There are two relevant cases.

THEOREM 1. *Consider an election of the scoring system $(\alpha_1, \ldots, \alpha_k)$ and candidates $c_1, \ldots, c_k$ with initial scores $s_1, \ldots, s_k$. Let the set of manipulative voters be V' of cardinality $||V'|| = m$ with weights $w_1, \ldots, w_m$.*

There exists a manipulation ensuring the victory of c_1 iff there exists a $(k-1)$-way permutation partition of the tuples $\{\{s_2, \ldots, s_k\}\} \cup \{\{w_1\alpha_2, \ldots, w_1\alpha_k\}, \ldots, \{w_m\alpha_2, \ldots, w_m\alpha_k\}\}$ with maximum subset sum of at most $s_1 + \alpha_1 \sum_{1 \leq i \leq m} w_i$.

It should be noted that in order to apply this reduction, it is only necessary that the each voter's contribution to the scores of each candidate depend only upon his or her position in the voter's preference ordering. It is not necessary for the system to have a fixed scoring vector. Such elections may or may not be of interest to computational social choice theory, and will not be evaluated in this paper.

PROOF. Without loss of generality, each voter from V' will choose c_1 as his or her first preference, giving it a final score of $s_1 + \alpha_1 \sum_{1 \leq i \leq m} w_i$. To ensure the victory of c_1, none of the final scores of candidates $c_2, \ldots, c_k$ can exceed this figure. As each voter of weight w_i is free to choose a permutation of scores $w_i\alpha_2, \ldots, w_i\alpha_k$ for candidates $c_2, \ldots, c_k$, this corresponds to the problem of k-Way Permutation Partition containing a vector containing these scores for each voter in V', as well as a vector representing the current scores of the $k-1$ non-distinguished candidates.

A manipulation ensuring the victory of c_1 thus exists iff a such a $(k-1)$-way permutation partition not exceeding the final score of c_1 mentioned earlier exists. □

THEOREM 2. *Consider a q-veto election of candidates $c_1, \ldots, c_k$ with initial scores $s_1, \ldots, s_k$. Let the set of manipulative voters be V' of cardinality $||V'|| = m$ with weights $w_1, \ldots, w_m$. Without loss of generality, suppose $s_k \geq s_2, \ldots, s_{k-1}$.*

There exists a manipulation ensuring the victory of c_1 iff there exists a $(k-1)$-way permutation partition of the tuples $\{\{s_k - s_2, \ldots, s_k - s_k\}\} \cup \{\{\underbrace{w_1, \ldots, w_1}_{q}, 0, \ldots, 0\}, \ldots, \{\underbrace{w_m, \ldots, w_m}_{q}, 0, \ldots, 0\}$ with minimum subset sum of at least $s_k - s_1$.

PROOF. In this case, we are counting the total weight of the vetoes each non-distinguished candidates receives. Without loss of generality, no vetoes are given to the distinguished candidate. With some modification, this reduction also applies to cases of scoring protocols of the form $(\underbrace{\alpha, \ldots, \alpha}_{k-q}, \alpha_{k-q+1}, \ldots, \alpha_k)$ for $\alpha > \alpha_{k-q+1} \geq \ldots \geq \alpha_k$.

□

6. EXPERIMENTAL RESULTS

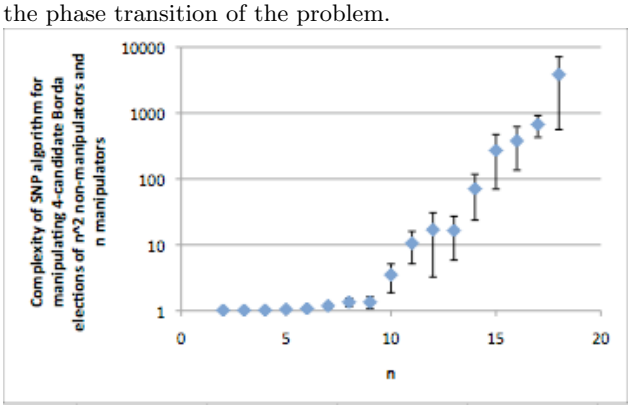
We test the feasibility of using algorithms of k-Way Permutation Partition for solving manipulation problems of various election systems using the connections in the previous section. To eliminate the issues of computer architecture and focus on the algorithm computationally, in each case, as a benchmark, we count the number of branches evaluated in invoking the algorithms in question, as opposed to runtime, for cases of interest. We also evaluate the standard error to provide 95% confidence intervals on the experimental data, to ensure statistical significance.

We choose voter weights uniformly from the range [0, 65536), and voter preference orderings uniformly as a random permutation. This is consistent with the definition of random voters in [18].

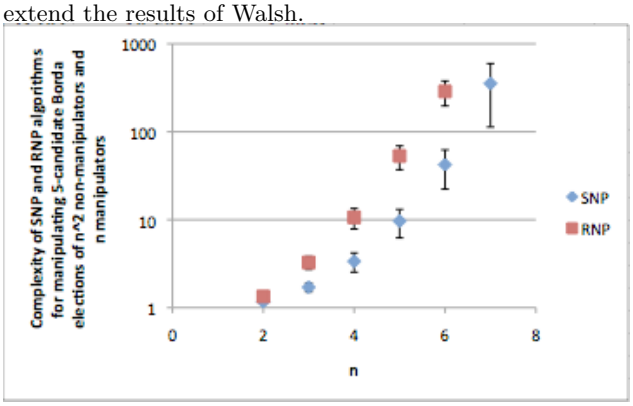
Our first test is for the case of manipulating weighted 2-approval and veto elections of a fixed number, k, of candidates, as the $(k-1)$-Way Permutation Problem instances for these cases are relatively similar to that of $(k-1)$-Way Partition. We found that, for these simple cases, the runtime

is similar to that of the 3-candidate cases tested by Walsh, that most cases can be solved in an average of about one branch. Particularly of interest, hard instances are not directly related to the phase transition of this problem. However, these results require the use of $(k-1)$-Way Permutation Partition, due to the arbitrary nature of the initial scores. Due to the nature of the reductions involved, we presume that this result also applies to scoring protocols of the form $(\alpha_1, \alpha_2, 0, \ldots, 0)$ for $\alpha_1 > \alpha_2$.

We test the case of 4-candidate Borda elections, which is the simplest case of a scoring protocol not of the form mentioned above. As with all of the plots in this section, we plot the 95% confidence interval of the mean of our testing. As demonstrated in [18] for elections of three candidates, the phase-transition of this problem also occurs for $m = cn^2$ for some constant $c > 0$. In the following chart, we plot the runtime of invoking the SNP algorithm for an instances of manipulating 4-candidate Borda elections of $m = n^2$ non-manipulative voters and n manipulators, as this is within the phase transition of the problem.

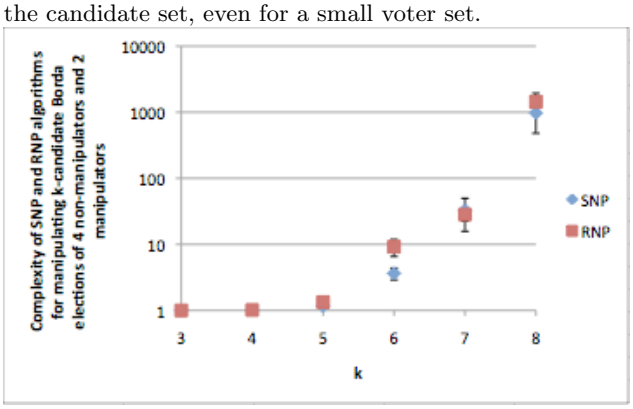


As we can see in this plot, the runtime of evaluating uniformly random instances of manipulating weighted 4-candidate Borda elections soars exponentially for instances near the phase transition of this problem. This result is in direct contrast to the results in Walsh for 3-candidate elections. Our next test case involves 5-candidate Borda elections, for which we have a choice between the SNP and RNP algorithms. As seen below, we observe that the RNP algorithm is significantly slower than that of SNP, contrary to the results in Korf in the general setting. Neither algorithms extend the results of Walsh.

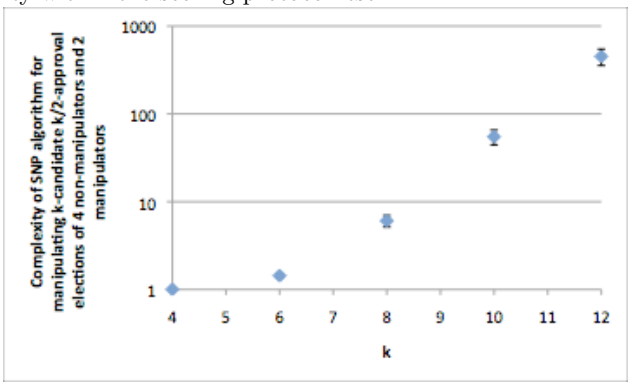


Our next two tests involves scaling the number of candidates in the election. We wish to investigate how the runtime complexity of the algorithms in question scale with the number of candidates in the election.

Below, we are plotting the runtimes of our algorithms on the case of manipulating k-candidate Borda elections for a fixed number of non-manipulators and manipulators, 4 and 2 respectively. The exponential results demonstrate that the SNP and RNP algorithms do not scale well with the size of the candidate set, even for a small voter set.



In our last test case below, we test a simplier family of scoring protocols, that of k-candidate $\frac{k}{2}$-approval elections, using the SNP algorithm (the runtime of the RNP algorithm is similar). This case is of interest as it is the most complex approval case for our algorithms, as one may choose to partition approvals or vetoes. The exponential nature of these results also show that the runtime of these algorithms stem from the number of candidates, as opposed to the complexity within the scoring protocol itself.



7. CONCLUSIONS AND FUTURE WORK

These new algorithms show how the problems of k-Way Partition and the known algorithms of such can be applied to the seemingly unrelated problem of manipulating weighted scoring systems. These new findings allow one to develop a better understanding of the inner workings of these manipulation problems. Known algorithms of partition-type problems, including Sequential Number Partitioning (SNP) and Recursive Number Partitioning (RNP) have the property of finding a good partition relatively quickly when one exists, and this property appears to extend to that of our new problems. Although RNP has been demonstrated to be faster than SNP for the special case of k-Way Partition, this simplification does not appear to extend more generally here, particularly for more complex systems such as Borda.

In each case, the best-known algorithms for Korf, and very natural extensions and generalizations thereof, do not validate the conclusions of Walsh and Nisan for elections of more than three candidates, but instead show that scoring protocols in general are not directly vulnerable to manipu-

lation via the known algorithms of partition-like problems. However, this may be due to either the strength of elections having more than three candidates, or weakness of the best-known algorithms for k-Way Partitioning and the extensions given in this paper. We leave this as an open problem.

As an open problem, we note that there are also inherent weaknesses of the known algorithms of partition. The obvious weakness of RNP, for instance, is that it can only be applied to cases of k-Way Permutation Partition for k even. This weakness is demonstrated clearly in Korf, in which it is shown that 3-Way Partition is almost as slow as 4-Way Partition, and by extension, similar results hold for k-Way Permutation Partition. In the cases where RNP would otherwise be more efficient than SNP, it is not available as an option for k odd as the CKK algorithm used at the top-level division is designed to enumerate partitions which are very close to even.

A second weakness of RNP relates to the top-level CKK division, which divides the initial set of labeled elements into two. Recall that in the CKK algorithm, the algorithm tracks a list of 2-way partitions of subsets of the original set. If either subset in any of the 2-way partitions in such a list cannot be furthered partitioned $\frac{k}{2}$ ways in a way better than the best partition found so far, it is clearly fruitless to continue on this search branch, as the final 2-way partition cannot yield a better overall k-way partition. One way to capitalize on this insight is to perform the algorithm recursively on the instances of $\frac{k}{2}$-Way Permutation Partition at the search nodes of interest. Unfortunately, this will only prune the top-level search tree only if a better partition does not exist, and may be expensive, as the number of such nodes, absent pruning, may be exponential. A possible open problem of interest is thus a study of the tradeoffs in making such prunings, determining fast heuristics of when to perform this test within the tree.

Resolving these two weaknesses of the RNP algorithm can help explain the anomaly of why despite being faster than SNP in k-Way Partition [13, 14], it behaved unusually slow in general for k-Way Permutation Partition.

Another problem of interest involves using variations of these new algorithms for polynomial-time approximation algorithms, a problem studied in the context of different election systems in [3]. Since it is also known that the CKK algorithm, and possibly the extensions thereof, are especially fast at finding a good partition when one exists, approximation and probabilistic algorithms may exist to capitalize on the tradeoffs in performing an incomplete search. Interesting optimization and approximation functions may include minimizing the number of manipulators needed, or relaxing the restrictions of the manipulation problem. The problem Partition has a polynomial-time approximation scheme (PTAS), which may also be extended to some of the extensions used here.

8. ACKNOWLEDGEMENTS

We wish to offer our special thanks to Dr. E. Hemaspaandra and three anonymous referees for their input.

This work is supported in part by NSF grant IIS-0713061

9. REFERENCES

[1] M. Ballester and G. Haeringer. A characterization of single-peaked preferences. *UFAE and IAE Working Papers*, 2006.

[2] J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, pages 227–241, 1989.

[3] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Approximability of manipulating elections. *Proceedings of the* 23^{rd} *AAAI Conference*, pages 44–49, 2008.

[4] V. Conitzer, J. Lang, and T. Sandholm. When are elections with few candidates hard to manipulate? *Journal of the ACM, Volume 54, Issue 3, Article 14*, pages 1–33, 2002.

[5] C. Dwork, R. Numar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. *Proceedings of WWW10*, pages 613–622, 2001.

[6] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: societies with single-peaked preferences are more open to manipulation and control. *Proceedings of TARK*, pages 118–127, 2009.

[7] M. Garey and D. Johnson. Computers and intractability: A guide to the theory of NP-completeness. *W.H. Freeman and Company*, 1979.

[8] A. Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, pages 587–601, 1973.

[9] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, pages 73–83, 2007.

[10] K. Karmarkar and R. Karp. The differencing method of set partitioning. *Technical Report, Computer Science Division, University of California, Berkeley*, 1982.

[11] R. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

[12] R. Korf. From approximate to optimal solutions: A case study of number partitioning. *Proceedings of the* 14^{th} *IJCAI Conference*, pages 266–272, 1995.

[13] R. Korf. Multi-way number partitioning. *Proceedings of the* 28^{th} *IJCAI Conference*, 2009.

[14] R. Korf. Objective functions for multi-way number partitioning. *Third Annual Symposium on Combinatorial Search*, 2010.

[15] J. Pitt, L. Kamara, M. Sergot, and A. Artikis. Voting in multi-agent systems. *The Computer Journal, 10.1093/comjnl/bxh164*, 2006.

[16] M. Satterthwaite. Vote elicitation: Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory 10 (April 1975)*, pages 187–217, 1975.

[17] L. Wai and L. Ho. Rank aggregation for meta-search engines. *Proceedings of the* 13^{th} *international World Wide Web conference on Alternate track papers and posters*, pages 384–385, 2004.

[18] T. Walsh. Where are the really hard manipulation problems? the phase transition in manipulating the veto rule. *Proceedings of the* 28^{th} *IJCAI Conference*, pages 324–329, 2009.

Learning Agents

Using Iterated Reasoning to Predict Opponent Strategies

Michael Wunder
Rutgers University
mwunder@cs.rutgers.edu

Michael Kaisers
Maastricht University
michael.kaisers@maastrichtuniversity.nl

John Robert Yaros
Rutgers University
yaros@cs.rutgers.edu

Michael Littman
Rutgers University
mlittman@cs.rutgers.edu

ABSTRACT

The field of multiagent decision making is extending its tools from classical game theory by embracing reinforcement learning, statistical analysis, and opponent modeling. For example, behavioral economists conclude from experimental results that people act according to levels of reasoning that form a "cognitive hierarchy" of strategies, rather than merely following the hyper-rational Nash equilibrium solution concept. This paper expands this model of the iterative reasoning process by widening the notion of a level within the hierarchy from one single strategy to a distribution over strategies, leading to a more general framework of multiagent decision making. It provides a measure of sophistication for strategies and can serve as a guide for designing good strategies for multiagent games, drawing it's main strength from predicting opponent strategies.

We apply these lessons to the recently introduced Lemonade-stand Game, a simple setting that includes both collaborative and competitive elements, where an agent's score is critically dependent on its responsiveness to opponent behavior. The opening moves are significant to the end result and simple heuristics have achieved faster cooperation than intricate learning schemes. Using results from the past two real-world tournaments, we show how the submitted entries fit naturally into our model and explain why the top agents were successful.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

Keywords

Iterated reasoning, cognitive models, multiagent systems, POMDPs, repeated games

1. INTRODUCTION

In many domains where multiple strategic actors are present, it is becoming increasingly common to find computer programs in place of human decision-makers. Algorithmic trading [14], automated ad auctions [12], and botnets [7] are just a few examples of the multiagent problem that have emerged over the past decade. The key challenge of such settings is the deliberate unpredictability of other adaptive agents that can prevent the formation of reliable responses. On the other hand, if others are trying to predict us, there is an opportunity to discover the pattern by which they attempt to do so. Multiagent learning has been motivated by successes in machine learning and several branches of economics to answer the question of how computer agents should make decisions when multiple decision makers are present that may not have the same goals or incentives [13]. The task at hand is actually two separate but related tasks: to *predict* the behavior of other unknown players, and to *respond* in turn. Unlike the single agent case, here agent designers need to recognize that other modelers are changing and attempting to anticipate their agent's actions.

One popular approach to building intelligent agents is to apply reinforcement-learning techniques adapted from single agent environments. Often, for learning to make speedy progress, algorithm designers rely on assumptions about opponents that are not always explicit, and we would like to have a way to explore them and understand how they arise as they do. We might notice that multiagent learning in multiple round games raises similar questions to those of reinforcement learning. Players need to learn how to act in the long-run, how to escape from undesirable locally optimal outcomes, and they need to learn quickly. One difference is that the issue of time can have a big impact on the eventual result of a game with multiple agents, while learners in fixed single-agent environments will typically reach the same policy regardless of the pace of experience. For example, in a game where two players get a high reward for cooperating with each other at the expense of a third, it pays off to be one of the first two cooperators—the third agent may never achieve high reward.

Another option is to model opponent behavior directly, by using recursive modeling [10], Interactive POMDPs [9], or Networks of Influence Diagrams [8]. The famous RoShamBo game is one domain where recursive reasoning has demonstrated its relevance and applicability [1, 6]. The obstacle that disrupts progress in this area is that modeling can go on endlessly, as an agent forms ever more complicated models using simpler models as parts, often in a rather unstructured way. Behavioral economists address very similar issues from a slightly different angle. Using experiments on humans playing games, they have found a great deal of evidence that people use strategic reasoning to make decisions, but only up to a point. Indeed, this reasoning conforms to a well-defined cognitive hierarchy, or a related level-k model,

Cite as: Using Iterated Reasoning to Predict Opponent Strategies, Michael Wunder, Michael Kaisers, John Robert Yaros, Michael Littman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 593-600.

composed of levels of thinking [2]. This model can apply to games with two agents or larger population games.

One limitation of previous iterated best response models is that there is a tendency to pick an exact strategy to represent each level. Assuming that agents can be classified as one of a few ideal types is one option for modeling opponents, but does not capture the aspect that players can belong to multiple types. We address the opponent model selection problem by allowing for a distribution of agents to represent each level. The appeal of using distributions is that uncertainty over opponents leads to multiple best responses, and sometimes there is no principled way to choose from among them. We can use this feature to cover uncertainties about implementations or simplified approximate versions of optimal strategies, which play a role in bounded reasoning models. Our enriched model, called a Parameterized I-POMDP, highlights to the user the most important strategies of the game, and identifies their relationships. Given a feasible unknown strategy to test, the framework allows us to directly measure the amount of reasoning that lies behind it by comparing it to constructed strategies derived from a thorough reasoning process.

To illustrate this process, we utilize the recently introduced Lemonade-stand Game (LG) as an example setting where playing one's opponents is more important than playing the game. LG is played by three players, which is more complicated than the simplest 2-player case, but still small enough where the pairwise interactions are major factors. This simple game leads to an elegant analysis, even with the complications of triadic interaction. The main message of our framework is that learning agents can use a number of ways to plan against opponents, but in the end success depends mostly on the distribution of types in the population. The model guides theoretical analysis of the game and its application is demonstrated with actual agents from competitions. Such competitions have a history of focusing researchers on important issues and providing a wide selection of approaches that can be mined for data. This method of mining data to discover aspects of the underlying reasoning model is an exciting emerging branch of computer science [15, 16].

The next section, 2, provides more detail about existing models in both computer science and economics. In Section 3, we introduce our proposed extension to those earlier models. Section 4 explains the LG. Section 5 applies our framework to LG and derives the resulting levels, resulting in a hierarchy over the space of reasonable strategies. Section 6 uses previous tournament submissions as evidence that the new model works in some interesting types of games.

2. BACKGROUND

The cognitive hierarchy model (CH) [3] and its cousin, level-k thinking [4], have been used by behavioral economists to explain observed human behavior. CH consists of an initial level of base strategies combined with a series of levels found by repeatedly taking the best response of lower levels. The level-k model operates by responding to just level $k-1$ instead of levels $0, ..., k-1$. In these investigations, the games are generally simple enough that it is straightfoward to construct the hierarchy. Experimental data then provides knowledge about the frequencies of the various levels, and therefore properties like the average level in a population. CHs are useful in population games or 2-agent games alike, but usually they consist of one-shot experiments, obviating the need to build complex sequential models at each level. While we will primarily consider games played through computer agents and not directly by people, the same underlying process is present in both systems.

From the computer-science or machine-learning perspective, this setting has been formalized as an Interactive Partially Observable Markov Decision Process, or I-POMDP. This development synthesizes the considerable work done on single agent POMDPs with multiagent approaches such as the Recursive Modeling Method (RMM) [11]. This formulation is ideal for sequential or repeated games where unknown opponents have limited reasoning capabilities.

POMDPs are similar to the standard Markov Decision Process except it is not assumed that an agent knows what state it is currently in. A solver must use observations to infer the likely state by updating beliefs over the state space. An I-POMDP is a POMDP that has interactive states in place of states, and joint actions in place of actions [9]. This interactive state is the cross product of environmental states and internal states of agents present in the game. The interactive state space is constructed recursively starting where other agents are represented strictly as a stochastic part of the state. In other words, in the simplest interactive state other agents are assumed to have no reasoning capacity or sensitivity to payoffs, but instead exist as a noisy component of the environment. Then, we build more advanced interactive states in new I-POMDPs to represent further or higher opponent reasoning. We can use this technique to reach any level of sophistication that can be reasonably computed, but in practice only a finite number of nested levels are used.

Associate $\mathsf{I\text{-}POMDP}_i$ with agent i and the only other agent is j. The definition generalizes to more agents. An $\mathsf{I\text{-}POMDP}_i = \langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle$ has the following features:

- IS_i is the set of interactive states $IS_i = S \times \pi_j$ where S is the set of states from the environment and π_j is the set of policies for agent j.
- A is the set of joint actions $A_i \times A_j$.
- T_i is the transition function $T_i : S \times A \times S \to [0, 1]$. The transition model, along with the internal decisions for policy π_j, determine the next interactive state, but we assume that agent i does not directly control other agents in its environment.
- Ω_i is the set of observations.
- O_i is the observation function $O_i : S \times A \times \Omega \to [0, 1]$.
- R_i is the reward function $R_i : IS_i \times A \to R$.

Policies at each level k are derived from the beliefs $b_{j,k-1}$ over the policies and states of the previous level $k-1$. Define the following spaces.

- $IS_i^0 = S, \qquad \pi_j^0 = IS_i^0 \to A_j \in H_0$
- $IS_i^1 = IS_i^0 \times \pi_j^0, \qquad \pi_j^1 = b_{j,1}(IS_i^1) \to A_j \in H_1$
  .
  .
  .
- $IS_i^L = IS_i^{L-1} \times \pi_j^{L-1}, \qquad \pi_j^L = b_{j,L}(IS_i^L) \to A_j \in H_L$

The strength of this formalism is that it suggests a relatively general algorithm for computing policies and works well with good initial beliefs. A weakness is that the set of opponent policies to include is unspecified and therefore the solution breaks down when those beliefs do not match reality. We attempt to mitigate this flaw by keeping a range of solutions to represent our initial uncertainty. The framework also has the advantage that the solver has some flexibility in selecting the planning problem to attack, which could allow it to select simpler to reduce the computationally demands. Currently there is no predescribed way to achieve this aim, but it is another goal for our extended model.

3. PARAMETERIZED I-POMDPS

Our framework, entitled Parameterized Interactive Partially Observable Markov Decision Processes (PI-POMDPs) is a model for recursively deriving a set of policies that respond to less advanced policies for use in highly structured domains. We extend the I-POMDP framework by building an entire profile of policies at each nested level in place of a single solution. Instead of a single policy, the solution will be the hierarchy of policies computed at each level.

Define for agent i rule-based policy $H : IS \rightarrow A_i$ to be a basic rule that maps states to actions. One example of a rule would be $A_i^t = A_i^{t-1}$, signifying constant action. Then, a parameterized policy $\pi : \mathbf{R} \rightarrow H$ maps real vector $X \in [0, 1]$ to some rule. X could indeed be used to represent any adjustable feature of an agent, but we will assume that X_r is the probability of playing rule H_r. Note the rule is not fixed for the whole game, but rechosen every time step.

The parameterization of policies begins right away when deciding which beliefs over initial strategies to start with. There may be several options that incorporate the idea of non-reasoning policies, so we end up with $\pi^0(X)$ for agents at level 0 to weight each tactic. In turn, following the I-POMDP mechanism, the $\pi_j(X)$s for each agent j and value X are used to construct an instance of a POMDP problem. Instead of optimizing over all X to arrive at a single policy, compute a range of policies as X changes. If possible, we will attempt to condense all of these rules into a single new parameterized policy π^1 with as few input dimensions as possible, to represent the result of a step of reasoning over level 0. This way, we do not have to make the decision about which strategies are valid for the next level derivation. All of them are kept as a part of the final model. While it is possible for games to take place in states in the environment, we will consider the partial observability to consist of uncertainty over the parameterized policies present in the agent's population.

4. LEMONADE-STAND GAME

Recently, the Lemonade-stand Game was introduced to demonstrate the interaction complexity that can arise in a game from simple rules [17]. The game is played by three lemonade vendors on a circular island with n beaches, where typically $n = 12$, arranged like the numbers on a clock. Each morning, the vendors have to set up on one of the beaches, not knowing where the other vendors will show up. Assuming the beach visitors are uniformly distributed and buy their lemonade from the closest vendor, the payoff for the day is equal to the distance to the neighboring lemonade vendors. For convenience denote $D(A_i, A_j)$ the distance function between agents i and j on the side with no other agent in between. Then, $R_0^t = \sum_{j=1}^{2} D(A_0^t, A_j^t)$.

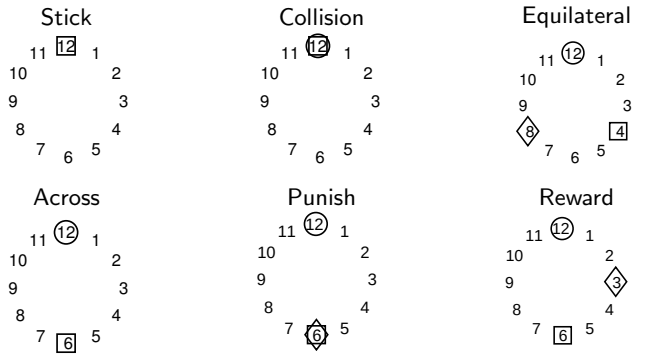


Figure 1: This figure depicts key strategic patterns of the lemonade game. Each of the six diagrams refers to a (partial) joint action, and similarly a strategic move by □, expecting opponents to play ◇ and ○. As the domain is on a ring, the patterns are rotation insensitive.

In game-theoretic terms, LG is a 12-action normal form game on a ring, where the payoff function equals the sum of distances to the right and left neighboring vendor. As a corollary, the cumulative payoff of the three players is 24. The only exceptional formations are when multiple agents conflict by choosing the same action (Collision). If two vendors choose the same action, they receive a reward of 6 and create the most favorable condition for the third agent who receives the maximum of 12. If all three vendors choose the same action, each receives 8. There is no special property about any of the 12 locations on Lemonade Island. The game is played repeatedly for T days and the joint action is observable. T is set to 100 so that agents can learn about the opponents' behavior from previous rounds.

The dynamics of this game are particularly interesting because it involves a sense of competition, as the gains of one always have to be compensated by the loss of others, as well as a sense of cooperation, because two agents can coordinate a joint attack on the third. Figure 1 shows an overview of the key strategic patterns in the LG. Each agent has to choose an action, and the simplest move is to stick with the initial action from then on (Stick). The Equilateral pattern splits the payoff evenly into 8 for each agent, but from worst case perspective is dominated by the cooperative action Across. Once two agents coordinate on the action Across, they will share 18, relegating the third agent to 6 regardless of the action it chooses. As an illustration of its simplicity, □ must only find a predictable player ○ and use the action opposite to it. ○ can be completely oblivious as long as it is predictable (say, a pure Stick player).

If an agent finds its opponents in a consistent Across pattern, it will lose unless it can entice at least one opponent to break formation. In a simple form illustrated in Figure 1, bottom right, ◇ can alternate between using the same action as □ and an action halfway between ○ and □. □ will get the same utility whether it is Across from ◇ or ○ during the Reward phase of ◇, but would choose Across from ◇ if it wants to avoid low utility during its looming Punish phase, essentially switching partners.

5. LEVELS OF REASONING IN LG

The Lemonade-stand game is an ideal example of competitive collaboration. That is, a player able to convince another player to cooperate with it can achieve a higher average score to the disadvantage of the third player. Of course, each player has to choose the "friendlier" player to cooperate with, with the knowledge that any attempts may be tracked by the other players. Ultimately, the two players who work together best will achieve the highest scores.

It appears that players have many repeated turns for observation and experimenting. In reality many matches are settled in the first several rounds, as agents seek partners and mutual history is established. Cooperation, however it is defined, is self-reinforcing. Therefore, strategies in this game put a premium on speed over data collection when finding optimal actions. This property means that traditional learning methods, like gradient ascent or regret matching tend to be outperformed by very simple rules. Because there are many possible Nash equilibria in the game, it is also unclear which ones are optimal and how to reach them. Our aim is a model that can explain such phenomena and yield strategies that at least outperform the simplest heuristics. An alternative approach [5] to this game identifies a stable equilibrium and classifies agents as leaders or followers according to who initiates the equilibrium pattern. While this strategy works in some scenarios, in some cases it is possible to identify several levels of leading and following. It also makes no judgments about whether one is superior to the other, or how one might measure that performance.

5.1 Long-run Optimal Behavior

LG translates into our PI-POMDP model, with several simplifications. In this case, $\Omega \in A$, so that $O : IS \times A \times A \rightarrow [0, 1]$. In addition, there is only one state in the environment, which means there are only pseudo-states depending on the agents' behavior that is conditioned on the current A^t.

- IS_i is the set of interactive states $IS_i = S \times \pi_j$ where S is the set of states from the environment and π_j is the set of policies for agent j.
- S is defined by the time step in the finite horizon case.
- $A_i = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$.
- $H_{\textbf{Stick}}$: A basic action type that repeats the same action as the last turn
- $H_{\textbf{Uniform}}$: Play a random action.
- $\pi_j^0(X_0) = a \in \{H_{stick}, H_{uniform}\}$ s.t. $P(a = H_{stick}) = X_0, P(a = H_{uniform}) = 1 - X_0$.
- Ω_i is equivalent to $\bigcup A_j \forall_j$.
- O_i is determined by the policy of the opposite agent.
- R_i is the sum of distances to the players on either side.

To analyze this game rigorously using a PI-POMDP, we begin like most iterated reasoning models with a base level of non-reasoners, called Level 0 or $L0$. Analogously to an inductive proof, L0 forms the basis for the rest of the hierarchy. First, these base strategies are defined, and subsequently, the higher layers can be constructed by iteratively applying the reasoning step. Here, we define a step of reasoning to be a policy that maximizes the score against either a distribution over previous levels, or a selection of agents from those levels, by solving the POMDP formed by them. To avoid losing information, a parameterized policy (responding to a previous distribution of lower-level policies) represents the next level.

In many games, a base strategy of a single uniform distribution over all actions suffices: H_j^{Uniform}. In repeated games like LG, there exists another trivial action Stick, which leads to the basic notion of sticky strategies. Stickiness, as measured by the likelihood that a player remains in place, plays an important role in this game because it makes action prediction simple. As such, the rule $H_j^{\mathsf{Stick}} : A_j^t = A_j^{t-1}$ deserves a place among base strategies. In typical constant-sum games, a non-changing strategy is easy to defeat. In LG it is a powerful strategy on its own, as it forces other players to take action beneficial to the sticky player, such as to move away from it.

We take the general $\pi_j^0(X)$ base level L0 to be composed of H_j^{Uniform} and H_j^{Stick}, with a single real parameter X_0 to control the relative frequency of each. Consider the two opponents to be named B_k and C_k, where k is the amount of reasoning the agent's strategy contains. The solving agent's perspective is denoted agent I. The L0 strategy for B_0 is defined by an initial random action and the probability X_0 to Stick with the previous action in the following turns, or otherwise pick a new random action. Y_0 is the corresponding value for agent C_0. For $\pi_j^0(0) = H_j^{\mathsf{Uniform}}$ or $\pi_j^0(1) = H_j^{\mathsf{Stick}}$, L0 takes the form of a uniformly random (L0-U) or constant strategy (L0-C) respectively. We refer to this policy π_j^0 as $\pi_j^{\mathsf{Semi\text{-}random}}(X)$. Define $\hat{X}$ as the current estimate of X for an opponent B_0, and $\hat{Y}$ as the estimate of Y for opponent C_0, signifying the same strategy. In other words, given a sequence of observations from two unknown strategies implementing π^0 with values X and Y, we must make statistical conclusions about those values given our experience. We are then faced with finding a long-run strategy given observations $\hat{X}$ and $\hat{Y}$. Although a POMDP solver could be utilized to simulate the reasoning, here proofs are presented because the steps are very open to analysis.

THEOREM 1. *The optimal L1 strategy for agent I_1, π_I^1, is to maximize the distance D from the other two agents, giving $H_I^{\mathsf{Between\text{-}across}} = W_B \mathsf{Across}(A_B) + W_C \mathsf{Across}(A_C)$ where $W_B = \frac{X(1-Y)}{X(1-Y)+Y(1-X)}$ and $W_C = 1 - W_B$ are weights that determine how much $H_I^{\mathsf{Between\text{-}across}}$ should favor each of the Across actions. This strategy will prefer to be Across from the player who Sticks more often.*

PROOF. Since L0 agents do not respond to the actions of agent I_1, the POMDP reduces to a simple MDP. That is, action A_I has no effect on the transitions of the opponents, so the best action is found by calculating the expected utility of each spot, given X, Y, and the current placement of B_0 and C_0. If C_0 Sticks at location 0 and B_0 is random over all locations, the expected value of action a for $a > 0$ is:

$$V(a) = \frac{6}{12} + \frac{12}{12} + \frac{\max(0, a-1)}{12}\left(12 - \frac{a}{2}\right) + \frac{11-a}{12}\left(6 + \frac{a}{2}\right).$$

The first term is the event that B_0 lands on I_1. The second term is the event that B_0 lands on C_0. The third term is the event that B_0 lands in the short distance between C_0 and I_0, and the fourth term is the event of landing on the

large distance side. Taking the derivative, we then find that (when $n > 0$)

$$\begin{aligned} V(a) &= 6 + a - \frac{a^2}{12} \\ V'(a) &= 1 - \frac{a}{6} = 0 \\ a &= 6. \end{aligned}$$

The optimal action is 6, directly across from 0, where the expected value is $V(a) = 6+6-\frac{6^2}{12} = 12-3 = 9$. All actions on the far side of B_0 and C_0 have the same value when both players Stick or act uniformly, so we just care about the case when one of the two switches. Assume that B_0 is at 0 and C_0 is D_{BC} spaces away clockwise, where $D_{BC} \leq 6$ w.l.o.g. The value of action a when C_0 Sticks and B_0 is random is

$$\begin{aligned} V(a) &= 6 + a - D_{BC} - \frac{(a - D_{BC})^2}{12} \\ V'(a) &= 1 - \frac{a - D_{BC}}{6}. \end{aligned}$$

Therefore, the marginal value when B_0 Sticks with probability X and C_0 Sticks with probability Y is

$$V'(a) = \left(1 - \frac{a}{6}\right) X(1-Y) + \left(1 - \frac{a - D_{BC}}{6}\right) Y(1-X) = 0$$

$$a = \frac{6X(1-Y) + (D_{BC}+6)Y(1-X)}{X(1-Y) + Y(1-X)}.$$

Intuitively, the first term of the numerator weights the position directly across from B_0, and the second term does the same for C_0. Therefore the optimal action depends on the relative values of X and Y. □

The implication of this theorem is that an L1 strategy is predisposed toward choosing the action across from the more stable player. In general an agent observing that $X = Y$ causes the agent to always maximize its distance to the closest agent. Of course, in the initial rounds of a game, there are not enough observations to accurately forecast these unknowns. There are various ways to implement this policy, from the method of estimating $\hat{X}$ and $\hat{Y}$ to its reliance on priors of $\hat{X}$ and $\hat{Y}$. Assume $\hat{X}_0 = \frac{X_1 + c_{\textsf{Stick}}}{2X_1 + c_{\textsf{Total}}}$ where $c_{\textsf{Stick}}$ is the number of Stick moves and $c_{\textsf{Total}}$ is the current time steps. Here, the new parameter $X_L \in [0, \infty]$ represents the degree of attachment to the prior $\hat{X}_0 = \frac{1}{2}$, such that X_L^{-1} is the learning rate at which this estimate converges to the true value. With few observations, $\hat{X}_0$ will be noisy for low X_L (high learning rate). A high learning rate implies that if one player is constant but the other moves, this strategy will move sharply across from the constant player. When both players have been constant, W is undefined because the random (or half-random) cases does not occur, and therefore in that case there are a range of optimal actions. Since another feasible implementation is to assume both players are constant until there is contrary evidence, there is certainly room for parameterizing the preferred response in this case. However, given asymmetric behaviors, the theorem holds, where the constant player is the preferred partner.

For L2, we are looking for the best strategy given some combination of the first two levels, which is partially observable in the PI-POMDP. L2 optimizes against a distribution of L0s and L1s. The new PI-POMDP is therefore distributed across these two levels, as well as the range of parameters. We have already solved the exclusive L0 case, which will determine the default L2 behavior unless something close to L1 is observed.

When examining the rest of this PI-POMDP, this new type adds two elements to the policy calculation, which again depend on the parameters of the policy. First, the move-away-from-closest-player factor, represented by a slower learning rate and strong commitment to equal priors, exerts an influence on future levels to move directly across from the other player. Second, the punish-movers factor makes this movement less rewarding.

THEOREM 2. *Against* $H_{B_1}^{\textsf{Between-across}}$ *and* $H_{C_1}^{\textsf{Between-across}}$*, the optimal rule for agent* I_2 *is* $H_{I_2}^{\textsf{Stick}}$*.*

PROOF. (Sketch) With two L1s B_1 and C_1, each L1 is trying to move away from its two opponents. L1 is continually estimating $\hat{X}$ and using the estimate to adapt its strategy, which is to follow across from the other two agents, according to relative stickiness. An optimal rule here is just $H_I^{\textsf{Stick}}$ because B_1 prefers to move Across(I_2) over a moving agent, which C_1 certainly is. This tendency means that whenever B_1 registers a move by C_1, it moves a little farther from I_2. This new move then registers as a move for C_1, which in turn updates its action, and so on. This repetitive rule may reach oscillations, but the net effect will be to maneuver away from I_2, to the benefit of I_2. This policy is correct across the range of X_L. □

The interesting case is when the L2 player is up against one L0 and one L1 because essentially π_{C_0} "leads" and π_{B_1} "follows". The asymmetry of strategies allows for a new rule to emerge. In effect, π_{B_1} is constructed to move away from the semi-random π_{C_0}, but also from our agent I_2. We can use this tendency to our advantage in the best response. B_1 tends to play $H_{B_1}^{\textsf{Across}}$ from the C_0 with weighting $W_C = \frac{\hat{Y}_0(1-\hat{Z}_0)}{\hat{Y}_0(1-\hat{Z}_0)+\hat{Z}_0(1-\hat{Y}_0)}$ where $\hat{Z}_0$ is the staying probability of I_2. Thus, in that case we would hope to keep $\hat{Z}_0$ greater than $\hat{Y}_0$.

THEOREM 3. *Against* $H_{B_1}^{\textsf{Between-across}}$ *and* $\pi_{C_0}^{\textsf{Semi-random}}(Y)$*, the optimal rule for agent* I_2 *is* $H_I^{\textsf{Stick}}$ *until the number of moves of* $\pi_C^{\textsf{Semi-random}}(Y)$ *reaches a certain threshold* m*, and then to either move* Across(C_0) *if* C_0 *is too close or* Stick *as* (B_1) *moves closer to* Across(I_2)*.*

PROOF. (Sketch) We will consider the extreme cases where $Y_0 = 0$ or $Y_0 = 1$, and $X_L = 0$ or $X_L = \infty$. If $Y_0 = 0$, then $W_C \to 0$ when $\hat{Z}_0 > 0$ and $B_1 \to$ Across(I_2), regardless of the value of X_L. To accelerate this beneficial response, I_2 needs to Stick. If $Y_0 = 1$, then $H_{B_1}^{\textsf{Between-across}}(X_L)$ depends on the value of X_L. As $X_L \to 0$, B_1 is very sensitive to differences in moving probability. $W_C \to 1$ when $Z_0 < 1$ and $B_1 \to$ Across(C_0). To prevent this harmful response, I_2 should Stick as much as possible, but recognizing that the location of C_0 matters. It is preferable that C_0 be far from I_2 since B_1 will make room for it. In the worst case, if I_2 moves Across(C_0) it gets a minimum reward of 6 and expects a reward of 9, so that action is optimal if the current configuration gives a lower score. As $X_L \to 1$, B_1 retains more committment to its priors and has a high affinity for moving exactly in between I and C, unless a large difference $(\hat{Z}_0 - \hat{Y}_0)$ accrues. Therefore it is safer in that case for I_2 to move Across(C_0) if the learning rate X_L^{-1} is small. Therefore, depending on the relative values of X_L and Y_0, it may be optimal either to Stick or Across. □

We should note here that L2 can only classify opponents in one of two ways. The special case is L1 behavior, which is confined to a window of actions generally across from the two opposing players. The default classification is L0, which is defined by some combination of constant action and uniform random action. The significance of this simple modeling is that L2 would classify itself as L0, albeit with a high staying probability. Because L1 and L2 both have a tendency to move Across from the stickier players given sufficient information, this property will be selected at future levels. In fact, as more reasoning is applied, optimal strategies will start as constant for longer and longer as they attempt to out-wait earlier types. In those cases where all players have been constant from the beginning of the game, the decision about when to move is determined by the cost of remaining in the same location combined with the degree of reasoning ascribed to the opponents. In this case, the higher strategies are discouraged from moving at all due to this tendency to punish moving players. We can therefore consider the parameter X_2 to mean probability of moving into an Across position, especially when the current position is suboptimal.

The iterated best-response methods employed here do not necessarily adhere to the principle of auto-compatibility, whereby players do well against copies of themselves. Evolutionary strategy selection would pursue this goal more closely. A game with two of the same agent and one that is different would take on a new focus, where other forms of cooperation may be attainable that involve breaking the simple delayed across-move found by interated best response.

6. EXPERIMENTS

The levels of LG, while useful, are theoretical constructs. Nonetheless, the basic elements of this account arose in a group of agents developed independently. This section shows the viability of the level-based analysis by applying it to the two rounds of open LG competitions, one in Dec. 2009 and the other in Dec. 2010. The submitted strategies were a diverse collection. No two were alike and ranged from complete uniform action to near constant, to Across-seeking and initiating, and many in between.

To apply the model to real agents, we would like to classify each strategy by level or as a hybrid between levels. If our PI-POMDP model is a good fit for LG, populations consisting of agents that correspond to a similar mix of levels should behave, and score, in roughly the same way as their idealized counterparts. Since each level has its unique strengths and weaknesses, performance depends on the makeup of the population and specifically the relative frequency of each level. For the purposes of this paper, we classify a strategy by inspecting how it scores against idealized strategies from each of the levels we identified. See Figure 2 and Tables 1 and 2, right hand side, for these estimated levels. We ran the submitted agents against strategies over various values for the relevant parameters, such as $X_0, X_1, X_2 \in [0, 0.5, 0.75, 0.9, 0.95, 1.0]$. Using the derived strategies as benchmarks to compare to, we take the squared difference between unknown agent and level representative, and find the smallest difference between two adjacent scorings, say Level 2.95 and 2.975.

The rankings of the players in both tournaments provide a rough correlation to the amount of reasoning. The bottom half of the 2009 performers act like the base assumption strategies. The top half behave like those derived in

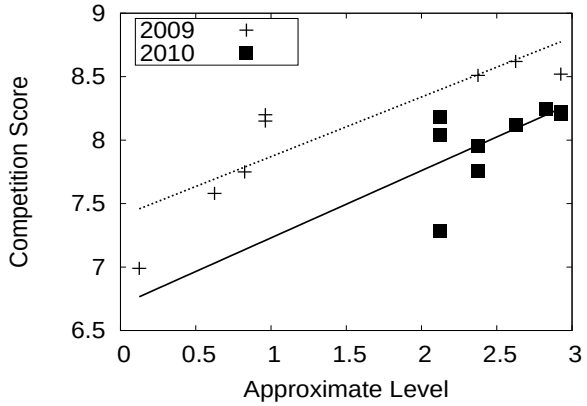


Figure 2: Estimated levels of competitors in two Lemonade-stand Game tournaments. Both sets of agents show positive correlation between reasoning and performance. R^2 values are 0.77 for 2009 and 0.34 for 2010. The more recent agents show a shift to higher reasoning levels, as well as a compression of scores.

the higher levels of the PI-POMDP model. From the 2010 dataset we find that on average reasoning has shifted up a level. Players identify the Across position as a goal state, but the top performers are more patient to get there, which implies more reasoning according to the model.

Another prediction of this model is that agents that perform too much reasoning do less well than those that go just beyond the average level of the population. We ran tournaments with both sets of agents and two additional agents drawn from the model population. As Tables 1 and 2 show, in the 2009 competition a strategy that is close to the Level 1 ideal (but modified for fast Across) outperforms the rest, while a higher level strategy at Level 2.975 only gets to the middle of the pack. In the 2010 population, this ordering is reversed. Note that winning the competition is, in a sense, easy. Given our analysis, the only missing information is a guess of the average reasoning level of the population. Nevertheless, without access to the complete set of submitted agents, identifying the appropriate reasoning level is a serious challenge.

7. CONCLUSION

This article introduced a PI-POMDP analysis for repeated games and applied it to the Lemonade-stand Game competition. In the competition, simple heuristics outperformed intricate learning schemes, suggesting that PI-POMDP or CH analysis might be preferable to domain-general best responses in strategic interactions. The Lemonade-stand Game rewards strategies that trade off patient exploration for speed and commitment. Those participants who opt for too much exploration over model-based responses suffer against more carefully optimized strategies. The model demonstrates that players must employ some basic heuristics in the early stages of a game. If they do not, they risk getting classified as the less responsive, consistent, or coop-

Table 1: 2009 LSG Tournament results including two agents inspired by the PI-POMDP hierarchy (italicized). The winners are in bold. Level 0.83 would correspond to a player that Sticks with probability of 0.83, but random the rest of the time. An agent that would qualify as Level 2.63 would mean that a player Sticks when in an advantageous starting position. When its initial spot is less beneficial than it is constant with probability equal to 0.63, and the rest of the time moves Across from another player, preferring the more constant one. In cases where it is already Across from a player, it remains in place by choosing the same action.

Rank	Strategy (Affiliation)	Score	Error	Level	Parameterized Level
1.	*PI-POMDP Level 1.0 modified (New addition)*	8.72	± 0.0071	$L1$	1.00
2.	**EA^2 (Southampton/Imperial)**	8.56	± 0.0069	$L2$	2.63
3.	CoOpp (Rutgers)	8.51	± 0.0055	$L2$	2.38
4.	ModifiedConstant (Pujara, Yahoo!)	8.48	± 0.0076	$L2$	2.93
5.	*PI-POMDP Level 2.975 (New addition)*	8.10	± 0.0083	$L2$	2.98
6.	Waugh (Carnegie Mellon)	8.00	± 0.0087	$L0$	0.96
7.	ACT-R (Carnegie Mellon)	7.88	± 0.0086	$L0$	0.96
8.	GreedyExpectedLaplace (Princeton)	7.43	± 0.0086	$L0$	0.83
9.	FrozenPontiac (U Michigan)	7.38	± 0.0075	$L0$	0.63
10.	Kuhlmann (U Texas Austin)	6.94	± 0.0054	$L0$	0.13

Table 2: 2010 LSG Tournament results including two agents inspired by the PI-POMDP hierarchy.

Rank	Strategy (Affiliation)	Score	Error	Level	Parameterized Level
1.	*PI-POMDP Level 2.975 (New addition)*	8.30	± 0.0099	$L2$	2.98
2.	**TeamUP (Southampton/Imperial)**	8.25	± 0.0099	$L2$	2.83
3.	Waugh (Carnegie Mellon)	8.19	± 0.0094	$L2$	2.93
4.	ModifiedConstant (Pujara, Yahoo!)	8.17	±0.0097	$L2$	2.93
5.	Matchmate (GA Tech)	8.15	±0.0095	$L2$	2.13
6.	Shamooshak (Alberta)	8.10	±0.0094	$L2$	2.25
7.	GoffBot (Brown)	7.97	±0.0108	$L2$	2.13
8.	Collaborator (Rutgers)	7.95	±0.0105	$L2$	2.38
9.	Meta (Carnegie Mellon)	7.80	±0.0102	$L2$	2.38
10.	*PI-POMDP Level 1.0 modified (New addition)*	7.80	±0.0096	$L1$	1.00
11.	Cactusade (Arizona)	7.27	±0.0085	$L2$	2.13

erative partner and suffering as a result.

Despite the difficulty of behavior forecasting, there is no question that learning can play a role, even among higher level strategies. However, that learning needs to take place in the proper space, or else a strategy will not have the capacity to react to basic heuristics. For instance, the top three 2009 players did adapt somewhat in response to their opponents. They did so by recognizing that they were not playing against distributions like those found in single-agent domains, but other players who understood the rules and were prepared to leverage them against slower players. The PI-POMDP framework identifies this reasoning process and is able to suggest a strategy that performs much better than previous agents. The resulting population profile gives insight to predict our opponents and respond preemptively.

In sum, the PI-POMDP analysis achieves good predictions of the strategies' performances. Furthermore, it has revealed characteristic properties of the LG. Future work will aim to show its applicability to further domains and establish the method as a framework to understand similar multiagent games of this kind.

8. ACKNOWLEDGMENTS

The authors would like to thank the National Science Foundation for support on this project via NSF HSD-0624191.

9. REFERENCES

[1] D. Billings. The first international roshambo programming competition. *ICGA Journal*, 23, 2000.

[2] C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.

[3] C. F. Camerer, T.-H. Ho, and J.-K. Chong. A cognitive hierarchy model of games. *Quarterly Journal of Economics*, 119:861–898, 2004.

[4] M. Costa-Gomes, V. Crawford, and B. Broseta. Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235, 2001.

[5] E. M. de Côte, A. Chapman, A. M. Sykulski, and N. R. Jennings. Automated planning in adversarial repeated games. *UAI*, 2010.

[6] D. Egnor. Iocaine powder. *ICGA Journal*, 23, 2000.

[7] N. Friess, J. Aycock, and R. Vogt. Black market botnets. *MIT Spam Conference*, 2008.

[8] Y. Gal. *Reasoning about Rationality and Beliefs*. PhD thesis, Harvard University, June 2006.

[9] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of AI Research (JAIR)*, 24:49–79, 2005.

[10] P. J. Gmytrasiewicz and E. H. Durfee. A rigorous, operational formalization of recursive modeling. *Proceedings of the First International Conference on*

Multi-Agent Systems, pages 125–132, 1995.

[11] P. J. Gmytrasiewicz and E. H. Durfee. Rational communication in multi-agent environments. *Autonomous Agents and Multi-Agent Systems Journal*, 4:233–272, 2001.

[12] P. R. Jordan, M. P. Wellman, and G. Balakrishnan. Strategy and mechanism lessons from the first ad auctions trading agent competition. *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.

[13] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction.* Morgan and Claypool, 2008.

[14] J. Niu, K. Cai, P. McBurney, and S. Parsons. An analysis of entries in the first TAC market design competition. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, December 2008.

[15] D. Ray, B. King-Casas, P. R. Montague, and P. Dayan. Bayesian model of behaviour in economic games. *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[16] J. R. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal form games. *The Twenty-Fourth Conference on Artificial Intelligence (AAAI-10)*, 2010.

[17] M. Zinkevich. The lemonade game competition. http://tech.groups.yahoo.com/group/lemonadegame/, December 2009.

Cognitive Policy Learner: Biasing Winning or Losing Strategies

Dominik Dahlem
SENSEable City Laboratory
Massachusetts Institute of Technology
Cambridge, USA
dahlem@mit.edu

Jim Dowling
Computer Systems Laboratory
Swedish Institute of Computer Science
Stockholm, Sweden
jim.dowling@sics.se

William Harrison
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
bill.harrison@cs.tcd.ie

ABSTRACT

In continuous learning settings stochastic stable policies are often necessary to ensure that agents continuously adapt to dynamic environments. The choice of the decentralised learning system and the employed policy plays an important role in the optimisation task. For example, a policy that exhibits fluctuations may also introduce non-linear effects which other agents in the environment may not be able to cope with and even amplify these effects. In dynamic and unpredictable multiagent environments these oscillations may introduce instabilities. In this paper, we take inspiration from the limbic system to introduce an extension to the weighted policy learner, where agents evaluate rewards as either positive or negative feedback, depending on how they deviate from average expected rewards. Agents have positive and negative biases, where a bias either magnifies or depresses a positive or negative feedback signal. To contain the non-linear effects of biased rewards, we incorporate a decaying memory of past positive and negative feedback signals to provide a smoother gradient update on the probability simplex, spreading out the effect of the feedback signal over time. By splitting the feedback signal, more leverage on the win or learn fast (WoLF) principle is possible. The cognitive policy learner is evaluated using a small queueing network and compared with the fair action and weighted policy learner. Emphasis is placed on analysing the dynamics of the learning algorithms with respect to the stability of the queueing network and the overall queueing performance.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Experimentation

Keywords

Multiagent Reinforcement Learning, Stochastic Policies

1. INTRODUCTION

Multiagent Reinforcement Learning (MARL) techniques have been successfully applied to a number of domains, ranging from general sum games [12, 14] to application areas such as packet routing [18], robot control [15], and resource allocation [20, 21]. A theoretical framework for sequential decision and multiagent problem settings is provided by the formalisms of the decentralised Markov Decision Process (DEC-MDP) [5, 10]. Planning-based solution methods have been devised to solve these models offline. However, their complexity increases dramatically if no reward or transition model is available or the number of agents goes beyond small scenarios. In contrast, online approximate solutions have been shown to be useful in solving DEC-MDP problems [25]. Their central idea is that models of learning and memory are continuously updated and incorporated into a trial-and-error interaction within the agent's local context. Agents learn using only local information, but they should support near optimal global decision making. In unison, all agents contribute to the global goal of optimising some system objective. Simultaneous and independent interactions, however, pose a challenge to multiagent systems, because they are non-deterministic, may have non-linear effects, and may lead to slow convergence characteristics or even diverge. Some research directions tackle these difficulties by modelling the other agents in the environment [9] or by providing a mechanism to communicate feedback of parallel optimisation processes underway in the environment [10, 19].

Additionally, the modelling assumptions of DEC-MDP often need to be extended to capture the application specific constraints. For example, for packet routing or task allocation networks, the service stations or nodes have limited capacity to service requests and limited resources to store waiting tasks. Networked systems exhibit a level of complexity that is very challenging to deal with. In the absence of direct communication links between nodes sharing a common resource, coordination is difficult to achieve to optimally utilise this common resource. For example, consider the queueing network presented in Figure 1 which is used for all evaluation scenarios. Both agents (nodes 6 and 7) share a common resource (node 4) and may observe that the common resource offers enough capacity to service their individual requests. As such, both agents may decide to utilise this resource at the same time causing potential congestion. Under certain conditions, this may lead to fluctuating performance that may cascade through the network. Consequently, autonomous agents need to mitigate the occurrence of cascades (non-linear effects) and adapt quickly to changing conditions.

In this paper, we introduce two new features to the weighted policy learner: an inherent bias that magnifies or depresses rewards depending on how far they diverge from the average expected reward for different actions in that state, and, secondly, a transient memory of recent rewards for actions that smooth out the current

Cite as: Cognitive Policy Learner: Biasing Winning or Losing Strategies, Dominik Dahlem, Jim Dowling, William Harrison, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 601-608.

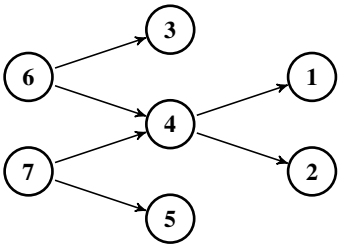


Figure 1: Queueing Network

reward. Both of these features dampen cascading effects of large changes in rewards at key nodes, preventing system hysteresis, but still enabling agents to converge to a stable stochastic policy. Moreover, the cognitive policy learner offers greater control over the extent of the win or learn fast strategy by interpreting the positive and negative feedback signals separately.

We evaluate the cognitive policy learner using a small queueing network given in Figure 1 and compare it with the fair action and weighted policy learner. We investigate whether modulating the strength of feedback signals can have a stabilising impact on the learning system. Emphasis is given to analysing the dynamics of the learning algorithms with respect to the stability of the queueing network and the overall queueing performance. Our results show improved queueing performance compared to the fair action learner, and similar queueing performance to the weighted policy learner. However, our results suggest that our cognitive policy learner yields a more stable multiagent learning system compared to the weighted policy learner, as it has a significantly lower total mean-squared training error for the $SARSA(0)$ steepest-descent gradient update.

2. BACKGROUND

This section provides the background to the collaborative multiagent reinforcement learning environment for queueing networks. We assume that the queueing network is given as a directed acyclic graph, which implies that all interactions between the agents are directed and do not form any cycles. Similar in concept to the collective intelligence framework of Wolpert et al. [23], a subworld, ψ_i, constitutes a number of queueing agents that together complete a task for agent i. Each agent can be viewed as though it is striving to maximise its own reward function with the consequence of improving the performance of the subworld as a whole. The engineering discipline is based on division of labour, where the system is sub-divided into smaller parts. The solution of the decentralised optimisation problem is brought about in a bottom-up fashion. More formally, a subworld can be defined as

DEFINITION 1 (SUBWORLD). *A **subworld**, ψ_i, is a subgraph of the queueing network comprising all agents j reachable from agent i.*

- *The queueing network induces nested subworlds. At the leaf nodes of the queueing network subworlds consist of empty sets.*
- *A **path**, p_i, in subworld ψ_i represents a realisation of a local queueing task assignment to agent i.*
- *$\mathcal{W}_i$ is a set of all possible paths in subworld ψ_i.*

With the help of the subworld definition, the multiagent sequential decision problem can be formalised in a DEC-MDP given in Definition 2.

DEFINITION 2. *An n-agent continuous state **DEC-MDP** of a queueing network is defined by a tuple $\mathcal{M} = \langle \mathcal{DAG}, \mathcal{A}, \mathcal{S}, \mathcal{P}, \mathcal{R} \rangle$, where*

- *$\mathcal{DAG}$ is the directed acyclic graph prescribed by the agent's interactions. Each agent is represented as a vertex on the graph and the arcs between the agents represent available actions to the respective agents.*
- *$\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is the finite set of actions and is given by the possible interactions.*
- *$\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ is finite set of queueing network states, which can be factored into local states $\mathcal{S}_i$ for each agent i, including queueing metrics such as delay, utilisation, or number of events in the queue.*
- *$\mathcal{P}_{w_i} = P\{s_{t+1} = s' \mid s_t = s, \vec{a}_t \exists p_i \in \mathcal{W}_i\}$ is the transition probability of state s' for agent i when the actions $\vec{a}$ comprising path p_i have been taken in state s.*
- *$\mathcal{R}_{w_i} = \mathbb{E}\{r_{t+1} \mid s_t = s, \vec{a}_t \exists p_i \in \mathcal{W}_i, s_{t+1} = s'\}$ is the expected value of the next reward for agent i when actions $\vec{a}$ are taken in state s and transitioning to the next state s'.*

It is important to note that a policy must exist for which the aggregated arrival rates at each node of the queueing network do not yield unstable queues. More specifically, this implies that solving the traffic equations

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 (\mathbf{I} - \mathbf{Q})^{-1}, \tag{1}$$

where $\boldsymbol{\lambda}_0$ is the vector of external Poisson arrival rates for each node in the network and $\mathbf{Q}$ specifies the transition probabilities derived from the policy, requires that the stability criterion, $\frac{\lambda_i}{\mu_i} < 1$, holds for each node. Here, $\boldsymbol{\mu}$ is the vector of exponential service rates, which is considered fixed and represents the nodes' capability to service incoming requests.

Following [25], each agent observes local reward signals, which are given as the negative task processing time. Longer task completion times are less desirable, which makes this reward function a natural choice. This includes all local processing times of the task at each service station (agent) where no communication delay is assumed. Then the value function for a local policy π_i is defined with respect to the average expected reward as:

$$\rho_i(\pi_i) = \lim_{N \to \infty} \frac{1}{N} \mathbb{E}\left[\sum_{t=0}^{N-1} r_i^t \mid \pi_i \right], \tag{2}$$

where r_i^t is the observed reward at time t and it depends on the global states of the queueing network. However, unlike [25], our model cannot be reformulated into an average-reward factored DEC-MDP, because the rewards received by each agent are not independent, that is the global reward is not equal to the sum of the local rewards. Noting that the reward is the response time of the completion of a local task, one can informally see that a reward can only be given when the local task is completed. An external request for a task entering the system at agent i yields an execution path $p_i \in \mathcal{W}_i$ through this agent's subworld ψ_i. Each agent along this path maximises its own reward function forming their respective subworlds. Consequently, since the rewards are computed based on completion times of local tasks a reward relationship of $r_1 < r_2 < \cdots < r_i$ is established, so each r_i is the negative value of the response time of the local task t_i.

This reward structure is said to be global, because it incorporates the rewards of all agents involved in completing a task. More importantly, the reward function has optimal substructure. This means that if we take the negative task completion times as rewards, the credit given to a fulfilled task is apportioned fairly among the agents involved in the completed task. This yields a cooperative multiagent system, as distinct from local reward functions that encourages competitive behaviour among selfish agents. An advantage of this reward structure is that no communication is required to correlate rewards and apportion the reward fairly.

Therefore the agents' reward functions are not mutually independent and offline planning approaches are more difficult to achieve. Moreover, the lack of an explicit reward and transition model increases the complexity of solving such a system and consequently is only feasible for the most simple cases. Online approaches, however, offer a scalable and approximate alternative. Here, we use a standard backpropagation feedforward neural network with one hidden layer on each arc of the task network to estimate the Q-values for each action given a state vector [16]. The temporal difference scheme SARSA(0) can then be expressed as the general gradient-descent update rule for neural network training as

$$\Delta\boldsymbol{\omega}_{t+1} = \alpha[\upsilon_{t+1} - Q(s_t, a_t)]\nabla_{\boldsymbol{\omega}_t} Q(s_t, a_t) + \eta\Delta\boldsymbol{\omega}_t, \quad (3)$$

$$\upsilon_t = r_t + \lambda Q(s_t, a_t), \quad (4)$$

where η is a constant representing the momentum, which determines the effect of past changes to the weight vector, $\boldsymbol{\omega}$, and $\nabla_{\boldsymbol{\omega}_t} Q(s_t, a_t)$ is the vector of partial derivatives of the value function $Q(s_t, a_t)$ with respect to the weight vector $\boldsymbol{\omega}_t$. The action-value estimation is updated every time a task in the queueing network is completed. That means, that all value functions of all arcs in the queueing network that were involved in forwarding a request to the next sub-task will be updated according to $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + \Delta\boldsymbol{\omega}_{t+1}$. The optimal action-value function Q^* is estimated with a parametric function approximator, $Q_{\boldsymbol{\omega}}$, where $\boldsymbol{\omega}$ is the vector of weights as given above. The neural network function approximator is instantiated with one hidden layer and 10 hidden neurons. The agents take only local information into account to train the Q-value function. In all evaluation scenarios of this paper, the delay $\hat{w}_i$ in the local queue forms the input to the neural network. The delay is calculated as the difference between the time of arrival of a task at a node in the queueing network and the time of scheduling the task. The state vector can easily be extended with other queueing metrics, such as current utilisation or the number of task assignments waiting to be scheduled. The queueing discipline is first-in-first-out. This means that as the node is processing a task all tasks arriving at the same time are put into a waiting queue. As the node finishes processing tasks, tasks in the waiting queue are dequeued on a first come, first served basis.

3. RELATED WORK

Multiagent reinforcement learning has seen significant contributions in packet routing [8, 18, 22, 23]. Q-routing was one of the first multiagent approaches applied to routing [8]. Q-routing by itself has its roots in the Bellman-Ford shortest path routing algorithm [4]. The original Q-learning algorithm has routing performance comparable to the Bellman-Ford algorithm under low load. However, since Q-routing uses estimates of the delivery time of a packet, it tends to congest paths if a better performing link has been over-estimated. This problem persists, because Q-routing is a deterministic protocol that always chooses the best performing link to deliver a packet. An attempt to mitigate choosing sub-optimal paths communication between adjacent nodes is established to update estimates before decisions are made. It turns out, however, that high-load scenarios result in fluctuations with routing performance worse off than in the basic Q-learning algorithm. This result can be reconciled with stochastic policies that adjust mixed strategies with continuously adapting agents.

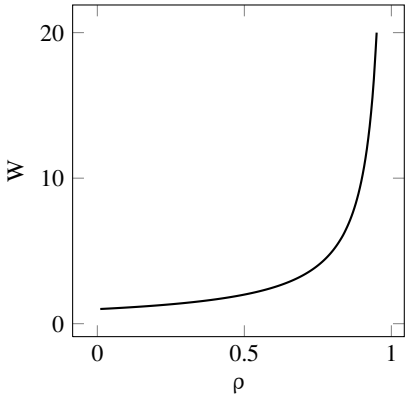


Figure 2: Mean Response Time versus Utilisation

This can be readily understood by studying the sensitivities of queueing metrics with respect to the utilisation of a single queue. Little's theorem states that the long-term average number of events in a stable queue is equal to the long-term average arrival rate multiplied by the long-term average time an event spends in the system [6]. Based on this theorem, the response time versus the utilisation can be examined, $W = \frac{1/\mu}{1-\rho}$, where $\rho = \frac{\mu}{\lambda}$ is the utilisation. Assuming $\mu = 1$ and varying the utilisation rate, the response time has two regimes as shown in Figure 2. For utilisation rates below ~ 70% the response time grows linearly. But for higher utilisation rates the response time has an exponential growth. So, if greedy link selection is employed, sub-optimal decisions have a dramatic impact if the queue is in the sensitive regime. This is why the fluctuations are observed with the extended Q-routing algorithm described above.

Tao et al. introduced a multiagent, partially observable Markov Decision Process for packet routing. Each node in the network is parameterised by a real-valued vector for each destination/outgoing link pair [18]. This vector is adjusted in order to ascend the gradient of the expected long-term average reward for all nodes. The rewards are computed at the destinations of the packets and broadcast into the network. The probability of selecting a link is calculated according to the Gibbs distribution using this vector. This approach is very similar to the application of collective intelligence (COIN) for packet routing [23]. Tumer and Wolpert mathematically formalised collective intelligence solutions and proved that "Tragedy of the Commons" does not exist under certain conditions [19]. These are that the environment can be factored into sub-worlds, which encompass all nodes that share the same destination. The reward is also computed at the destinations and is broadcast in its own sub-world. Each agent maximising its long-term average reward also leads to maximising the global reward. In an extended study, Wolpert and Tumer show that COIN-based models for network routing almost always avoid the Braess' paradox [22]. Braess' paradox states that selfish routing behaviour on a network can result in a lower throughput when additional capacity through a new edge in the network is introduced. In particular, the ideal shortest path algorithm introduced side-effects that lead to the observation of the Braess' paradox. With a COIN-based ap-

proach, Braess' paradox can almost always be avoided while at the same time exhibiting significantly improved global throughput performance.

In highly dynamic and large environments modelling other agents or extra communication effort becomes prohibitively expensive. Direct policy learning algorithms are a promising alternative, because they align with the expected reward for the actions. A direct policy called Fair Action Learning (FAL) is presented in [24]. FAL approximates the policy gradient of each state-action pair using the difference between the expected Q-value for the state and its actual Q-value. As such it learns a stochastic policy that increases the probability of actions receiving a higher reward then the current average. Consequently, FAL will converge to a fair policy reflecting the expected reward for all actions and states. However, if one action is always more favourable than the other ones, FAL will converge to a deterministic policy, which is not always desirable.

The weighted policy learner (WPL) addresses this issue of ensuring that all actions have a minimum probability of being selected [1, 2]. The WPL algorithm was also designed with the need of quickly converging to a stable stochastic policy. This is achieved by performing a gradient ascent towards a stable policy and slowing down learning gradually for as long as the gradient does not change direction and learn fastest when the gradient changes direction. This policy ensures that no action probabilities converge to a deterministic policy using a euclidean projection onto the probabilistic simplex, where each probability is greater than a given value ϵ. Mathematically, this projection is equivalent to solving a constrained optimisation problem for which closed-form and efficient solutions exist whose complexity are linear in time [11]. This is advantageous in settings where agents have a large number of actions. The euclidean projection is given as $\Pi_X(x) = \arg\min_{x' : \text{valid}(x')}(x - x')$, which returns a policy that is closest to x and satisfies the constraints that it sums to 1 and action probabilities are greater than a given parameter ϵ. The weighted policy learning (WPL) algorithm has been applied to distributed task allocation, a similar setting as described in this article, where stochastic stable policies are desirable [2].

Both FAL and WPL use the expected Q-value for the state and its actual Q-value to calculate the gradient. This is in contrast to "Win or Learn Fast" (WoLF) algorithms, such as Generalised Infinitesimal Gradient Ascent WoLF (GIGA-WoLF) [7], which use approximations to determine when an agent is moving towards or away from a Nash Equilibrium.

4. COGNITIVE POLICY LEARNER

This section introduces a policy learning algorithm that is inspired by the limbic system of the brain. There are two basic concepts underlying the cognitive policy learner: firstly, an inherent bias that magnifies or depresses rewards depending on how far they diverge from the average expected reward for different actions in that state, and, secondly, a transient memory of recent rewards for that action that smooth out the current reward. Rewards are categorized as either *positive* or *negative*, depending on whether they are higher or lower than the average expected reward, respectively. Both positive and negative rewards are scaled by the amount they differ from the average expected reward and a fixed bias called the amplitude, $A^{+/-}$. A^+ scales positive rewards, while A^- scales negative rewards. In addition to biases, a transient memory model stores an accrued sum of recent positive rewards, $c^+(a)$, and recent negative rewards, $c^-(a)$. Both $c^{+/-}(a)$ are decayed over time at a configurable rate of decay, $r^{+/-}$. To give an example, this enables us to define a reward model that amplifies positive rewards and spreads out the assignment of the reward over time. So, a positive reward may persist for longer than the current time step. Additionally, the factor for the amplitude can be used to interpret the intensity of the positive or negative feedback signal. For example, by assigning an amplitude twice as high to the positive signal compared to the negative signal, positive signals have a larger impact on the update step on the probabilistic simplex than negative ones. This setting embodies some notion of risk aversion, because punishment does not induce a rapid update of one's strategy to avoid similar negative experiences. While, there might be situations favouring such a setting, it is more intuitive and in fact more natural to have risk-averse agents. Hence the win or learn fast strategies.

CPL is presented in Algorithm 1. The basic principle is similar to the weighted policy learner. Before conducting the update on the policy simplex, the memory for each signal is decayed, and the new signal is multiplied by the amplitude and added to the decayed signal. The memory signals are bounded, i.e., $0 < c^+(a) \leqslant s(a)_{\max}$ and $s(a)_{\min} \leqslant c^+(a) < 0$, where $s(a)_{\min\,/\,\max}$ are the minimum negative and maximum positive observed feedback signal. This means that the accrued feedback signals cannot attain values higher than the single strongest component of the feedback signal. The resultant positive and negative signals are added together, giving $\Delta(a)$, and the vector of all such signals for all actions, Δ, is used to proceed with the policy projection routine $\pi \leftarrow \Pi_X(\pi + \zeta\Delta)$. ζ denotes the update rate also used in FAL and WPL.

Algorithm 1: CPL: Cognitive Policy Learner

Input: $Q(s, a)$, the expected reward for executing action a in state s
Input: $c^+(a)$ & $c^-(a)$, the accrued reward/punishment signal for action a
Input: $A^{+/-}$ & $r^{+/-}$, the amplitude and decay rate for the respective feedback signals

$\bar{Q} = \sum_{a \in A} \pi(a) Q(s, a)$
foreach *action* $a \in A$ **do**
  $c^+(a) \leftarrow c^+(a) * e^{r^+ t}$
  $c^-(a) \leftarrow c^-(a) * e^{r^- t}$
  Decay
  $s(a) \leftarrow Q(s, a) - \bar{Q}$
  if $s(a) > 0$ **then** $c^+(a) \leftarrow c^+(a) + A^+ * s(a)$
  else $c^-(a) \leftarrow c^-(a) + A^- * s(a)$
  $\Delta(a) \leftarrow \max(c^-(a), s(a)_{\min}) + \min(c^+(a), s(a)_{\max})$
end
$\pi \leftarrow \Pi_X(\pi + \zeta\Delta)$
Output: A new policy π
Output: Updated reward/punishment signals $c^+(a)$ & $c^-(a)$

The amplitude parameter can be tuned in four different ways to modulate the effects of the positive and negative feedback signals:

1. $A^+ > A^-, r^+ > r^-$: Positive feedback signals are amplified more than negative ones. Also, accrued positive rewards decay at a slower rate.

2. $A^+ > A^-, r^+ < r^-$: Positive feedback signals are amplified more than negative ones. In contrast to the previous case, accrued negative rewards decay at a slower rate.

3. $A^+ < A^-, r^+ > r^-$: Negative feedback signals are amplified more than positive ones. Also, accrued negative rewards decay at a slower rate.

4. $A^+ < A^-, r^+ < r^-$: Negative feedback signals are amplified more than negative ones. In contrast to the previous case, accrued positive rewards decay at a slower rate.

If both decay rates are set to $-\infty$ and $A^+ = A^-$ then the fair action learner is recovered. If $A^+ < A^-$, then the cognitive policy learner resembles the effects of the weighted policy learner with a win or learn fast strategy without taking the accrued reward/punishment signals into account.

5. EVALUATION

We evaluate the cognitive policy learner using a small queueing network given in Figure 1. The respective external Poisson arrival, $\boldsymbol{\lambda}_0$, and Exponential service rates, $\boldsymbol{\mu}$, are given in Table 1. The network represents three decision makers, nodes 4, 6, and 7. Each node's objective is to optimise the routing decisions of the tasks based on the negative completion times. Node 4 experiences neighbours 1 and 2 with different external arrival and service rates. Node 1 has a lower intrinsic utilisation than node 2 and consequently, node 4 needs to learn a policy that balances this difference such that the reward is maximised, or the task completion times are minimised in the long run. Both nodes 6 and 7 rely on the assignment of tasks given to node 4. Due to the reward functions having optimal substructure, the queueing performances of nodes 6 and 7 improve if node 4 learns an optimal policy. Node 7 has a higher external arrival rate of tasks and therefore receives a higher number of feedback signals from its task assignments than node 6. This implies that node 7 may be faster in recognising deteriorating performances than any of its neighbours. Both nodes share a common resource (node 4) and have each a private resource (node 3 and 5 respectively). Because node 3 has a much lower intrinsic utilisation than node 5, node 6 may be the first to utilise this resource in case node 4 deteriorates.

Table 1: Arrival and Service Rates

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**
λ	0.33	0.51	0.04	0.11	0.21	0.32	0.51
μ	0.68	0.9	0.48	0.76	0.58	0.55	0.55

The initial policy assumes uniformly random probabilities and three policy learning algorithms are evaluated including CPL, the fair action learner (FAL) [24] and the weighted policy learner (WPL) [1]. The underlying euclidean projection is the same for all three algorithms to ensure that action probabilities do not attain values less than a specified parameter $\epsilon = 0.05$. Each algorithm was individually optimised within the range of parameters $\alpha \in [0.0001; 0.1]$, $\lambda \in [0.01; 0.9]$, $\eta \in [0.01; 0.5]$, $\zeta \in [0.1; 0.0001]$ for both FAL and WPL and additionally $A^+ \in [0.01; 2.0]$, $A^- \in [0.01; 2.0]$, $r^+ \in [-20.0; -0.01]$, $r^- \in [-20.0; -0.01]$ for CPL using Gaussian Process Regression [3, 10, 17]. The results of this global optimisation are summarised in the following Table 2.

In all three algorithms the learning rate, discount factor, and the momentum for the $SARSA(0)$ gradient-update descent (Equation 3) are 0.1, 0.9 and 0.5 respectively. The update factor on the policy simplex, ζ, is low for FAL and high for WPL and CPL. The optimal parameters for the cognitive policy learner resemble the win or learn fast strategy, because both the amplitude for the negative signal is higher and the decay rate slower. This means that the memory for negative signals persists for a longer period of time. This result is interesting in that the global simulation optimisation technique found optimal values for CPL that reflect risk-aversion.

Table 2: Optimal Learning Parameters

	FAL	**WPL**	**CPL**
α	0.1	0.1	0.1
λ	0.9	0.9	0.9
η	0.5	0.5	0.5
ζ	0.0001	0.1	0.1
A^+			1.76
A^-			2.0
r^+			-1.7
r^-			-4.55

The analysis of the performance and the dynamics of the different algorithms are based on at least 10 replications of simulation runs using the optimal parameters. These simulation runs are also controlled to be within 90% confidence intervals with a relative error of 10% [13].

Figures 3 and 4 present the queueing results with respect to mean utilisation of the queueing network and total average delay in the queues. The delay measures the time a task waits in the queue until it can be serviced, since each node in the queueing network can only process one task at a time.

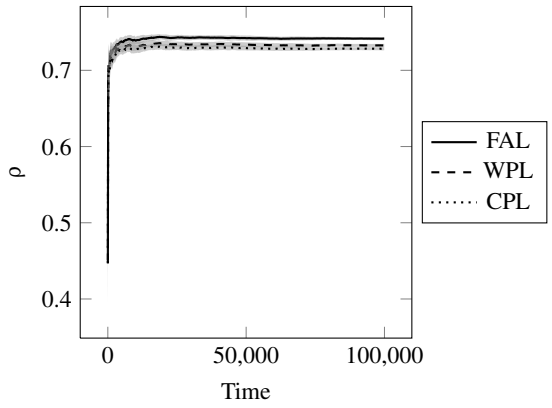


Figure 3: Utilisation

Both utilisation and delay are inferior for the fair action learner, while WPL and CPL show similar queueing performance. The utilisation and the 95% confidence interval half-widths for WPL and CPL respectively are $73.3\%(\pm 0.0009)$ and $72.8\%(\pm 0.0009)$.

Figure 5 shows the percentage of unstable nodes across the replications in order to illustrate the dynamics of WPL and CPL in the steady state using Equation 1. An unstable queue is defined as having a utilisation rate larger than 100% in the steady state. Intuitively, unstable queues show a behaviour of processing the tasks slower than they arrive, which leads to a growing waiting queue.

This calculation is equivalent of assuming the current policy to be fixed. FAL does not yield unstable queues at any given time and is, therefore, not shown in this plot. Nodes 1 and 4 show similar values for the percentage of unstable queues (23% and 8% respectively). But the percentage of unstable queues is increased for CPL for nodes 2 and 5 (2.5 and 9 percentage points higher). This may be explained by the fact that both their respective alternative paths have a lower intrinsic utilisation and since CPL has an accrued memory of the feedback signals, it is slower to adapt to rapidly

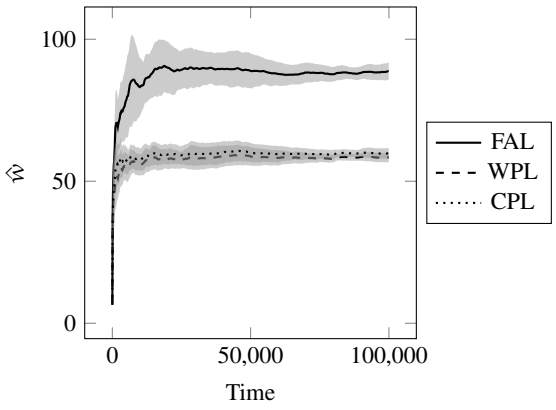


Figure 4: Delay

changing conditions.

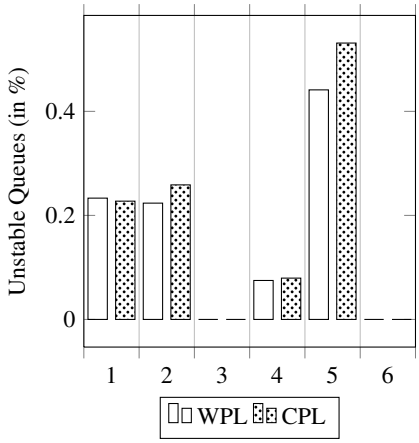


Figure 5: Stability of the Queues

The dynamics of queueing performance measures can be captured in several ways. The first metric we use is based on the matrix of routing probabilities $\mathbf{Q}$ derived from the employed policy. The following quantity captures learning as distinct from random behaviour, where uniformly random decisions manifest themselves in all outgoing links, $\deg^+(\cdot)$, in the network having equal probabilities of being selected. The distance measure from random behaviour is denoted as

$$d_n = \|\mathbf{Q}_{(n)} - \mathbf{Q}^r_{(n)}\|_1, \ \forall\, n \in \mathcal{V} \,\&\, \deg^+(n) > 0, \qquad (5)$$

where $\|\cdot\|_1$ is the ℓ_1-norm of a vector, i.e., $\|\mathbf{a}\|_1 = \sum_{i=1}^{n} |\mathbf{a}_i|$ and $\mathbf{Q}^r_{nj} = \frac{1}{\deg^+(n)}$ is the probability of taking a uniformly random action for all actions j available to n. The probability of taking actions $\mathbf{Q}_{(n)}$ is derived from the employed policy. This measure is bounded by $d_n = 0$, if the action selection probabilities are uniformly random, and $\sup\{d_n\} = 2$ for deterministic action selection as $\deg^+(n) \to \infty$. Also, $d_n = 0,\ n \in \mathcal{V} \,\&\, \deg^+(n) = 1$.

This metric does not make any qualitative statement about learning behaviour, because it cannot be ruled out that uniformly random behaviour is actually the best policy. Instead it gives an indication of how distinctive the learnt policies are.

Figure 6 shows the result for this metric. FAL learns the least distinctive policies in the queueing network, which means that the policy updates are very close to the initial policy configuration of uniformly random decisions. Additionally, the policy gradient update factor, ζ, for FAL is low, suggesting that FAL prefers small incremental changes to the policy. CPL in turn learns the most distinctive policies. These results show that temporarily unstable queues in the steady state lead to a higher throughput in the algorithms considered in this evaluation.

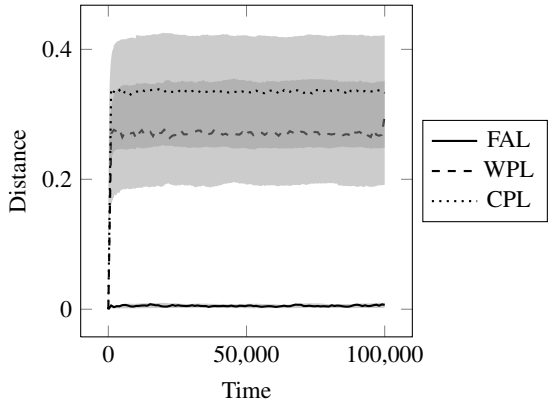


Figure 6: Distance from Uniformly Random Decision Policies

In order to quantify the level of distinctiveness of the policies we evaluate the ownership of node 4. The ownership metric is calculated as the normalised fourth column of the routing matrix $\mathbf{Q}$, where nodes 6 and 7 are the only predecessors. Figure 7 depicts this evolution of ownership. In all cases node 6 directs most of its tasks towards node 3, while node 7 directs most of its tasks towards node 4. Since node 7 also has a higher arrival rate with the same service rate compared to node 6, node 7 dominates node 4. This plot also mirrors our previous result that CPL learns more distinctive policies, i.e., the spread between nodes 6 and 7 is higher for CPL. Importantly, FAL exhibits barely any fluctuations in its learning dynamics. This shows that FAL learns a stochastic stable policy, while WPL and CPL learn stochastic unstable policies. An interesting result, however, is that this instability (which exists only in the steady state) yields a better performing system as a whole.

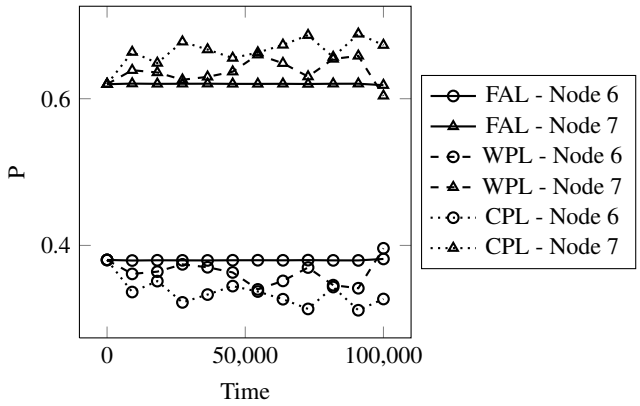


Figure 7: Ownership of Node 4

In order to understand the extent of the fluctuations, the coefficient of variation of the Q-value estimation is calculated based on a buffer of the last 100 Q-value estimations:

$$c_v(Q) = \frac{\sigma_Q}{\mu_Q}. \quad (6)$$

Figure 8 shows the densities of the coefficient of variation for each decision making node individually. Intuitively, one would assume that the coefficient of variation scales with the height of the queueing network, i.e., the least dependent node (here node 4) has lower values than the nodes that depend on it. Interestingly, this is only observed for FAL, which can be interpreted as FAL's learning dynamics results in cascading effects. Because of this behaviour, FAL learns the least distinctive policies and also performs poorly compared to WPL and CPL. Because the values for the coefficient of variation are significantly higher than the ones with WPL and CPL, the densities are left out of Figure 8.

WPL and CPL on the other hand do not exhibit cascading effects. In fact, the fluctuations observed for those algorithms appear to have a stabilising impact on the learning dynamics. For all nodes, the absolute value of the coefficient of variation is slightly higher for CPL compared to WPL.

Figure 9 presents the total mean-squared error of the loss function of the neural network excluding FAL, because it is significantly higher than WPL and CPL again. This plot suggests that the CPL algorithm yields a more stable reinforcement learning system than WPL as its total mean-squared error is significantly lower.

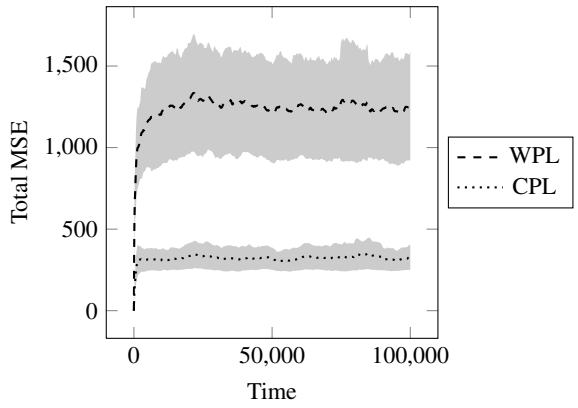


Figure 9: Neural Network Error

To summarise the results, FAL has inferior queueing performance than WPL and CPL. FAL also learns the least distinctive policies, which may be attributed to the large fluctuations in the coefficient of variation of the Q-value estimation. These fluctuations lead to cascading effects in the fair action learner and therefore this policy is not a good candidate for autonomous agents in queueing networks. Instead, both WPL and CPL learn stochastic unstable policies with respect to the instantaneous steady state distribution. However, these instabilities only persist temporarily with agents continuously adapting to the changing conditions in the queueing environment. In fact, these instabilities do not exhibit cascading effects. The coefficients of variation are approximately normally distributed with a slight skewness towards the left. This is in contrast to the distribution for FAL, which has a long tail representing rare events. In multiagent systems behaviours that can be characterised by long-tailed distributions introduce challenges for the other agents to adapt accordingly.

Finally, the mean-squared error of the neural network is the lowest for the CPL algorithm, which implies more stable reinforcement learning updates. However, the structure of the neural network is considered fixed in this paper. Optimising with respect to the neural network structure itself may be one way of reducing the mean-squared error for both FAL and WPL.

6. CONCLUSION AND FUTURE WORK

This paper investigated an extension to the weighted policy learner which modulates the strength of positive and negative feedback. The cognitive policy learner is inspired by the limbic system of the brain. The feedback signal is split into two parts, positive and negative, with respect to the current estimate of the Q-value function. Each signal is given free parameters to model an amplitude and a decay factor. This way the win or learn fast strategy obtains a higher level of control in terms of the updates on the probabilistic simplex. We showed that the cognitive policy learner performs as well as the weighted policy learner.

The empirical investigation of a small queueing network also revealed that the fair action learner exhibits cascading effects in the queueing network. This means that deteriorating performance closer to the leaf nodes of the network has a detrimental impact on the queueing performance of the other nodes dependent on them. This behaviour was not observed with the weighted and cognitive policy learners where the variations of the Q-value estimation is much better behaved. Biasing a losing strategy and maintaining a transient memory of received rewards and punishments results in a more stable multiagent learning system, which was shown to reduce the total mean-squared error of the $\mathrm{SARSA}(0)$ steepest descent gradient update.

Future work will investigate more dynamic and larger queueing settings. Also, the global optimisation of the simulation parameters need to be analysed with respect to how sensitive the optimal values are to small perturbations.

7. ACKNOWLEDGMENTS

The work described in this paper was supported by the Science Foundation Ireland. We thank the anonymous reviewers for their helpful comments.

8. REFERENCES

[1] S. Abdallah and V. Lesser. Learning the task allocation game. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 850–857, New York, NY, USA, 2006. ACM.

[2] S. Abdallah and V. Lesser. Multiagent reinforcement learning and self-organization in a network of agents. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[3] B. Ankenman, B. L. Nelson, and J. Staum. Stochastic kriging for simulation metamodeling. In *2008 Winter Simulation Conference (WSC)*, pages 362–370. IEEE, December 2008.

[4] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.

[5] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[6] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. WileyBlackwell, 2nd edition edition, May 2006.

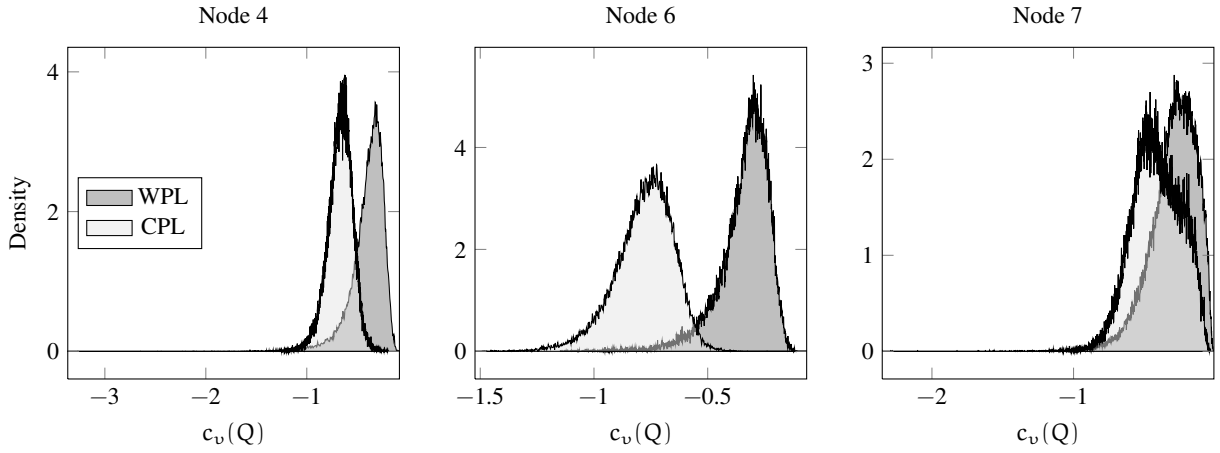


Figure 8: Distribution of CVs

[7] M. Bowling. Convergence and no-regret in multiagent learning. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 209–216. MIT Press, Cambridge, MA, 2005.

[8] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in Neural Information Processing Systems 6*, volume 6, pages 671–678, 1994.

[9] G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1090–1097, Washington, DC, USA, 2004. IEEE Computer Society.

[10] D. Dahlem and W. Harrison. Collaborative function approximation in social multiagent systems. In *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, pages 48–55, Los Alamitos, CA, USA, September 2010. IEEE Computer Society.

[11] J. Duchi, S. S. Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 272–279, New York, NY, USA, 2008. ACM.

[12] J. Hu and M. P. Wellman. Nash q-learning for general-sum stochastic games. *J. Mach. Learn. Res.*, 4:1039–1069, 2003.

[13] A. M. Law and D. W. Kelton. *Simulation Modelling and Analysis*. McGraw-Hill Education - Europe, April 2000.

[14] M. L. Littman. Friend-or-foe q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 322–328. Morgan Kaufmann, 2001.

[15] R. Makar, S. Mahadevan, and M. Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 246–253, New York, NY, USA, 2001. ACM.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, chapter 8, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[17] J. Staum. Better simulation metamodeling: The why, what, and how of stochastic kriging. In *Simulation Conference, 2009. WSC 2009. Winter*, December 2009.

[18] N. Tao, J. Baxter, and L. Weaver. A multi-agent policy-gradient approach to network routing. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 553–560, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[19] K. Tumer and D. Wolpert, editors. *Collectives and the Design of Complex Systems*. Springer, 1 edition, May 2004.

[20] K. Verbeeck, A. Nowé, J. Parent, and K. Tuyls. Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Autonomous Agents and Multi-Agent Systems*, 14(3):239–269, November 2006.

[21] K. Verbeeck, J. Parent, and A. Nowé. Homo egualis reinforcement learning agents for load balancing. In *Innovative Concepts for Agent-Based Systems*, pages 81–91. Springer Berlin / Heidelberg, 2003.

[22] D. H. Wolpert and K. Tumer. Collective intelligence, data routing and braess' paradox. *J. Artif. Int. Res.*, 16(1):359–387, 2002.

[23] D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 952–958, Cambridge, MA, USA, 1999. MIT Press.

[24] C. Zhang, V. Lesser, and P. Shenoy. A multi-agent learning approach to online distributed resource allocation. In *IJCAI 2009, Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, pages 361–366, July 2009.

[25] C. Zhang, V. R. Lesser, and S. Abdallah. Self-organization for coordinating decentralized reinforcement learning. In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen, editors, *AAMAS*, pages 739–746. IFAAMAS, 2010.

Agent-mediated Multi-step Optimization for Resource Allocation in Distributed Sensor Networks

Bo An, Victor Lesser, David Westbrook
Department of Computer Science
University of Massachusetts, Amherst, USA
{ban,lesser,westy}@cs.umass.edu

Michael Zink
Dept. of Electrical and Computer Engineering
University of Massachusetts, Amherst, USA
zink@ecs.umass.edu

ABSTRACT

Distributed collaborative adaptive sensing (DCAS) of the atmosphere is a new paradigm for detecting and predicting hazardous weather using a large dense network of short-range, low-powered radars to sense the lowest few kilometers of the earths atmosphere. In DCAS, radars are controlled by a collection of Meteorological Command and Control (MC&C) agents that instruct where to scan based on emerging weather conditions. Within this context, this work concentrates on designing efficient approaches for allocating sensing resources to cope with restricted real-time requirements and limited computational resources. We have developed a new approach based on explicit goals that can span multiple system heartbeats. This allows us to reason ahead about sensor allocations based on expected requirements of goals as they project forward in time. Each goal explicitly specifies end-users' preferences as well as a prediction of how a phenomena will move. We use a genetic algorithm to generate scanning strategies of each single MC&C and a distributed negotiation model to coordinate multiple MC&Cs' scanning strategies over multiple heartbeats. Simulation results show that as compared to simpler variants of our approach, the proposed distributed model achieved the highest social welfare. Our approach also has exhibited similarly very good performance in an operational radar testbed that is deployed in Oklahoma to observe severe weather events.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Design, Experimentation, Performance

Keywords

Multi-agent systems, sensor networks, coordination, negotiation

1. INTRODUCTION

Over the last 6 years we have been developing and deploying a new paradigm called collaborative adaptive sensing of the atmosphere (CASA) for detecting and predicting hazardous weather [5, 15]. This new paradigm is achieved through a *distributed*, *collaborative*, *adaptive sensing* (DCAS) architecture. Distributed refers to the use of large numbers of small radars, whose range is short enough to see close to the ground in spite of the Earth's curvature and to avoid resolution degradation caused by radar beam spreading. Collaborative operation refers to the coordination of the beams from multiple radars to cover the blind, attenuated or cluttered regions of their neighbors and to simultaneously view the same region in space (when advantageous), thus achieving greater sensitivity, precision, and resolution than possible with a single radar. Adaptive refers to the ability of these radars and their associated computing and communications infrastructure to dynamically reconfigure in response to changing weather conditions and end-user needs. The principal components of a CASA DCAS system includes the sensors (radars); algorithms that detect, track, and predict meteorological hazards; interfaces that enable end-users to access and interact with the system; storage; and an underlying substrate of distributed computation that dynamically processes sensed data and manages system resources. At the heart of a DCAS system is its Meteorological Command and Control (MC&C) that performs the system's main control loop - ingesting data from the remote radars, identifying meteorological features in this data, reporting features to end-users, and determining each radar's future scan strategy based on detected features and end-user requirements.

MC&Cs' resource allocation problem of deciding radars' scan strategies is challenging due to a number of reasons. First, DCAS is an end-user driven approach and different users (e.g., National Weather Service (NWS) whose role is to issue severe weather watches and warnings, regional Emergency Managers (EMs) whose role is to alert the public about weather hazards and to coordinate first responders) have different data collection preferences and needs. Second, in DCAS, adaptive radars are controlled by a collection of MC&Cs and each MC&C tries to find the best scan strategy for the set of radars it controls. While such a distributed model brings some good properties (e.g., robustness, scalability), the problem of coordinating scan strategies of multiple MC&Cs arises as radars belonging to different MC&Cs may have an overlapping region. In certain situations, it is advantageous to have two or more radars focus their scans on overlapping regions in the atmosphere to provide accurate estimation of wind velocity vectors. In some other situations, a single radar's scanning can provide very high quality data and coordination can allow other radars to scan other meteorological features. Third, DCAS is a real-time system and radars must be re-tasked by the MC&Cs every 60 seconds, which defines the system heartbeat interval [5, 15]. Therefore, the optimization for allocating radar resources should be completed in less than 60 seconds. Furthermore, the strategy space of each radar is infinite since each scan action can be represented by a region in the atmosphere.

In the previous resource allocation model [3], all the MC&Cs are myopically optimizing every "single" heartbeat's utilities without explicitly taking into account end-users' various needs over

Cite as: Agent-mediated Multi-step Optimization for Resource Allocation in Distributed Sensor Networks, Bo An, Victor Lesser, David Westbrook and Michael Zink, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 609-616.

multiple heartbeats. It turns out this model leads to poor performance since subsequent changes in the environment may make these myopic decisions not appropriate anymore. For instance, an MC&C may repeatedly scan a high utility phenomenon (no matter how many times it has been scanned before) and thus miss some less important phenomena. Furthermore, a user's data requirement can be satisfied in different ways over multiple heartbeat intervals, which implies that it needs to search over multiple system heartbeats to find radars' optimal scanning strategies and address potential conflicts of available resources (sensing, computation, and bandwidth) over multiple heartbeats. In addition, predictions about future events [9] are useless in this optimization framework.

The focus of this paper is investigating the practicality of applying real-time distributed multi-step optimization approaches in a real application involving complex resource allocation. We found that a real-time distributed multi-step optimization approach is feasible and it contributes to better performance. This paper proposes a novel distributed resource allocation approach to address diverse user preferences over multiple heartbeats. We introduce the concept "goal" (or constraint) to represent end-users' preferences on radars' scan strategies. Each goal specifies the region of a phenomenon over multiple heartbeats and how well a user's preference is satisfied given radars' scan strategies over multiple heartbeats. Then the resource allocation problem can be formulated as a continuous time constraint optimization problem. The goal based formulation allows us to reason ahead about allocations based on expected requirements of goals over multiple heartbeats and prediction about future weather phenomena. Given that the strategy space of each radar is continuous and the real-time requirement, it is impractical to exhaustively search all the possible strategies. Alternatively, each MC&C finds approximate local optimal solutions employing a genetic algorithm. Different strategies are mapped into chromosomes and genetic operators like mutation, selection, and crossover are employed. A distributed asynchronous negotiation model is used to coordinate the scan strategies of multiple MC&Cs. Each MC&C always notifies its current multi-heartbeat strategy to its neighbor MC&Cs. Based on the current strategies of its neighbor MC&Cs, an MC&C proposes to change its strategy and decides whether to make the change based on the marginal utilities of its neighbors' strategy changes. This asynchronous negotiation continues until the heartbeat deadline approaches. Simulation results show that as compared to other mechanisms, the proposed distributed model achieved the highest social welfare. We have also verified the performance of our approach in the operational radar testbed deployed in Oklahoma while it was responding to actual severe weather events. These empirical results mirror the positive results achieved in our simulation studies.

The remainder of this paper is organized as follows. Section 2 discusses related work. We next formalize the resource allocation problem in Section 3. We then discuss the genetic algorithm for finding each MC&C's local optimal strategies. The distributed negotiation model is presented in Section 5. Section 6 reports simulation results. Section 7 discusses the performance of our approach in the real sensor system and Section 8 concludes this paper.

2. RELATED WORK

The development of decentralized optimization and coordination techniques to achieve good system-wide performance is a fundamental challenge for practical distributed sensor networks, which mainly comes from various constraints, e.g., realtime response, limited communication and computational resources. While multi-agent systems community has developed a variety of techniques for distributed resource allocation in sensor networks [4, 8], these approaches cannot be directly applied to our special domain with complex user preferences.

The problem of decentralized coordination can be formulated as a distributed constraint optimization problem (DCOP), which makes it possible for us to use a wide range of existing algorithms for DCOP, e.g., ADOPT [6]. However, these complete algorithms cannot be directly applied to problem due to their limitations such as high computational complexity and large size of exchanged messages. Furthermore, these algorithms are for one-step optimization but our problem is a continuous optimization problem. While there have been numerous approximate stochastic algorithms based on entirely local computation for solving DCOPs [14], these algorithms often converge to poor-quality solutions because agents typically communicate only their preferred state, failing to explicitly communicate utility information [8]. Max-sum algorithm has recently been applied to the sensor network domain (e.g., [12]), our recent study [2] showed that the max-sum algorithm did not outperform the approach in [3] and had a much worse performance when there were more overlapping radars.

Negotiation has been used in distributed sensor networks in the past; however, previous techniques are not entirely appropriate for our setting. In the argumentation-based approach [11], an initiator attempts to recruit other sensors to scan a specific task. In our domain, a per goal negotiation would not be feasible based on time limitations. Contract-net based negotiation schemes [10] in which agents make bids based on utility calculations face similar limitations. If the contract net protocol is adopted, every time an MC&C's neighbor changes its scan strategy, that MC&C must perform potentially as many optimizations for marginal utility calculations as the size of the powerset of the boundary goals belonging to it. The similar problem exists while adopting combinatorial auctions [1]. The negotiation model for single step optimizations for the DCAS system [3] fails to capture users' preferences over multiple heartbeats and accordingly, may result in low social welfare due to lack of reasoning about future actions. In addition, the synchronous negotiation protocol in [3] may have bad performance due to its lack of concurrency in real time optimization.

3. PROBLEM FORMULATION

This section formalizes the problem of optimizing resource allocation which has *observed phenomena* as its input and *scan commands* as its output. The following components are involved in solving the meteorological control problem: goal generation, local optimization that generates scan commands for each MC&C's radars, and negotiation which coordinates MC&Cs' scan actions.

3.1 Goal Generation

In the current design, the DCAS system dynamically adapts radar scans at 60 second intervals to sense the evolving weather and disseminates information to users based on their changing and diverse preferences for data. An NWS forecaster may analyze the vertical structure of a storm to determine whether to issue a warning by viewing a sector scan at multiple elevations, while an emergency manager may require two radars to collaborate in order to pinpoint the location of the most intense part of a storm for spotter deployment, and a researcher may require 360 degree scans at all elevations to initialize a numerical weather prediction model. These diverse information preferences require different radar scan strategies. We use *scan goals* to formulate diverse user preferences and phenomena regions over multiple heartbeats. A goal g specifies:

- Generation time $T_s(g)$.
- Deadline $T_e(g)$. There could be a goal existing for only one heartbeat, i.e., $T_e(g) = T_s(g)$. It is also possible that $T_e(g) -$

$T_s(g) > 0$, i.e., the satisfaction of the goal may involve scan actions over multiple heartbeats.

- Scan area(s). A goal g is either to find new phenomena or to find more details of an existing known phenomenon. For the former case (e.g., 360° scan), the scan area will not change over time. For the latter case, as a phenomenon moves over time, the scan areas at different heartbeats may be different which depends on the moving speed of a phenomenon and how its shape is expected to change over time. Let $A(g,t)$ denote the scan area of goal t at heartbeat $t \in [T_s(g), T_e(g)]$ and such information can be gained by prediction [9]. A goal may be updated later due to imprecise prediction. An area A (or part of it) may fall within the coverage of a radar r, i.e., $\Psi(A,r) = \text{true}$.
- Utility calculation function $U_g(s_{\mathcal{R}}^{T_s(g)\rightarrow T_e(g)})$ which defines how well the goal is satisfied based on radars $\mathcal{R}$'s scan actions $s_{\mathcal{R}}^{T_s(g)\rightarrow T_e(g)}$ from heartbeat $T_s(g)$ to $T_e(g)$.

In summary, goals specify 1) in what manner different kinds of weather phenomena should be scanned by radars and 2) how well different user groups are satisfied given radars' scan strategies. A goal generation rule specifies when the rule is triggered to generate a new goal and how to set the properties of the new goal. A simple example of goal generation rules is that each radar needs to do a 360° scan every 5 minutes (heartbeats).

Note that each MC&C generates goals individually. It's possible that two MC&Cs generate goals for the same phenomena which is located on the overlapping area of the two MC&Cs. In such cases, coordination mechanisms (Section 5) are used to resolve such conflicts. When a goal is generated to find the details of a known phenomenon, the goal will also specify its "regions" in the future based on the prediction about the phenomenon's moving speed and change of its shape. Therefore, an MC&C may also update the properties of an existing goal based on its new observations. This update is important as prediction made at goal generation time may not be accurate enough.

3.2 Goal Satisfaction

Let the set of MC&Cs be $\mathcal{M} = \{M_1, \ldots, M_{|\mathcal{M}|}\}$ and the set of radars be $\{\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{M}|}\}$, where $\mathcal{R}_i$ is the set of radars controlled by MC&C M_i and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$. Each radar has its coverage area and the coverage area of an MC&C includes the coverage areas of all its radars. MC&C M_i is a neighbor of M_j if their coverage areas overlap. Let $\mathcal{NM}_i$ denote the set of neighbor MC&Cs of M_i. Let $\mathcal{G}_i^t$ be the set of goals generated for M_i at the beginning of heartbeat t, i.e., for any $g \in \mathcal{G}_i^t$, $T_s(g) = t$. Accordingly, goals generated by all MC&Cs at heartbeat t is $\mathcal{G}^t = \cup_{M_i \in \mathcal{M}} \mathcal{G}_i^t$. Let $\mathcal{G}_i^{t\rightarrow t'}$ be the set of goals generated for MC&C M_i from heartbeat t to heartbeat t', i.e., $\mathcal{G}_i^{t\rightarrow t'} = \cup_{t \le t'' \le t'} \mathcal{G}_i^{t''}$. A goal g is *active* at time t if $t \in [T_s(g), T_e(g)]$. Let $\mathcal{AG}_i^t$ be the set of *active* goals of MC&C M_i at heartbeat t, i.e., $\mathcal{AG}_i^t = \{g | g \in \mathcal{G}_i^{0\rightarrow t}, T_e(g) \le t\}$. Accordingly, active goals for all MC&Cs at heartbeat t is $\mathcal{AG}^t = \cup_{M_i \in \mathcal{M}} \mathcal{AG}_i^t$. Out of the set of goals in $\mathcal{AG}_i^t$, some are *boundary* goals $\mathcal{BG}_i^t$. A goal $g \in \mathcal{AG}_i^t$ is a boundary goal if there exits a radar r' belonging to another MC&C and one of goal g's scan area from time t could be partially covered by r', i.e., $\Psi(A(g,t'), r') = \text{true}$ for some $r' \in \mathcal{R}_j$ and $t \le t' \le T_e(g)$.

We assume that the set of end-users are $\mathcal{K}$. Let $w_k(g)$ be the weight associated with user $k \in \mathcal{K}$ for goal g. The user weight $w_k(g)$ reflects 1) the relative priority of user k with respect to other users and 2) the importance of goal g from user k's perspective. The values of $w_k(g)$ are set by high-level system user policies. A radar's scan action (strategy) can be defined to be the start and end angles of the sector to be scanned by an individual radar for a fixed interval of time (a heartbeat). Utility evaluation of a goal depends on both scan quality and weight. Quality measures how well an area is scanned, with quality depending on the amount of time a radar spends sampling a voxel in space, the degree to which an area is scanned in its (spatial) entirety, and the number of radars observing an area.

Quality function: The quality $Q(A, s_r)$ of scanning an area A using scan action s_r by a single radar r can be defined as

$$Q(A, s_r) = F_c(c(A, s_r)) \times \left[\beta F_d(d(r, A)) + (1-\beta) F_w(\frac{wd(s_r)}{360})\right]$$

where $wd(s_r)$ is the size of sector s_r, $a(r, A)$ is the minimal angle that would allow r to cover A, $c(A, s_r) = \frac{wd(s_r)}{a(r,A)}$ is the coverage of A by s_r, $h(r, A)$ is the distance from r to geometric center of A, $h_{max}(r)$ is the range of radar r, $d(r, A) = \frac{h(r,A)}{h_{max}(r)}$ is the normalized distance from r to A, and β is a tunable parameter. F_c captures the effect on quality due to the percentage of the area covered. F_w captures the effect of radar rotation speed on quality. F_d captures the effects of the distance from the radar to the geometrical center of the phenomenon area.

A scan area may be scanned by more than one radar in the same heartbeat. $Q(A, s_{\mathcal{R}}^t)$ is the maximum quality obtained for scan area A over a set of radars $\mathcal{R}$ and their scan actions $s_{\mathcal{R}}^t$ at time t. If the phenomenon corresponding to the scan area A is a *pinpointing* phenomenon, $Q(A, s_{\mathcal{R}_i}^t)$ is defined as $Q(A, s_{\mathcal{R}}^t) = \sum_{r \in \mathcal{R}} Q(A, s_r^t)$ where s_r^t is the scan action for radar r at time t. Otherwise, $Q(A, s_{\mathcal{R}}^t) = \max_{r \in \mathcal{R}} Q(A, s_r^t)$.

We can get user k's utility $U_g(k, s_{\mathcal{R}}^t)$ of satisfying the goal g given the scan actions $s_{\mathcal{R}}^t$ by combining the weight component and the quality component. Formally

$$U_g(k, s_{\mathcal{R}}^t) = \begin{cases} \delta^{(t - T_s(g))} w_k(g) Q(A(g,t), s_{\mathcal{R}}^t) & \text{if } T_s(g) \le t \le T_e(g) \\ 0 & \text{otherwise} \end{cases}$$

where $\delta \in (0.1]$ is a discount factor reflecting a user's eagerness of scanning a phenomenon earlier.

Let $U_g(k, s_{\mathcal{R}}^{t\rightarrow t'})$ be user k's utility of satisfying goal g based on a series of scan actions $s_{\mathcal{R}}^{t\rightarrow t'} = \{s_{\mathcal{R}}^t, \ldots, s_{\mathcal{R}}^{t'}\}$ from t to t'. There are multiple ways of defining $U_g(k, s_{\mathcal{R}}^{t\rightarrow t'})$, e.g., $\max_{t \le q \le t'} U_g(k, s_{\mathcal{R}}^q)$, $\max_{t \le q < t'} (U_g(k, s_{\mathcal{R}}^q) + U_g(k, s_{\mathcal{R}}^{q+1}))$, $\max_{t \le p < q \le t'} (U_g(k, s_{\mathcal{R}}^p) + U_g(k, s_{\mathcal{R}}^{q+1}))$, or $\sum_{t \le q \le t'} U_g(k, s_{\mathcal{R}}^q)$. Given actions $s_{\mathcal{R}}^{t\rightarrow t'}$, the aggregate utility $U_g(s_{\mathcal{R}}^{t\rightarrow t'})$ for satisfying a goal g is the sum $\sum_{k \in \mathcal{K}} U_g(k, s_{\mathcal{R}}^{t\rightarrow t'})$ of utilities of all users.

3.3 Formulation of the Optimization Problem

The objective of the optimization is to satisfy the set of goals $\mathcal{G}^0, \mathcal{G}^1, \ldots, \mathcal{G}^\infty$. At heartbeat t, MC&Cs need to determine optimal radar scanning actions at t and later heartbeats for active goals $\mathcal{AG}^t$. However, limited computational resources preclude that we could compute the optimal actions from now to the infinite future. Instead, we adopt the *receding horizon control* principle by focusing on the optimal actions $s_{\mathcal{R}}^{t\rightarrow t+l-1}$ in heartbeats of length l:

$$\arg \max_{s_{\mathcal{R}}^{t\rightarrow t+l-1}} \sum_{g \in \mathcal{AG}^t} U_g(s_{\mathcal{R}}^{0\rightarrow t-1} \cup s_{\mathcal{R}}^{t\rightarrow t+l-1})$$

This formulation is in some sense "myopic" as, in fact, MC&Cs need to consider what's going to happen in the future while deciding "optimal" actions at heartbeat t. As it is not possible to obtain perfect information about future states, a guaranteed optimal solution is not possible to obtain (even neglecting the computational intractability nature of the problem at hand). Although the optimization process at heartbeat t will output a schedule over multiple

heartbeats, only the scan strategies at time t will be executed by radars. At time $t+1$, each MC&C updates its goal sets, possibly generates new goals, and runs the optimization algorithm again.

4. LOCAL OPTIMIZATION OF EACH MC&C

This section discusses how an MC&C M_i searches scanning actions for its radars $\mathcal{R}_i$ given the set of active goals $\mathcal{AG}_i^t$ at heartbeat t. The optimization problem of MC&C M_i at time t is to find the best scan strategy $s_{\mathcal{R}_i}^{t\rightarrow t+l-1}$ for its radars. Formally,

$$\arg \max_{s_{\mathcal{R}_i}^{t\rightarrow t+l-1}} \sum_{g\in\mathcal{AG}_i^t} U_g(s_{\mathcal{R}_i}^{t\rightarrow t+l-1} \cup s_{\mathcal{R}_i}^{0\rightarrow t-1} \cup s_{\mathcal{R}_{-i}}^{0\rightarrow t+l-1})$$

where $s_{\mathcal{R}_{-i}}^{0\rightarrow t+l-1}$ are the strategies of M_i's neighbor MC&Cs, which can be known to M_i through information exchange - negotiation.

The search depth l should be no larger than $\max_{g\in\mathcal{AG}_i^t} T_e(g)$ and setting search depth l involves a number of tradeoffs. With a larger search depth l, M_i has a larger space to coordinate the scan strategies of its radars. However, as only the scan strategy at time t will be executed, the optimal strategy $s_{\mathcal{R}_i}^{t\rightarrow t+l-1}$ found at time t may be not optimal in practice. Furthermore, the computational complexity of searching for optimal strategies increases with the search depth l. In addition, the prediction of the movement of observed phenomena could be inaccurate. When search depth l is large, the propagation of such inaccuracy could lead to poor performance of the scan strategies.

Since the strategy space of each radar is continuous, we first discretize the radar's strategy space such that the start and end angles of each strategy can only be in $\{0, 5, 10, \ldots, 360\}$.[1] Then for each goal $g \in \mathcal{AG}_i^t$ and each radar $r \in \mathcal{R}_i$ such that $\Psi(A(g,t'), r) = \text{true}$ at $t \le t' \le T_e(g)$, generate the minimum sector that can cover the region $A(g,t')$ and add the sector to the candidate strategy set $\mathcal{S}_r^{t'}$ of radar r. If $\mathcal{S}_r^{t'}$ contains more than λ strategies, combine two randomly selected strategies into one strategy and this process continues until $|\mathcal{S}_r^{t'}| = \lambda$. The maximum size of M_i's strategy space is $|\mathcal{R}_i|^{\lambda^l}$. For ease of analysis, we assume that each strategy in $\mathcal{S}_r^{t'}$ has an ID ranging from 0 to $|\mathcal{S}_r^{t'}|-1$. Similarly, we give each radar $r \in \mathcal{R}_i$ an ID ranging from 0 to $|\mathcal{R}_i| - 1$.

The complexity of the optimization problem precludes an MC&C from using an exhaustive search to find its optimal solution. Alternatively, we use a genetic algorithm (GA) to search the (nearly) best solution. The GA generates a sequence of populations as the outcome of a search method. The individuals of the population are scan strategies over multiple heartbeats. Each strategy combination can be represented as a matrix of size $|\mathcal{R}_i| \times l$ in which column j represents radars' scanning strategies at heartbeat $t+j$ and row i represents radar i's scanning strategies from heartbeat t to heartbeat $t+l-1$. Let the matrix for a strategy combination be X. Then $x_{i,j}$ represents radar i's scanning strategies from heartbeat $t+j$ and it follows that $x_{i,j} \in [0, |\mathcal{S}_i^{t+j}| - 1]$.

An individual's fitness value is determined by the utility of all the goals $\mathcal{AG}_i^t$ while all radars take strategies of the individual. The evolution starts from a population of randomly generated individuals. In each generation, operators selection, crossover (recombining existing genetic materials in new ways) and mutation (introducing new genetic materials by random modifications) are used to form a new population. The new population is then used in the next iteration of the algorithm. The algorithm terminates when the local optimization deadline τ (e.g., 5 seconds) has been reached, or the population is stable (e.g., 95% of the individuals have the same highest fitness value). When the genetic algorithm terminates, the chromosome that has the highest fitness is extracted and the decoded strategies are the best strategies for the MC&C.

[1]This does not have a substantial impact on the system since we always scan a little wider than the edges of a phenomena anyway.

Algorithm 1: The Negotiation Algorithm for MC&C M_i

Let $\Theta \in \{\text{wstrategy}, \text{wproposal}\}$ represent the status of MC&C M_i.
Let function `GetTime()` return the current time.
Let Ω_{str}/Ω_{move} be the queue to store other MC&Cs' strategy update/proposals.
Initialization:
  a). Send goal set $\mathcal{AG}_i^t$ to its neighbor MC&Cs ($\Omega_{str} = \mathcal{NM}_i$);
  b). Run the genetic algorithm and get optimal scanning strategies $s_{\mathcal{R}_i}$;
  c). Send $s_{\mathcal{R}_i}$ to all neighbor MC&Cs;
  d). Set $\Theta =$ wstrategy and $nowt =$ `GetTime()`;
while *optimization deadline has not expired* **do**
  if Θ=*wstrategy* **and** $\Omega_{str} \neq \emptyset$ **then**
    if $(\text{GetTime}() - nowt) > \xi$ **or** $\Omega_{str} = \mathcal{NM}_i$ **then**
      Run the genetic algorithm and get new optimal strategies $s'_{\mathcal{R}_i}$;
      if *MC&C M_i can gain positive marginal utility by using $s'_{\mathcal{R}_i}$* **then**
        Send $s'_{\mathcal{R}_i}$ with its marginal utility to all neighbor MC&Cs;
        Set $\Theta =$ wproposal, $\Omega_{move} = \emptyset$,
        $nowt =$ `GetTime()`;
  else if Θ=*wproposal* **and** $(\text{GetTime}() - nowt) > \xi$ **then**
    if *The marginal utility of MC&C M_i by using $s'_{\mathcal{R}_i}$ is higher than its neighbor MC&Cs' marginal utility* **then**
      Set $s_{\mathcal{R}_i} = s'_{\mathcal{R}_i}$, $\Omega_{str} = \emptyset$;
      Send $s'_{\mathcal{R}_i}$ to all neighbor MC&Cs $\mathcal{NM}_i$;
    Set $\Theta =$ wstrategy, $nowt =$ `GetTime()`;

When each MC&C runs the local optimization algorithm separately resulting in a scan strategy based on its local (partial) view of the physical space, efficiency loss may occur. One such source of quality degradation is the loss of the ability to cooperatively scan pinpointing phenomena on boundaries, which can be solved by coordinating scans between MC&Cs and sharing resulting raw data. Another source of lessened quality are redundant scans which can be alleviated by allowing MC&Cs to share abstract level information regarding goals located in boundaries. The limitations of the fully distributed optimization lead us to study the coordination problem of distributed MC&Cs.

5. NEGOTIATION BASED COORDINATION

This section extends the negotiation model in [3] to accommodate 1) user's complex preferences over multiple heartbeats and 2) the need of concurrency during negotiation. In [3], all the MC&Cs ignore end users' preferences over multiple heartbeats and are only maximizing the social welfare of a "single" heartbeat. Accordingly, the model in [3] may lead to poor performance due to its lack of reasoning ahead. In [3], MC&Cs conduct synchronous negotiation. That is, after an MC&C makes proposals to its neighbors, it will respond to its neighbors only after it has received all responses from its neighbors. While the synchronous protocol can guarantee that the social welfare will improve after each round of negotiation, it may be unreasonable for an MC&C to wait for responses from its neighbors given the real time constraints and bounded computational resources.

Algorithm. 1 shows how the distributed negotiation is conducted between MC&Cs at heartbeat t. For a boundary phenomenon, it is possible that one MC&C observes it while another MC&C fails to discover it. Before MC&Cs begin the main stages of negotiation, each MC&C communicates with its neighbors MC&Cs to make sure its boundary goals are also in the goal sets of other MC&Cs.

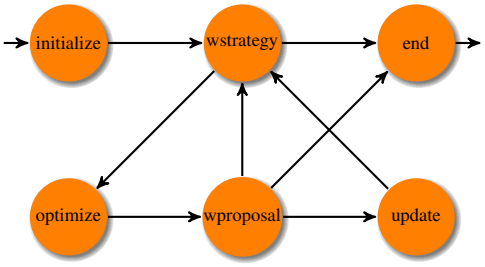


Figure 1: Finite state machine for distributed negotiation

Then each MC&C runs its local optimization algorithm (Section 4) to generate its initial strategy over multiple heartbeats. Next each MC&C shares its initial configuration with its neighbor MC&Cs. Then the main stages of negotiation starts. The negotiation stops when the heartbeat deadline (60 seconds) approaches.

Since an MC&C's optimal strategy depends on its neighbor MC&Cs' scan strategies. After one neighbor MC&C changes its strategy, M_i's optimal strategy $s_{\mathcal{R}_i}$ may be not optimal anymore. Thus, M_i can run its local optimization algorithm again to find its new optimal scan strategy $s'_{\mathcal{R}_i}$. M_i's marginal utility $UM_i(s'_{\mathcal{R}_i}, s_{\mathcal{R}_i})$ when switching to strategy $s'_{\mathcal{R}_i}$ is $\sum_{g\in\mathcal{AG}_i^t} U_g(s'_{\mathcal{R}_i} \cup s_{\mathcal{R}_i}^{0\to t-1} \cup s_{\mathcal{R}_{-i}}^{0\to t+l-1}) - \sum_{g\in\mathcal{AG}_i^t} U_g(s_{\mathcal{R}_i} \cup s_{\mathcal{R}_i}^{0\to t-1} \cup s_{\mathcal{R}_{-i}}^{0\to t+l-1})$.

If $UM_i(s'_{\mathcal{R}_i}, s_{\mathcal{R}_i}) > 0$, M_i may choose to use strategy $s'_{\mathcal{R}_i}$. Since it is possible that other MC&Cs change their strategies simultaneously, M_i's new strategy $s'_{\mathcal{R}_i}$ may be not optimal any more since the optimality of $s'_{\mathcal{R}_i}$ is based on the assumption that M_i's neighbor MC&Cs don't change their strategies. To overcome the efficiency loss due to concurrency, a synchronization mechanism is used: An MC&C first proposes a strategy move by reporting its new strategy as well as its marginal utility to its neighbors, and then it changes its strategy if and only if its marginal utility is higher than the marginal utilities of its neighbor MC&Cs whose proposed *moves* are in *conflict* with the MC&C's proposed move. Since MC&Cs operate in real-time, it is possible that M_i fails to receive the proposal from one of its neighbor MC&Cs or it has to wait for a long time before receiving all proposals. To improve concurrency, we introduce a waiting deadline $\xi > 0$. MC&C M_i will decide whether to make a move after the waiting deadline expires.

Assume that M_i received a message from M_j indicating that M_j will change its strategy from $s_{\mathcal{R}_j}$ to $s'_{\mathcal{R}_j}$. M_j's move $s'_{\mathcal{R}_j}$ is in conflict with M_i's move $s'_{\mathcal{R}_i}$ if both moves will change the utility of some active goals $\mathcal{G} \subseteq \mathcal{AG}_i^t$. Let $\mathcal{NM}_i(s'_{\mathcal{R}_i})$ be the set of neighbor MC&Cs whose proposed moves are in *conflict* with the M_i's proposed move $s'_{\mathcal{R}_i}$. If the marginal utility increase of M_i's proposal is higher than the marginal utility of any MC&C in $\mathcal{NM}_i(s'_{\mathcal{R}_i})$, MC&C M_i will change its strategy to $s'_{\mathcal{R}_i}$. The complexity of this conflict check for each MC&C is $\mathcal{O}(|\mathcal{M}|)$. Note that it is also possible that the utility of a goal set will increase when multiple MC&Cs change their strategies simultaneously, no matter whether their moves are in conflict with each other. Since an MC&C's changing its strategy will affect the utilities of its neighbor MC&Cs, an MC&C's making the optimal decision of whether to switch to its new strategy $s'_{\mathcal{R}_i}$ may depend on other MC&Cs' choice of whether to change to their strategies.

DEFINITION 1. *(Move selection) Assume that MC&Cs' current strategies are $s_{\mathcal{R}_1}, \ldots, s_{\mathcal{R}_{|\mathcal{M}|}}$, respectively. Assume that MC&Cs are proposing to use new strategies $s'_{\mathcal{R}_1}, \ldots, s'_{\mathcal{R}_{|\mathcal{M}|}}$, respectively. The move selection problem is to find out the set of moves to maximize the social welfare.*

THEOREM 2. *The move selection problem is $\mathcal{NP}$-hard.*

The theorem's proof is a straightforward reduction from the maximum matching problem (omitted due to space limitations). Considering the high complexity of finding MC&Cs' optimal decisions of changing their strategies and the dynamic feature of the system, we adopt the above conflict check approach which has a low complexity since each MC&C only needs to consider the marginal utilities of its neighbor MC&Cs.

Figure 1 shows an MC&C's finite state machine for the distributed negotiation protocol. After receiving data from radars, M_i runs the local optimization algorithm to find its initial strategy. After it sends its strategy to its neighbor MC&Cs, M_i is in the state wstrategy, which implies that M_i is waiting for other MC&Cs to report their current strategies. After M_i has received strategies from all its neighbors or its waiting deadline ξ has reached, it computes its new optimal strategy and notifies its neighbor MC&Cs. Then its state is wproposal which implies that M_i has sent out its move proposal and is waiting for other MC&Cs to report their move proposals. If M_i's marginal utility is higher than the other marginal utilities of conflicting move proposals it has received within the waiting time, it will change its strategy and notify its neighbor MC&Cs. Then its status will be changed to wstrategy. During negotiation, after M_i decides whether to make a move, it will wait for other MC&Cs' strategy update. If the optimization deadline is reached, the state is end and M_i sends out its current scan commands to all the radars under its control.

There are several important control parameters in our approach and we set the values for those control parameters through experimental tuning. MC&C M_i needs to decide its search depth for local multi-step optimization. One obvious rule is that the search depth l should be no larger than $\max_{g\in\mathcal{AG}_i^t} T_e(g)$. Although an MC&C has a better chance to coordinate its future actions with a larger search depth, having a large search depth brings several drawbacks. First, the MC&C's strategy space increases exponentially with l. After generating a strategy over multiple heartbeats at heartbeat t, the MC&C will run the optimization algorithm again at heartbeat $t+1$. That is, the strategy for future heartbeats generated at time t may be abandoned later. Furthermore, as each MC&C has imperfect knowledge about future events due to inaccurate prediction and about the strategies of other MC&Cs, the generated "optimal" strategy over multiple heartbeats may not be optimal in practice. We used a heuristic to decide the search depth for each MC&C by considering a goal's expected existence time. A goal $g \in \mathcal{AG}_i^t$ will exist for $T_e(g) - t + 1$ heartbeats starting from heartbeat t. The average existing time $\sum_{g\in\mathcal{AG}_i^t}(T_e(g) - t + 1)/|\mathcal{AG}_i^t|$ of active goals $\mathcal{AG}_i^t$ is chosen as the search depth. Simulation results show that the heuristic achieved the highest utility compared to other arbitrary approaches (e.g., $l = 1$, $l = \max_{g\in\mathcal{AG}_i^t} T_e(g)$) for setting the search depth.

Two additionally important control parameters for each MC&C are the time τ to run its local optimization and the waiting time ξ during negotiation. With longer time, an MC&C can get a solution closer to the local optimal solution. However, an MC&C may has a short time for negotiation if it spends too much time in local optimization. During negotiation, an MC&C needs to decide how long to wait for the messages from its neighbor MC&Cs. With the increase of waiting time ξ, the negotiation is more synchronous since an MC&C will have more knowledge about its neighbor MC&Cs before making a decision. We found through experiments that it's always better to allocate 6 seconds for each local optimization. When $\tau \ll 6$ seconds, the local optimization solution has a low quality. If $\tau \gg 6$ seconds, there is not much time to do negotiation given 60 seconds heartbeat deadline. Furthermore, we

found that it's always better to set $\xi \sim 4$ seconds. Therefore, MC&Cs may conduct 6 rounds of negotiation and we found that in most cases, negotiation converges (i.e., no MC&C can find a better strategy) in about 5 rounds of negotiation.

Our negotiation scheme has a number of features: 1) Each MC&C exchanges its scan plan (generated by local optimization) over multiple heartbeats with neighbor MC&Cs. 2) To reduce the utility loss due to concurrent strategy change, an MC&C changes its strategy if it has the highest marginal utility than that of its neighbor MC&Cs. 3) Negotiation is conducted asynchronously to increase concurrency of MC&Cs' strategy change. If we synchronize the negotiation protocol by setting a long waiting time, we can guarantee that the social welfare will monotonically increase with the ongoing negotiation as in [3]. We make tradeoffs between speeding up negotiation and guaranteeing monotonic increase of social welfare by setting the value of waiting deadline ξ by considering factors such as the communication delay distribution. When we set a long waiting deadline (i.e., there is no concurrency), the protocol is similar to the LID-JESP algorithm [7] which makes use of the distributed breakout algorithm (DBA) algorithm [13].

6. SIMULATION RESULTS

Evaluating the performance of the approach on the real radar system is difficult and complex. To better quantify the benefits of our approach, we turn to simulation results in more controlled settings.

6.1 Simulator

To determine how best to decentralize control, we have created an abstract simulation of the actual DCAS system. The simulator consists of a number of components. Radars are clustered into partitions, each of which has a single MC&C. Each MC&C has a feature repository where it stores information regarding phenomena in its spacial region, where each phenomenon represents a weather event. Goals are generated following goal generation rules given observed weather phenomena. The optimization function of each MC&C takes its scan goals and returns scans for each of its radars. The simulator additionally contains a function which abstractly simulates the mapping from physical events and scans of the radars to what the MC&C eventually sees as the result of those scans. Depending on the elevations scanned, the number of radars scanning, the type of phenomena, and the speed of scan, it assigns error values to the attributes of the phenomena within certain bounds. In this way, the MC&Cs do not see exactly what is there but rather something slightly off.

The parameters of each phenomenon (e.g., speed, density), each radar (e.g., radius), and each MC&C (e.g., the number of radars under control) reflect the current design of the real system. Phenomena may be either pinpointing or non-pinpointing. Goal generation and utility calculation in the simulator are the same as that in the real system. The radars have a range of approximately 30 kilometers and the optimization has to finish in 60 seconds. The communication delay between MC&Cs is based on the data gathered from the real system. The number of radars ranges between $[8, 100]$ and the number of radars controlled by each MC&C is ranged between $[4, 16]$. Each radar can have at most $\lambda = 8$ candidate strategies at any heartbeat which represent a wide range of strategies.

6.2 Benchmark approaches

Our *distributed negotiation (*$\mathbf{DN}$*)* model was compared with four other different approaches. Both the *centralized single-step optimization (*$\mathbf{CS}$*)* approach and *centralized multi-step optimization (*$\mathbf{CM}$*)* approach assume that there is a super MC&C controlling all the MC&Cs and the super MC&C runs the local optimization algorithm to find optimal strategies for all MC&Cs' radars. The only difference between $\mathbf{CS}$ and $\mathbf{CM}$ is that $\mathbf{CS}$ makes one heartbeat optimization while $\mathbf{CM}$ searches over multiple heartbeats. For the *fully distributed (*$\mathbf{FD}$*) approach*, MC&Cs optimize the utilities of their separate goal sets and don't communicate with each other. $\mathbf{FD}$ searches strategies over multiple heartbeats.

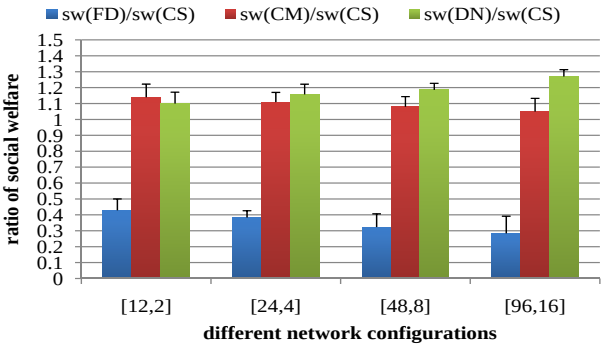


Figure 2: Ratio of social welfare and scale of the network.

6.3 Experimental settings

In our experiments, the simulator will model real phenomena generation and different approaches may generate different set of goals given their observations. Each approach will optimize its scan strategy based on its own goal set. Thus it is unfair to compare the performance of different approaches based on their own goal sets. Instead, we generate an *oracle* goal set based on the system's real phenomena. The social welfare of each approach is evaluated based on the actions generated by each approach and the set of oracle goals generated by the system. For an experiment, we run the system for multiple heartbeats (e.g., 200) and compute the average social welfare for each heartbeat, e.g., average social welfare $sw(\mathbf{DN})$ for the approach $\mathbf{DN}$.

An extensive amount of stochastic simulations was carried out for various resource allocation scenarios subjected to the following variables: 1) the scale of each MC&C, i.e., how many radars are controlled by an MC&C; 2) the density of phenomena, i.e., the frequency of new phenomena entering the radar network; 3) the speed of phenomena; and 4) the ratio of boundary goals. In the rest of this section, we report some representee simulation results.

6.4 Observations

6.4.1 *Scale of the sensor network*

On average, $\mathbf{DN}$ achieved a much higher social welfare than all other benchmark approaches. Figure 2 shows the ratios of the social welfare of approaches $\mathbf{FD}$, $\mathbf{CM}$ and $\mathbf{DN}$ to that of $\mathbf{CS}$ in networks of different scales in which each MC&C controls 6 radars ($[12, 2]$ implies 12 radars with 2 MC&Cs). We can see that 1) $\mathbf{DN}$ achieved slightly lower social welfare than $\mathbf{CM}$ if there are a small number of radars (e.g., 12) and 2) $\mathbf{DN}$ achieved higher social welfare other approaches if there are more than 12 radars and the advantage increases with the scale of the network. This result is intuitive since an MC&C's strategy space increases with the number of radars. Given the real time constraint, distributed optimization with coordination may achieve better performance than centralized optimization. For the two centralized approaches, $\mathbf{CM}$ achieved a higher social welfare than $\mathbf{CS}$ since $sw(\mathbf{CM})/sw(\mathbf{CS})$ is higher than 1.

We can also see that the fully distributed approach $\mathbf{FD}$ achieved much worse performance than other approaches due to lack of coordination. One interesting observation from the experiments is that

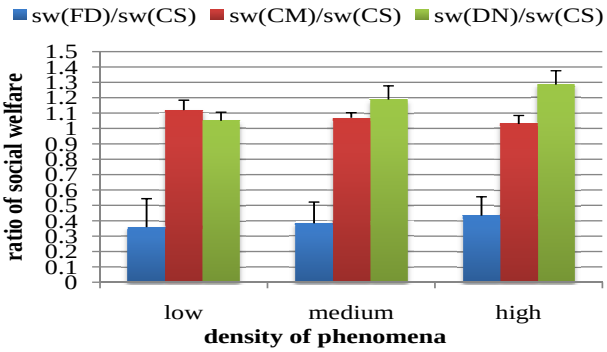


Figure 3: Average social welfare and phenomena density.

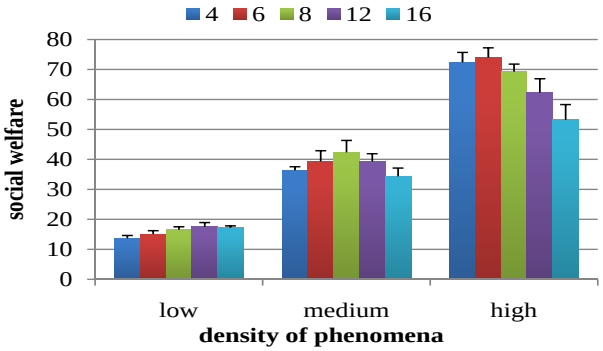


Figure 4: Average social welfare and network structure.

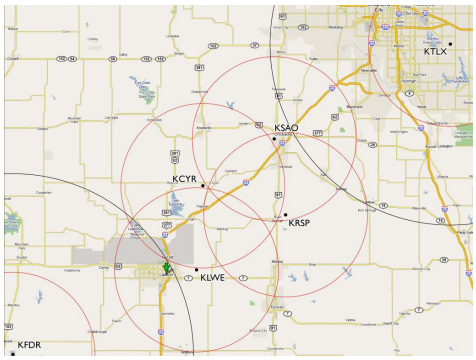


Figure 5: Location of the 4 IP1 radar nodes in Oklahoma.

FD's social welfare based on its *own* goal set is very high. However, **FD**'s social welfare based on the *oracle* goal set is low, which is partially due to *belief propagation*: if an MC&C has wrong belief about the real phenomena, it may still have wrong (or even worse) belief after it sends out scanning commands based on its old wrong belief. Through coordination, MC&Cs will "talk" to each other and accordingly, they may have a more accurate understanding of real phenomena.

6.4.2 *Density of phenomena and network structure*

We found through simulation that the density of phenomena had a large effect on the performance of different approaches. It is intuitive since, with more phenomena, more goals will be generated and the search space of the optimization problem increases. We use the average number η of phenomena per radar at each heartbeat to measure the density of phenomena. For our domain, an η in the range of $[0.5, 1]$ (respectively, $[1, 3]$ and $[3, 6]$) is considered as low (respectively, moderate and high). It can be found from Figure 3 that the advantage of **DN** over the other approaches increases with the increase of the phenomena density. In addition, for different phenomena densities, **CM** achieved a higher social welfare than **CS**, which had a much better performance than **FD**.

One important objective of simulation is to investigate how the performance of **DN** is affected by the network structure, i.e., the number of radars controlled by each MC&C. Intuitively, if an MC&C has to control a large number of radars, it cannot find a good solution given its heartbeat deadline. However, if each MC&C has only a small number of radars, an MC&C can find a local optimal solution but the global solution based on all MC&Cs' local optimal solutions may be much worse than the global optimal solution. Figure 4 shows how the performance of **DN** is affected by the number of radars controlled by each MC&C in a network with 48 radars. It can be found that 1) when the phenomena density is low, it is better to allow each MC&C to control relatively more radars (e.g., 12,); 2) when the phenomena density is medium, it is better to allow each MC&C to control around 8 radars; and 3) when the phenomena density is high, it is better to allow each MC&C to control a small number radars (e.g., 4, 6).

6.4.3 *Speed of phenomena and boundary goals*

We also observed that the advantage of **DN** over the other approaches increases with the increase of the ratio of boundary goals and moving speed of phenomena (figures omitted due to space limitation). With more boundary goals, coordination between MC&Cs becomes more important since it may improve the utilities of these boundary goals by removing redundant scans and having multiple radars to observe the same phenomenon. If a phenomenon moves fast, multiple MC&Cs may need to coordinate with each other to satisfy the goal existing for multiple heartbeats.

7. FIELD STUDY

We have implemented our approach on the IP1 Testbed, which is located in southwestern Oklahoma in the heart of tornado alley. Figure 5 shows the location and coverage area of the testbed. The testbed consists of four mechanically steered parabolic dish X-band radars atop small towers. The circles around KSAO, KCYR, KLWE, and KRSP show the 30 km coverage area of the IP1 radars. The nearest NEXRAD sites located near the IP1 testbed are the radars at Twin Lakes (KTLX) and Frederick (KFDR) and are shown here with 40 km and 60 km range rings. An interested reader can refer to [5, 15] for the IP1 system architecture.

We evaluated our approach during the 2010 CASA Spring Experiment from April 1st to June 15th. This time period corresponds to a yearly maximum of severe storms and tornadic weather in our testbed domain. We reran cases archived during this experiment period from severe weather events using our system emulator which simulates the behavior of the system in a non-closed loop fashion - that is we can verify the behavior of the scan optimization, but the supplied radar data is from the canned case, not from an actual regeneration of data using, for example, a radar simulator. Using this system emulation approach we verified the scanning behavior of the goal-based multi-step optimization.

Figure 6 shows one example of the scanning pattern for each radar from an emulated test case. On the left of Figure 6 shows the scanning actions of radars using our approach and on the right of Figure 6 shows radars' scanning actions using the previous approach described in [3]. Each radar does a pie shaped sector scan, the number of arcs on the edge denoting the number of elevation angles in the scan. It can be found in Figure 6 that the two approaches very often, but not always generated different scanning commands for the radars.

To further test goal-based multi-step optimization versus a baseline functionality of the system we disabled the previous system's time-since-last-scanned scan optimization heuristic and reran test cases where we compared this baseline system versus the fully

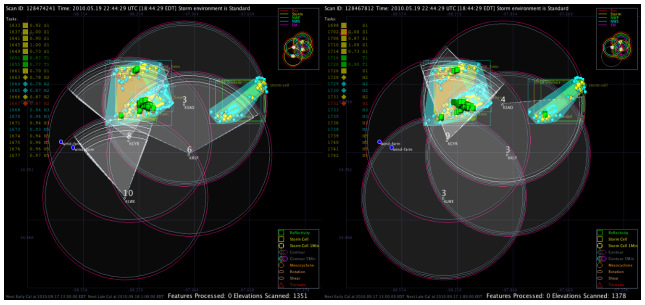


Figure 6: Scanning pattern for each radar. This snapshot was taken from data on May 19th 2010 22:44:29 (UTC).

functional goal-based multi-step optimization. We observed that the myopic baseline configuration generally exhibited a much "greedier" approach to scanning - often performing repeated scans of the same phenomena while ignoring other scan requirements such as satisfying user needs for regular low-level surveillance scans. These results with real data verified our simulation studies and showed that 1) goal based problem formulation more precisely models the needs of multiple end-users and 2) multi-step optimization together with negotiation based coordination efficiently schedules radars' scanning actions over multiple heartbeats.

It has been observed that our approach is significantly better at meeting the user specified "time-since-last-scan" requirements. In addition, our new approach avoided redundant scanning important phenomena and found phenomena failed to be observed by the old approach. Besides the numerical values that were obtained through the analysis of the real-time scanning actions generated by MC&Cs, we also got direct feedback from domain experts informing us that because we were using predictions of the future locations of phenomena we also were doing a better job of scanning the "leading edges" of storm systems. This is an important benefit because most of the interesting observables that lead to better warning by humans are located in these areas.

8. CONCLUSION

This paper presents a distributed resource allocation model combining heuristic search and asynchronous negotiation. In more detail, our contributions to the state of the art include:

- We introduce the concept "goal" to model end-users' preferences over multiple heartbeats and cast the complex sensing resource allocation problem as a continuous time optimization problem. The goal based formulation enhances modularity and improves the adaptivity of our approach to changing environments and user preferences. Each MC&C utilizes a genetic algorithm to find its local optimal strategy over multiple heartbeats given its neighbor MC&Cs' current strategies.
- We extended the distributed negotiation model in [3] by allowing MC&Cs to 1) exchange "plans" over multiple heartbeats and 2) make tradeoffs regarding local optimization time, negotiation time, and concurrency.
- We empirically show that our approach achieved better performance than some benchmark approaches.
- We have applied our approach to an operational radar testbed that is deployed in Oklahoma to observe severe weather events and it has exhibited much better performance than previous techniques.

Future research directions include improving the distributed resource allocation model. For instance, MC&Cs can make multilateral agreement through mediation that allows neighboring MC&Cs to make moves concurrently. It is also possible that the negotiation stops with a local optimal solution and it may be beneficial to accept some poor agreements to help in the long run. Our ongoing research will also focus on applying this framework to other large scale real-time optimization problems.

9. ACKNOWLEDGMENTS

We thank Michael Krainin and Chongjie Zhang for their help with building the simulator and designing the genetic algorithm. This work was supported primarily by the Engineering Research Centers Program of the National Science Foundation under NSF Cooperative Agreement No. EEC-0313747. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

10. REFERENCES

[1] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[2] Y. Kim, M. Krainin, and V. Lesser. Application of max-sum algorithm to radar coordination and scheduling. In *Proc. of the Twelfth International Workshop on Distributed Constraint Reasoning*, pages 5–19, 2010.

[3] M. Krainin, B. An, and V. Lesser. An application of automated negotiation to distributed task allocation. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 138–145, Nov. 2007.

[4] V. Lesser, C. Ortiz, and M. Tambe, editors. *Distributed Sensor Networks: A Multiagent Perspective*, volume 9. Kluwer Academic Publishers, May 2003.

[5] D. McLaughlin and Coauthors. Short-wavelength technology and the potential for distributed networks of small radar systems. *Bulletin of the American Meteorological Society*, 90:1797–1817, 2009.

[6] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, 2004.

[7] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: a synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 133–139, 2005.

[8] A. Rogers, D. Corkill, and N. Jennings. Agent technologies for sensor networks. *IEEE Intelligent Systems*, 24(2):13 –17, 2009.

[9] E. Ruzanski, Y. Wang, and V. Chandrasekar. Development of a real-time dynamic and adaptive nowcasting system. In *Proc. of the International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, 2009.

[10] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. *Eleventh National Conference on Artificial Intelligence*, pages 256–262, January 1993.

[11] L.-K. Soh and C. Tsatsoulis. Reflective negotiating agents for real-time multisensor target tracking. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1121–1127, 2001.

[12] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In *Proc. of the 21st international jont conference on Artifical intelligence*, pages 299–304, 2009.

[13] M. Yokoo and K. Hirayama. Distributed Breakout Algorithm for Solving Distributed Constraint Satisfaction Problems. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 401–408, 1996.

[14] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87, 2005.

[15] M. Zink, E. Lyons, D. Westbrook, J. Kurose, and D. Pepyne. Closed-loop architecture for distributed collaborative adaptive sensing of the atmosphere: Meteorological command & control. *International Journal for Sensor Networks*, 6(4), 2009.

Integrating Reinforcement Learning with Human Demonstrations of Varying Ability

Matthew E. Taylor, Halit Bener Suay, and Sonia Chernova
Lafayette College, taylorm@lafayette.edu
Worcester Polytechnic Institute, {benersuay, soniac}@wpi.edu

ABSTRACT

This work introduces *Human-Agent Transfer* (HAT), an algorithm that combines transfer learning, learning from demonstration and reinforcement learning to achieve rapid learning and high performance in complex domains. Using experiments in a simulated robot soccer domain, we show that human demonstrations transferred into a baseline policy for an agent and refined using reinforcement learning significantly improve both learning time and policy performance. Our evaluation compares three algorithmic approaches to incorporating demonstration rule summaries into transfer learning, and studies the impact of demonstration quality and quantity, as well as the effect of combining demonstrations from multiple teachers. Our results show that all three transfer methods lead to statistically significant improvement in performance over learning without demonstration. The best performance was achieved by combining the best demonstrations from two teachers.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Miscellaneous

General Terms

Algorithms, Performance

Keywords

Reinforcement Learning, Learning from Demonstration, Human/Agent Interaction, Transfer Learning

1. INTRODUCTION

Agent technologies for virtual agents and physical robots are rapidly expanding in industrial and research fields, enabling greater automation, increased levels of efficiency, and new applications. However, existing systems are designed to provide niche solutions to very specific problems and each system may require significant effort to develop. The ability to acquire new behaviors through learning is fundamentally important for the development of general-purpose agent platforms that can be used for a variety of tasks.

Existing approaches to agent learning generally fall into two categories: independent learning through exploration and learning from labeled training data. Agents often learn independently from exploration via *Reinforcement learning* (RL) [25]. While such techniques have had great success in offline learning and software applications, the large amount of data and high exploration times they require make them intractable for most real-world domains.

On the other end of the spectrum are *learning from demonstration* (LfD) algorithms [1]. These approaches leverage the vast experience and task knowledge of a person to enable fast learning, which is critical in real-world applications. However, human teachers provide particularly noisy and suboptimal data due to differences in embodiment (e.g., degrees of freedom, action speed) and limitations of human ability. As a result, final policy performance achieved by these methods is limited by the quality of the dataset and the performance of the teacher.

This paper proposes a novel approach: use RL *transfer learning* methods [28] to combine LfD and RL and achieve both fast learning and high performance in complex domains. In transfer learning, knowledge from a *source task* is used in a *target task* to speed up learning. Equivalently, knowledge from a source agent is used to speed up learning in a target agent. For instance, knowledge has been successfully transferred between agents that balance different length poles [19], that solve a series of mazes [5, 34], or that play different soccer tasks [29, 31, 32]. The key insight of transfer learning is that previous knowledge can be effectively reused, even if the source task and target task are not identical. This results in substantially improved learning times because the agent no longer relies on an uninformed (arbitrary) prior.

In this work, we show that we can effectively transfer knowledge from a human to an agent, even when they have different perceptions of state. Our method, *Human-Agent Transfer* (HAT): 1) allows a human teacher to perform a series of demonstrations in a task, 2) uses an existing transfer learning algorithm, *Rule Transfer* [27], to learn rule-based summaries of the demonstration, and 3) integrates the rule summaries into RL, biasing learning while also allowing improvement over the transferred policy.

We perform empirical evaluation of HAT in a simulated robot soccer domain. We compare three algorithms for incorporating rule summaries into reinforcement learning, and compare learning performance for multiple demonstration source, quantity, and quality conditions. Our findings show statistically significant improvement in performance for all variants of HAT over learning with no prior. Additionally, we find that exposure even to suboptimal demonstration training data results in significant improvements over random exploration, and combining demonstrations from multiple teachers leads to the best performance.

2. BACKGROUND

This section provides background on the three key techniques discussed in this paper: reinforcement learning, learning from demonstrations, and transfer learning.

Cite as: Integrating Reinforcement Learning with Human Demonstrations of Varying Ability, Matthew E. Taylor, Halit Bener Suay, and Sonia Chernova, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 617-624.

2.1 Reinforcement Learning

Reinforcement learning is a common approach to agent learning from experience. We define reinforcement learning using the standard notation of Markov decision processes (MDPs) [16]. At every time step the agent observes its state $s \in S$ as a vector of k state variables such that $s = \langle x_1, x_2, \ldots, x_k \rangle$. The agent selects an action from the set of available actions A at every time step. An MDP's reward function $R : S \times A \mapsto \mathbb{R}$ and (stochastic) transition function $T : S \times A \mapsto S$ fully describe the system's dynamics. The agent will attempt to maximize the long-term reward determined by the (initially unknown) reward and transition functions.

A learner chooses which action to take in a state via a policy, $\pi : S \mapsto A$. Policy π is modified by the learner over time to improve performance, which is defined as the expected total reward. Instead of learning π directly, many RL algorithms instead approximate the action-value function, $Q : S \times A \mapsto \mathbb{R}$, which maps state-action pairs to the expected real-valued return. In this paper, agents learn using Sarsa [17, 20], a well known but relatively simple temporal difference RL algorithm, which learns to estimate $Q(s, a)$. While some RL algorithms are more sample efficient than Sarsa, this paper will focus on Sarsa for the sake of clarity.

Although RL approaches have enjoyed multiple past successes (e.g., TDGammon [30], inverted Helicopter control [12], and agent locomotion [18]), they frequently take substantial amounts of data to learn a reasonable control policy. In many domains, collecting such data may be slow, expensive, or infeasible, motivating the need for ways of making RL algorithms more sample-efficient.

2.2 Learning from Demonstration

Learning from demonstration research explores techniques for learning a policy from examples, or demonstrations, provided by a human teacher. LfD can be seen as a subset of Supervised Learning, in that the agent is presented with labeled training data and learns an approximation to the function which produced the data.

Similar to reinforcement learning, learning from demonstration can be defined in terms of the agent's observed state $s \in S$ and executable actions $a \in A$. Demonstrations are recorded as temporal sequences of t state-action pairs $\{(s_0, a_0), ..., (s_t, a_t)\}$, and these sequences typically only cover a small subset of all possible states in a domain. The agent's goal is to generalize from the demonstrations and learn a policy $\pi : S \mapsto A$ covering all states that imitates the demonstrated behavior.

Many different algorithms for using demonstration data to learn π have been proposed. Approaches vary by how demonstrations are performed (e.g., teleoperation, teacher following, kinesthetic teaching, external observation), the type of policy learning method used (e.g., regression, classification, planning), and assumptions about degree of demonstration noise and teacher interactivity [1]. Across these differences, LfD techniques possess a number of key strengths. Most significantly, demonstration leverages the vast task knowledge of the human teacher to significantly speed up learning either by eliminating exploration entirely [6, 13], or by focusing learning on the most relevant areas of the state space [22]. Demonstration also provides an intuitive programming interface for humans, opening possibilities for policy development to non-agents-experts.

However, LfD algorithms are inherently limited by the quality of the information provided by the human teacher. Algorithms typically assume the dataset to contain high quality demonstrations performed by an expert. In reality, teacher demonstrations may be ambiguous, unsuccessful, or suboptimal in certain areas of the state space. A naïvely learned policy will likely perform poorly in such areas [2]. To enable the agent to improve beyond the performance of the teacher, learning from demonstration must be combined with learning from experience.

Most similar to our approach is the work of Smart and Kaelbling, which shows that human demonstration can be used to bootstrap reinforcement learning in domains with sparse rewards by initializing the action-value function using the observed states, actions and rewards [22]. In contrast to this approach, our work uses demonstration data to learn generalized rules, which are then used to bias the reinforcement learning process.

2.3 Transfer Learning

The insight behind *transfer learning* (TL) is that generalization may occur not only within tasks, but also *across tasks*, allowing an agent to begin learning with an informative prior instead of relying on random exploration.

Transfer learning methods for reinforcement learning can transfer a variety of information between agents. However, many transfer methods restrict what type of learning algorithm is used by both agents (for instance, some methods require temporal difference learning [29] or a particular function approximator [32] to be used in both agents). However, when transferring from a human, it is impossible to copy a human's "value function" — both because the human would likely be incapable of providing a complete and consistent value function, and because the human would quickly grow wary of evaluating a large number of state-action pairs.

This paper uses *Rule Transfer* [27], a particularly appropriate transfer method that is agnostic to the knowledge representation of the source learner. The ability to transfer knowledge between agents that have different state representations and/or actions is a critical ability when considering transfer of knowledge between a human and an agent. The following steps summarize Rule Transfer:

1a: Learn a policy ($\pi : S \mapsto A$) in the source task. Any type of reinforcement learning algorithm may be used.

1b: Generate samples from the learned policy After training has finished, or during the final training episodes, the agent records some number of interactions with the environment in the form of (S, A) pairs while following the learned policy.

2: Learn a decision list ($D_s : S \mapsto A$) that summarizes the source policy. After the data is collected, a propositional rule learner is used to summarize the collected data to approximate the learned policy by mapping states to actions.[1] This decision list is used as a type of inter-lingua, allowing the following step to be independent of the type of policy learned (step 1a).

3: Use D_t to bootstrap learning of an improved policy in the target task. For instance, previous work [27] provided three ways of leveraging this knowledge; two of these methods are discussed later in Sections 3.1 and 3.2.

2.4 Additional Related Work

This section briefly summarizes three additional lines of related work.

Within psychology, *behavioral shaping* [21] is a training procedure that uses reinforcement to condition the desired behavior in a human or animal. During training, the reward signal is initially

[1]Additionally, if the agents in the source and target task use different state representations or have different available actions, the decision list can be translated via inter-task mappings [27, 29] (as step 2b). For the current paper, this translation is not necessary, as the source and target agents operate in the same task.

used to reinforce any tendency towards the correct behavior, but is gradually changed to reward successively more difficult elements of the task. Shaping methods with human-controlled rewards have been successfully demonstrated in a variety of software agent applications [3, 7]. An alternate form of shaping is to change the task over time, or construct a task sequence for an agent to train on [26, 36]. In contrast to shaping, LfD allows a human to demonstrate complete behaviors, which may contain much more information than a sequence of rewards or suggested tasks.

Most similar to our approach is the recent work by Knox and Stone [9] which combines shaping with reinforcement learning. Their TAMER [8] system learns to predict and maximize a reward that is interactively provided by a human. The learned human reward is combined in various ways with Sarsa(λ), providing significant improvements. The primary difference between HAT and this method is that we focus on leveraging human demonstration, rather than estimating and integrating a human reinforcement signal.

The idea of transfer between a human and an agent is somewhat similar to *implicit imitation* [15], in that one agent teaches another how to act in a task, but HAT does not require the agents to have the same (or very similar) representations.

Allowing for such shifts in representation gives additional flexibility to an agent designer; past experience may be transferred rather than discarded if a new representation is desired. Representation transfer is similar in spirit to HAT in that both the teacher and the learner function in the same task, but very different techniques are used since the human's "value function" cannot be directly examined.

High-level advice and suggestions have also been used to bias agent learning. Such advice can provide a powerful learning tool that speeds up learning by biasing the behavior of an agent and reducing the policy search space. However, existing methods typically require either a significant user sophistication (e.g., the human must use a specific programming language to provide advice [11]) or significant effort is needed to design a human interface (e.g., the learning agent must have natural language processing abilities [10]). Allowing a teacher to demonstrate behaviors is preferable in domains where demonstrating a policy is a more natural interaction than providing such high-level advice.

3. METHODOLOGY

In this section we present HAT, our approach to combining LfD and RL. HAT consists of three steps, motivated by those used in Rule Transfer:

Demonstration The agent performs the task under the teleoperated control by a human teacher, or by executing an existing suboptimal controller. During execution, the agent records all state-action transitions. Multiple task executions may be performed.

Policy Summarization HAT uses the state-action transition data recorded during the Demonstration phase to derive rules summarizing the policy. These rules are used to bootstrap autonomous learning.

Independent Learning The agent learns independently in the task via reinforcement learning, using the policy summary to bias its learning. In this step, the agent must balance exploiting the transferred rules with attempting to learn a policy that outperforms the transferred rules.

In contrast to transfer learning, HAT assumes that either 1) the demonstrations are executed on the same agent, in the same task, as will be learned in the Independent Learning phase, or that 2) any differences between the agent or task in the demonstration phase are small enough that they can be ignored in the independent learning phase. Instead of transferring between different tasks, HAT focuses on transferring between different agents with different internal representations. For instance, it is not possible to directly use a human's "value function" inside an agent because 1) the human's knowledge is not directly accessible and 2) the human has a different state abstraction than the agent.

We next present three different ways that HAT can use a decision list to improve independent learning.

3.1 Value Bonus

The intuition behind the *Value Bonus* method [27] is similar to that of shaping in that the summarized policy is used to add a reward bonus to certain human-favored actions. When the agent reaches a state and calculates $Q(s, a)$, the Q-value of the action suggested by the summarized policy is given a constant bonus (B). For the first C episodes, the learner is forced to execute the action suggested by the rule set. This is effectively changing the initialization of the Q-value function, or, equivalently [33], providing a shaping reward to the state-action pairs that are selected by the rules.

We use $B = 10$ and $C = 100$ to be consistent with past work [27]; the Q-value for the action chosen by the summarized policy will be given a bonus of +10 and agents must execute the action chosen by the summarized policy for the first 100 episodes.

3.2 Extra Action

The *Extra Action* method [27] augments the agent so that it can select a *pseudo-action*. When the agent selected this pseudo-action, it executed the action suggested by the decision list. The agent may either execute the action suggested by the transferred rules, or it can execute one of the "base" MDP actions. Through exploration, the RL agent can decide when it should 1) follow the transferred rules by executing the pseudo-action or 2) execute a base MDP action (e.g., the transferred rules are sub-optimal). Were the agent to always execute the pseudo-action, the agent would never learn but would simply mimic the demonstrated policy.

As with the Value Bonus algorithm, the agent initially executes the action suggested by the decision list, allowing it to estimate the value of the decision list policy. We again set this period to be 100 episodes ($C = 100$).

3.3 Probabilistic Policy Reuse

The third method used is *Probabilistic Policy Reuse*, based on the π-reuse Exploration Strategy [4, 5]. In Probabilistic Policy Reuse, the agent will reuse a policy with probability ψ, explore with probability ϵ, and exploit the current value function with probability $1 - \psi - \epsilon$. By decaying ψ over time, the agent can initially leverage the decision list, but then learn to improve on it if possible. Note that Probabilistic Policy Reuse is similar to the recent TAMER+RL method #7 [9], where the agent tries to execute the action suggested by the learned human shaping reward, rather than follow a transferred policy.

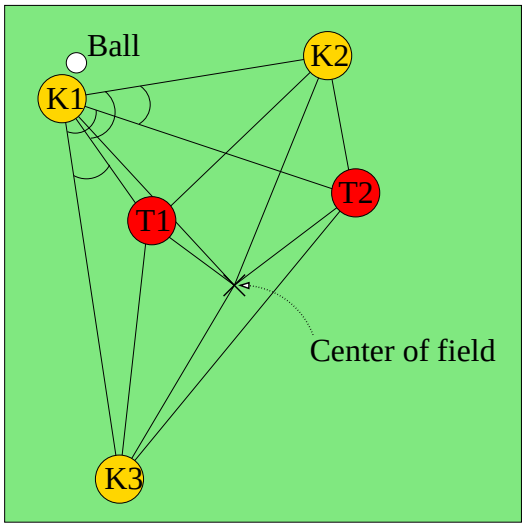


Figure 1: This diagram shows the distances and angles used to construct the 13 state variables used for learning with 3 keepers and 2 takers. Relevant objects are the 3 keepers (K) and the two takers (T), both ordered by distance from the ball, and the center of the field.

Figure 2: This figure shows a screenshot of the visualizer used for the human to demonstrate a policy in 3 vs. 2 Keepaway. The human controls the keeper with the ball (shown as a hollow white circle) by telling the agent when, and to whom, to pass. When no input is received, the keeper with the ball executes the `Hold` action, attempting to maintain possession of the ball.

4. EXPERIMENTAL VALIDATION

This section first discusses Keepaway [24], a simulated robot soccer domain and then explains the experimental methodology used to evaluate HAT.

4.1 Keepaway

Keepaway is a domain with a continuous state space and significant amounts of noise in the agent's actions and sensors. One team, the *keepers*, attempts to maintain possession of the ball within a 20m $\times$ 20m region while another team, the *takers*, attempts to steal the ball or force it out of bounds. The simulator places the players at their initial positions at the start of each episode and ends an episode when the ball leaves the play region or is taken away from the keepers.

The keeper with the ball has the option to either pass the ball to one of its two teammates or to hold the ball. In *3 vs. 2 Keepaway* (3 keepers and 2 takers), the state is defined by 13 hand-selected state variables (see Figure 1) as defined in [24]. The reward to the learning algorithm is the number of time steps the ball remains in play after an action is taken. The keepers learn in a constrained policy space: they have the freedom to decide which action to take only when in possession of the ball. Keepers not in possession of the ball are required to execute the `Receive` macro-action in which the player who can reach the ball the fastest goes to the ball and the remaining players follow a handcoded strategy to try to get open for a pass.

For policy learning, the Keepaway problem is mapped onto the discrete-time, episodic RL framework. As a way of incorporating domain knowledge, the learners choose not from the simulator's primitive actions but from a set of higher-level macro-actions implemented as part of the player [24]. These macro-actions can last more than one time step and the keepers have opportunities to make decisions only when an on-going macro-action terminates. Keepers can choose to `Hold` (maintain possession), $\texttt{Pass}_1$ (pass to the closest teammate), and $\texttt{Pass}_2$ (pass to the further teammate). Agents then make decisions at discrete time steps (when macro-actions are initiated and terminated).

To learn Keepaway with Sarsa, each keeper is controlled by a separate agent. Many kinds of function approximation have been successfully used to approximate an action-value function in Keepaway, but a Gaussian Radial Basis Function Approximation (RBF) has been one of the most successful [23]. All weights in the RBF function approximator are initially set to zero; every initial state-action value is zero and the action-value function is uniform. Experiments in this paper use the public versions 11.1.0 of the RoboCup Soccer Server [14], and 0.6 of UT-Austin's Keepaway players [23].

4.2 Experimental Setup

In the Demonstration phase of HAT, Keepaway players in the simulator are controlled by the teacher using the keyboard. This allows a human to watch the visualization and instruct the keeper with the ball to execute the `Hold`, $\texttt{Pass}_1$, or $\texttt{Pass}_2$ actions. During demonstration, we record all (s, a) pairs selected by the teacher. It is worth noting that the human has a very different representation of the state than the learning agent. Rather than observing a 13 dimensional state vector like the RL agent, the human uses a visualizer (Figure 2). It is therefore critical that whatever method used to glean information about the human's policy does not require the agent and the human to have identical representations of state.

To be consistent with past work [23], our Sarsa learners use $\alpha = 0.05$, $\epsilon = 0.10$, and RBF function approximation. After conducting initial experiments with five values of ψ, we found that $\psi = 0.999$ was at least as good as other possible settings. In the Policy Summarization Phase, we use a simple propositional rule learner to generate a decision list summarizing the policy (that is, it learns to generalize which action is selected in every state). For these experiments, we use JRip, as implemented in Weka [35].

Finally, when measuring speedup in RL tasks, there are many possible metrics. In this paper, we measure the success of HAT along three related dimensions. The initial performance of an agent in a target task may be improved by transfer. Such a *jumpstart* (relative to the initial performance of an agent learning without the benefit of any prior information), suggests that transferred information is immediately useful to the agent. In Keepaway, the jumpstart

is measured as the average episode reward (corresponding to the average episode length in seconds), averaged over 1,000 episodes without learning. The jumpstart is a particularly important metric when learning is slow and/or expensive.

The *final reward* acquired by the algorithm at the end of the learning process (at 30 simulator hours in this paper) indicates the best performance achieved by the learner. This value is computed by taking the average of the final 1,000 episodes to account for the high degree of noise in the Keepaway domain.

The *total reward* accumulated by an agent (i.e., the area under the learning curve) may also be improved. This metric measures the ability of the agent to continue to learn after transfer, but is heavily dependent on the length of the experiment. In Keepaway, the total reward is the sum of the average episode durations at every integral hour of training:

$$\sum_{t:0\rightarrow n} (\text{average episode reward at training hour } t)$$

where the experiment lasts n hours and each average reward is computed by using a sliding window over the past 1,000 episodes.[2]

5. EMPIRICAL EVALUATION

This section presents results showing that HAT can effectively use human demonstration to bootstrap RL in Keepaway agents.

To begin, we recorded a demonstration from a teacher (*Subject A*) which lasted for 20 episodes (less than 3 minutes). Next, we used JRip to summarize the policy with a decision list. The following rules were learned, where $state_k$ represents the k^{th} state variable, as defined in the keepaway task [23]:

if $(state_{11} \geq 74.84$ *and* $state_3 \leq 5.99$ *and* $state_{11} \leq 76.26)$ $\rightarrow$ *Action* $= 1$

elseif $(state_{11} \geq 53.97$ *and* $state_4 \leq 5.91$ *and* $state_0 \geq 8.45$ *and* $state_8 \leq 7.06)$ $\rightarrow$ *Action* $= 1$

elseif $(state_3 \leq 4.84$ *and* $state_0 \geq 7.33$ *and* $state_{12} \geq 43.66$ *and* $state_8 \leq 5.57)$ $\rightarrow$ *Action* $= 2$

else $\rightarrow$ *Action* $= 0$

While not the focus of this work, we found it interesting that the policy was able to be summarized with only four rules, obtaining over 87% accuracy on when using stratified cross-validation.

Finally, agents are trained in 3 vs. 2 Keepaway without using transfer rules (No Prior), using the Value Bonus, using the Extra Action, or using the Probabilistic Policy Reuse method. All learning algorithms were executed for 30 simulator hours (processor running time of roughly 2.5 hours) to ensure convergence.

Figure 3 compares the performance of the four methods, averaged over 10 independent trials. Using 20 episodes of transferred data from *Subject A* with HAT can improve the jumpstart, the final reward, and the cumulative reward. The horizontal line in the figure shows the average duration of the teacher's demonstration episodes; all four of the RL-based learning methods improve upon and outperform the human teacher. The performance of the different algorithms is measured quantitatively in Table 1, where significance is tested with a Student's t-test.

[2]Recall that the reward in Keepaway is +1 per time step, where a time step is a 10th of a simulator second. Thus, the reward for the first hour of training is always $60 \times 60 \times 10 = 36000$ — a metric for the total reward over time must account for the reward *per episode* and simply summing the total amount of reward accrued is not appropriate.

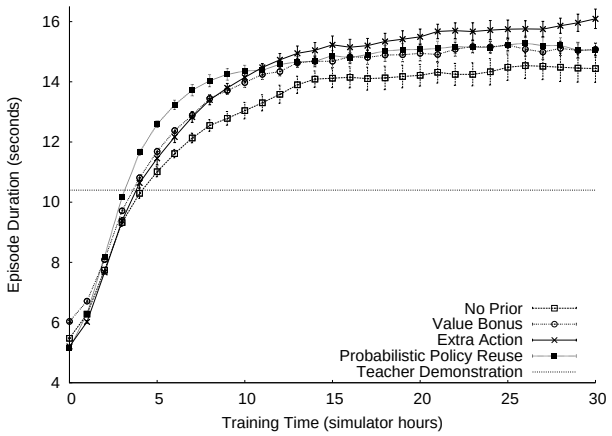


Figure 3: This graph summarizes performance of Sarsa learning in Keepaway using four different algorithms. One demonstration of 20 episodes was used for all three HAT learners. Error bars show the standard error in the performance.

Method	Jumpstart	Final	Total Reward
No Prior	N/A	14.3	380
Value Bonus	**0.57**	15.1	**401**
Extra Action	**-0.29**	**16.0**	**407**
Probabilistic Policy Reuse	**-0.30**	15.2	**411**

Table 1: This table shows the jumpstart, final reward and total reward metrics for Figure 3. Values in bold have statistically significant differences in comparison to the No Prior method ($p < 0.05$).

While the final reward performance of the all four methods is very similar (only Extra Action has a statistically significant[3] improvement over No Prior), the total reward accumulated by all three algorithms is significantly higher than with No Prior learning. This result is an indication that although the same final performance is achieved in the long term because the learning algorithm is able to learn the task in all cases, high performance is achieved *faster* by using a small number of demonstrations. This difference can be best observed by selecting an arbitrary threshold of episode duration and comparing the number of simulation hours each algorithm takes to achieve this performance. In the case of a threshold of 14 seconds, we see that No Prior learning takes 13.5 hours, compared to 10.1, 8.57 and 7.9 hours for Value Bonus, Extra Action and Probabilistic Policy Reuse respectively. These results show that transferring information via HAT from the human results in significant improvements over learning without prior knowledge.

Section 5.1 will explore how performance changes with different types or amounts of demonstration, while Section 5.2 discusses how teacher ability affects learning performance. In all further experiments we use the Probabilistic Policy Reuse method as it was not dominated by either of the other two methods. Additionally, in some trials with other methods we found that the learner could start with a high jumpstart but fail to improve as much as other trials. We posit this is due to becoming stuck in a local minimum. However, because ψ explicitly decays the effect from the rules, this phenomena was never observed when using Probabilistic Policy Reuse.

[3]Throughout this paper, t-tests are used to calculate significance, defined as $p < 0.05$.

5.1 Comparison of Different Teachers

Above, we used a single demonstration data set to evaluate and compare three algorithms for incorporating learned rules into reinforcement learning. In this section, we examine how demonstrations from different people impact learning performance of a single algorithm, Probabilistic Policy Reuse. Specifically, we compare three different teachers:

1. *Subject A*: This teacher has many years of research experience with the Keepaway task. (The same as Figure 3.)
2. *Subject B*: This teacher is new to Keepaway, but practiced for approximately 100 games before recording demonstrations.
3. *Subject C*: This teacher is an expert in LfD, but is new to Keepaway. The teacher practiced 10 games before recording demonstrations.

Each teacher recorded 20 demonstration episodes while trying to play Keepaway to the best of their ability. Figure 4 summarizes the results and compares performance of using these three demonstration sets against learning the Keepaway task without a prior. All reported results are averaged over 10 learning trials. Table 2 presents summary of the results, highlighting statistically significant changes in bold.

Method	Jumpstart	Final	Total Reward
No Prior	N/A	14.3	380
Subject A	**-0.30**	15.2	**411**
Subject B	**3.35**	**15.7**	**423**
Subject C	**0.15**	**16.2**	**424**

Table 2: This table shows the jumpstart, final reward and total reward metrics for Figure 4, where all HAT methods use Probabilistic Policy Reuse with 20 episodes of demonstrated play. Values in bold have statistically significant differences in comparison to the No Prior method.

All three HAT experiments outperformed learning without a bias from demonstration, with statistically significant improvements in total reward. However, as in any game, different Keepaway players have different strategies. While some prefer to keep the ball in one location as long as possible, others pass frequently between keepers. As a result, demonstrations from three different teachers led to different learning curves. Demonstration data from *Subjects A* and *C* resulted in a low jumpstart, while *Subject B*'s demonstration gave the learner a significant jumpstart early in the learning process. The final reward also increased for all three HAT trials, with statistically significant results in the case of *Subjects B* and *C*. These results indicate that HAT is robust to demonstrations from different people with varying degrees of task expertise.

An important factor to consider with any algorithm that learns from human input, is whether combining demonstrations from two or more different teachers helps the agent to learn faster, or whether exposure to possibly conflicting demonstrations from different teachers slows the learning process. In the following evaluation we compared five demonstration types:

1. *Subject A (20)*: Set of the original 20 demonstrations by Subject A: average duration of 10.4 seconds/episode
2. *Subject A (10)*: Set of 10 randomly selected demonstrations by Subject A: average duration 7.5 seconds/episode
3. *Subject C (20)*: Set of the original 20 demonstrations by Subject C : average duration of 11.3 seconds/episode

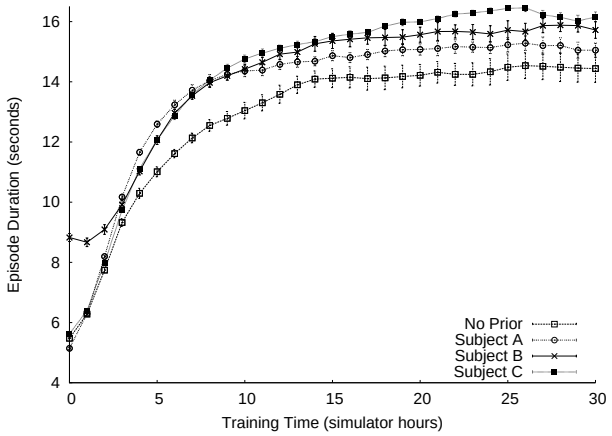


Figure 4: This graph summarizes performance of no prior learning and Probabilistic Policy Reuse learning using demonstrations from three different teachers. Each teacher performed demonstrations for 20 episodes. Error bars show the standard error in performance across 10 trials.

4. *Subjects A + C Best (20)*: The 10 best (longest) demonstration episodes each from Subjects A and C: average duration of 17.2 and 18.0 seconds/episode, respectively
5. *Subjects A + C Worst (20)*: The 10 worst (shortest) demonstration episodes each from Subjects A and C: average duration of 4.6 seconds/episode for both

This analysis provides insight about the impact of combining demonstrations from multiple teachers (conditions 1 and 3 vs. 4 and 5) and the impact of demonstration quantity (condition 1 vs. 2) and quality (condition 4 vs. 5). Figure 5 presents a comparison of the five learning conditions, and Table 3 summarizes the results.

Method	Jumpstart	Final	Total Reward
Subject A (20)	**-0.30**	15.2	**411**
Subject A (10)	**-2.23**	**15.8**	**407**
Subject C (20)	**0.15**	**16.2**	**424**
Subjects A + C Best	**2.15**	**15.7**	**431**
Subjects A + C Worst	**0.37**	**16.1**	**419**

Table 3: This table shows the jumpstart, final reward and total reward metrics for Figure 5, where all HAT methods use Probabilistic Policy Reuse with 20 demonstrated episodes. Values in bold have statistically significant differences in comparison to the No Prior method (not shown).

With respect to learning from multiple teachers, results show that combining data from different subjects leads to performance as good as or better than learning from a single teacher. Condition *Subjects A + C Best* performs better than either *Subject A* or *Subject C* alone, and significantly outperforms all other methods in the group, in large part due to the early lead it has due to its high jumpstart. Condition *Subjects A + C Worst* shows no statistically significant change in performance between it and learning from *Subject A* or *Subject C* alone.[4] This result is significant because it indicates that while quality is important, as shown by the

[4]Note that because we have few subjects, our claims of significance are limited to results from demonstrations with the three subjects tested. Future work will generalize our findings by considering many more subjects.

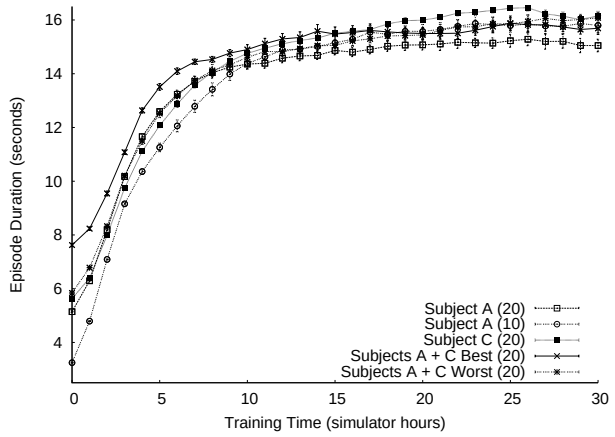


Figure 5: This graph summarizes performance of Probabilistic Policy Reuse learning using five different demonstration sets. Error bars show the standard error in performance across 10 trials.

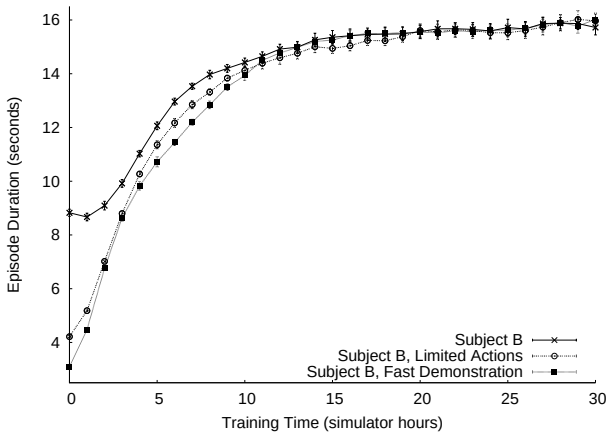


Figure 6: This graph summarizes performance of Probabilistic Policy Reuse learning using three sets of demonstrations from Subject B recorded under different simulator conditions: normal, fast and with limited actions. Each demonstration set consists of 20 episodes. Error bars show the standard error in performance across 10 trials.

difference between *Subjects A + C Best* and *Worst*, any demonstration is beneficial. The fact that the worst demonstrations still lead to performance well above *No Prior* learning is an indication that exposure to any training data is better than random exploration.

In fact, quantity of demonstration may matter more than quality, as shown by the comparison of conditions 1 and 2. Reducing the number of demonstrations by half resulted in a significant decrease in jumpstart. Although performance eventually recovered to achieve a final reward comparable to that of the other methods, achieving that result took longer and there is a statistically significant difference between the total reward of the two conditions.

Most significantly, we highlight that all demonstration-based methods, regardless of data source, quantity or quality, resulted in statistically significant performance improvements over *No Prior* learning. This critical result indicates that HAT learning can benefit from variable degrees of demonstration quality. The algorithm does not require the teacher to be a task expert and easily surpasses the performance of the teacher. In the following section, we further explore the effects of suboptimal demonstrations.

5.2 Impact of Teacher Ability on Learning

In the above experiments, all three teachers demonstrated the task to their best ability. In this evaluation, we alter the simulation environment to make the teacher's demonstrations inherently suboptimal. Specifically, we compare three types of demonstration:

1. *Subject B*: Same as above: average duration 10.5 sec./episode
2. *Subject B Fast*: Simulator speed during training was increased to approximately 5 times faster than real time: average duration 4.3 seconds/episode
3. *Subject B Limited Actions*: The teacher was limited to executing only two actions, `Hold` and $\texttt{Pass}_1$, disallowing passes to the further keeper: average duration 5.2 seconds/episode

The two test conditions are designed to handicap the teacher and reduce the quality of demonstrations, either by affecting reaction time (*Subject B Fast*) or by providing the learning agent with demonstrations of only a subset of the state/action space (*Subject B Limited Actions*). The handicapping effects were successful, reducing the average duration of the teacher's demonstration episodes by more than half.

Figure 6 presents a comparison of the three learning conditions and Table 4 summarizes the results. Importantly, we see again that poor teacher performance does not negatively impact the final performance of the agent. The data further supports our earlier findings that in the long-term, Probabilistic Policy Reuse can learn the task regardless of the initialization method, and there is no statistically significant difference in final reward values between conditions 1 and 2, and conditions 1 and 3. Statistically significant differences are observed, however, in the rate of learning, both with respect to jumpstart and total reward, indicating that suboptimal demonstrations slow the learning process. However, even with the added handicaps, learning from human data shows statistically significant improvements over *No Prior* learning.

Method	Jumpstart	Final	Total Reward
Subject B	**3.35**	**15.7**	**423**
Limited Actions	**-1.26**	**16.0**	**404**
Fast Demonstration	**-2.37**	**16.0**	**401**

Table 4: This table shows the jumpstart, final reward and total reward metrics for Figure 6, where all HAT methods use Probabilistic Policy Reuse. All demonstrations are 20 episodes, recorded by Subject B. Values in bold have statistically significant differences in comparison to the No Prior method (not shown).

6. FUTURE WORK AND CONCLUSION

This paper has introduced HAT, a novel method to combine learning from demonstration with reinforcement learning by leveraging an existing transfer learning algorithm. Using empirical evaluation in the Keepaway domain we showed that given training data from just a few minutes of human demonstration, HAT can increase the learning rate of the task by several simulation hours. We evaluated three different variants which used different methods to bias learning with the human's demonstration. All three methods performed statistically significantly better than learning without demonstration. Probabilistic Policy Reuse consistently performed at least as well as the other methods, likely because it explicitly balances ex-

ploiting the human's demonstration, exploring, and exploiting the learned policy. Additional evaluation using demonstrations from different teachers, combined demonstrations from multiple teachers, and suboptimal demonstrations all showed that HAT is robust to variations in data quality and quantity. The best learning performance was achieved by combining the best demonstrations from two teachers.

One of the key strengths of this approach is its robustness. It is able to take data of good or poor quality and use it well without negative effects. This is very important when learning from humans because it can naturally handle the noisy, suboptimal data that usually occurs with human demonstration. Its ability to deal with poor teachers opens up opportunities for non-expert users.

In order to better understand HAT and possible variants, the following questions should be explored in future work:

- Is it possible to identify the characteristics that make one set of demonstrations lead to better learning performance than another? Can we identify what influences jumpstart (e.g., Subject B's high jumpstart in Figure 4).
- Rather than performing 1-shot transfer, could HAT be extended so that the learning agent and teacher could iterate between learning autonomously and providing additional demonstrations?
- In this work, the human teacher and the learning agent had different representations of state, and in one case had different action sets. Will HAT still be useful if the teacher and agent are performing different tasks? How similar does the demonstrated task need to be to the autonomous learning task for HAT to be effective?

Acknowledgements

The authors would like to thank Shivaram Kalyanakrishnan for sharing his code to allow a human to control the keepers via keyboard input. We also thank the anonymous reviewers and W. Bradley Knox for useful comments and suggestions.

7. REFERENCES

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469 – 483, 2009.

[2] C. G. Atkeson and S. Schaal. Robot learning from demonstration. In *ICML*, 1997.

[3] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. P. Johnson, and B. Tomlinson. Integrated learning for interactive synthetic characters. *SIGGRAPH*, 2002.

[4] F. Fernández, J. García, and M. Veloso. Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems*, 58(7):866–871, 2010.

[5] F. Fernández and M. Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *AAMAS*, 2006.

[6] D. H. Grollman and O. C. Jenkins. Dogged learning for robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

[7] F. Kaplan, P.-Y. Oudeyer, E. Kubinyi, and A. Miklosi. Robotic clicker training. *Robotics and Autonomous Systems*, 38(3-4):197 – 206, 2002.

[8] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The TAMER framework. In *KCAP*, 2009.

[9] W. B. Knox and P. Stone. Combining manual feedback with subsequent MDP reward signals for reinforcment learning. In *AAMAS*, 2010.

[10] G. Kuhlmann, P. Stone, R. J. Mooney, and J. W. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI Workshop on Supervisory Control of Learning and Adaptive Systems*, 2004.

[11] R. Maclin and J. W. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22(1-3):251–281, 1996.

[12] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Inverted autonomous helicopter flight via reinforcement learning. In *International Symposium on Experimental Robotics*, 2004.

[13] M. Nicolescu, O. Jenkins, A. Olenderski, and E. Fritzinger. Learning behavior fusion from demonstration. *Interaction Studies*, 9(2):319–352, 2008.

[14] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.

[15] B. Price and C. Boutilier. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:569–629, 2003.

[16] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[17] G. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG-RT 116, Engineering Department, Cambridge University, 1994.

[18] M. Saggar, T. D'Silva, N. Kohl, and P. Stone. Autonomous learning of stable quadruped locomotion. In G. Lakemeyer, E. Sklar, D. Sorenti, and T. Takahashi, editors, *RoboCup-2006: Robot Soccer World Cup X*, volume 4434 of *Lecture Notes in Artificial Intelligence*, pages 98–109. Springer Verlag, Berlin, 2007.

[19] O. G. Selfridge, R. S. Sutton, and A. G. Barto. Training and tracking in robotics. In *IJCAI*, 1985.

[20] S. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158, 1996.

[21] B. F. Skinner. *Science and Human Behavior*. Colliler-Macmillian, 1953.

[22] W. D. Smart and L. P. Kaelbling. Effective reinforcement learning for mobile robots. In *ICRA*, 2002.

[23] P. Stone, G. Kuhlmann, M. E. Taylor, and Y. Liu. Keepaway soccer: From machine learning testbed to benchmark. In I. Noda, A. Jacoff, A. Bredenfeld, and Y. Takahashi, editors, *RoboCup-2005: Robot Soccer World Cup IX*, volume 4020, pages 93–105. 2006.

[24] P. Stone, R. S. Sutton, and G. Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.

[25] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.

[26] M. E. Taylor. Assisting transfer-enabled machine learning algorithms: Leveraging human knowledge for curriculum design. In *AAAI Symposium: Agents that Learn from Human Teachers*, 2009.

[27] M. E. Taylor and P. Stone. Cross-domain transfer for reinforcement learning. In *ICML*, 2007.

[28] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.

[29] M. E. Taylor, P. Stone, and Y. Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.

[30] G. Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

[31] L. Torrey, J. W. Shavlik, T. Walker, and R. Maclin. Relational macros for transfer in reinforcement learning. In *ILP*, 2007.

[32] L. Torrey, T. Walker, J. W. Shavlik, and R. Maclin. Using advice to transfer knowledge acquired in one reinforcement learning task to another. In *ECML*, 2005.

[33] E. Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal of Artifical Intelligence Research*, 19(1):205–208, 2003.

[34] A. Wilson, A. Fern, S. Ray, and P. Tadepalli. Multi-task reinforcement learning: a hierarchical Bayesian approach. In *ICML*, 2007.

[35] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[36] P. Zang, A. J. Irani, P. Zhou, C. L. I. Jr., and A. L. Thomaz. Learn via human-provided sequence of tasks using training regimens to teach expanding. In *AAMAS*, 2010.

Auction and Incentive Design

Incentive Design for Adaptive Agents

Yiling Chen
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
yiling@eecs.harvard.edu

Jerry Kung
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
jkung@fas.harvard.edu

David C. Parkes
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
parkes@eecs.harvard.edu

Ariel D. Procaccia
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
arielpro@seas.harvard.edu

Haoqi Zhang
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
hq@eecs.harvard.edu

ABSTRACT

We consider a setting in which a principal seeks to induce an adaptive agent to select a target action by providing incentives on one or more actions. The agent maintains a belief about the value for each action—which may update based on experience—and selects at each time step the action with the maximal sum of value and associated incentive. The principal observes the agent's selection, but has no information about the agent's current beliefs or belief update process. For inducing the target action as soon as possible, or as often as possible over a fixed time period, it is optimal for a principal with a per-period budget to assign the budget to the target action and wait for the agent to want to make that choice. But with an across-period budget, no algorithm can provide good performance on all instances without knowledge of the agent's update process, except in the particular case in which the goal is to induce the agent to select the target action once. We demonstrate ways to overcome this strong negative result with knowledge about the agent's beliefs, by providing a tractable algorithm for solving the offline problem when the principal has perfect knowledge, and an analytical solution for an instance of the problem in which partial knowledge is available.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Economics, Theory

Keywords

Coordination, economically-motivated agents, multiagent systems, principal-agent problem

Cite as: Incentive Design for Adaptive Agents, Yiling Chen, Jerry Kung, David C. Parkes, Ariel D. Procaccia, and Haoqi Zhang, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 627-634.

1. INTRODUCTION

Many situations arise in which a principal wishes to affect the decisions of an agent as he learns to make decisions. For example, a teacher wishes for a student to check answers. A coach wishes for an athlete to adopt particular techniques. A marketer wants a consumer to purchase a particular brand of a product. In these examples, an agent's belief about his valuation for available actions may change with experience through learning or other forms of belief updates. The student may initially check answers but notice that this is time consuming and stop before he becomes good at it. The athlete may adopt and improve a nevertheless imperfect technique and keep with it. The consumer may purchase another brand and develop a loyalty to that brand.

We consider problems in which the *principal* can provide incentives to lead the agent to select a desired action. The teacher can provide gold stars for students who check their answers. The coach can spend effort on teaching a preferred technique. The marketer can advertise or offer discounts on a product. In some cases the provided incentives may not only change the agent's current selection, but also the agent's future selections because he learns that a particular action has high intrinsic value.

We conceptualize this problem as *incentive design for adaptive agents*. An agent's decision problem is assumed to be a multi-armed bandit problem [9, 6]. The agent selects a single action at each time step, and only its belief on the value of that action may change. In addition to modeling learning agents, this models sequential decision problems in which an agent's value for an action adapts over time; e.g., a new toy loses appeal over time or becomes damaged, or a task is completed and an action no longer has value.[1] The principal can provide incentives to influence the agent's behavior, with the goal of inducing a desired action once or multiple times. We insist that the incentives do not affect an agent's (intrinsic) belief on the value of each action, conditional on actions taken.

In our main formulation, the principal has no information about the agent's beliefs on value. But we also consider

[1]We will sometimes use 'learning' to describe the behavior of the agent in the sequel, but intend for such descriptions to also apply to agents with more general adaptive processes.

a variant where the principal is informed. Without knowledge, the problem is to use a limited budget to induce a desired behavior even though incentives can have different consequences when provided at different times. The use of incentives is also somewhat limiting, in that we cannot force the agent to select a particular action.

Our results. We consider two settings, one in which the principal has a fixed budget at each time step and another where the principal has a fixed budget across time steps. In the case of a fixed budget at each time step, we show that the quickest way for a principal to induce a target action once is to assign the budget to this action and wait for the agent to want to select the action. This is optimal for any update process and even with complete knowledge of the update process. Thus, it is optimal for Bayesian learners, as well as heuristic learners, that fit within our general framework. We think this is an interesting finding: the agent's belief update process is left unchanged until the point at which the agent can be incentivized to select the target action. This incentive scheme is also optimal for inducing the goal action to be selected as many times as possible within a fixed number of time periods.

In the case where the principal has a fixed budget across time, the problem is further complicated because the principal needs to decide when to spend the budget. For inducing the target once, assigning the entire budget to the target action remains optimal even with knowledge of the update process. Since no money is spent when the target is not selected, this policy remains feasible for a fixed budget and is therefore optimal for this more constrained problem. But for inducing the goal multiple times, we show that without knowledge of future values, no deterministic or even randomized algorithm provides a bounded competitive ratio for approximating the optimal offline solution, that is, the one obtained when given knowledge of the entire belief sequence. We show that a tractable algorithm exists for finding optimal incentives in the offline problem, and demonstrate on a particular instance of the problem how partial knowledge about the update process and beliefs over values can be used for finding effective incentives.

Related work. In terms of designing incentives to influence an agent's behavior when the agent's preferences are unknown, this work is related to work by Zhang et al. [12, 13, 11] on *environment design* and *policy teaching*. *Environment design* considers the problem of perturbing agent decision problems in order to influence their behavior. *Policy teaching* considers the particular problem of trying to influence the policy of an agent following a Markov Decision Process by assigning rewards to states. In these papers the agent is assumed to have a particular way of making decisions and persistent preferences. This paper can be seen as part of a larger agenda of *online environment design*, where a principal aims to make limited changes to an environment so as to influence the decision of agents while their valuations are still changing, possibly due to learning.

We are not aware of any work on bandits problems that considers a principal who through incentives seeks to induce an adaptive agent to learn to select an action that is desired by the principal.[2] The most closely related work is by Stone and Kraus [10] on *ad hoc* teams. In an *ad hoc* team, there is a learner with values for actions that update based on the empirical mean of observed values, and a teacher who intervenes by taking actions, which lead the agent to make another observation and update its beliefs. The goal is to maximize the combined performance of the teacher and the learner. The main finding is that it is never optimal to teach the worst arm, notably because teaching this is costly and the agent learns that this is the worst arm on its own at no additional loss. On surface level, this seems similar to our positive result on providing incentives on the target action: our agent must learn on its own that the other actions are not as good. However, our setting is quite different in that we cannot directly demonstrate a particular action to the agent but must intervene through incentives. Moreover, the principal's goal need not be aligned with that of the agent, and is ignorant of the agent's values or update process, which can be arbitrary.

Brafman and Tennenholtz [4] consider a teaching setting where a teacher can perform actions within a game to influence the behavior of a learner. However, in this setting there are no incentives and for the most part there is no cost to teaching.

Our problem is also somewhat similiar to the problem of reward shaping within reinforcement learning, where the goal is to adjust an agent's reward feedback in order to improve its performance in a complex environment [8, 7]. However, the assumptions we make are quite different. For example, the agent is not programmable, its values are not observed, and the shaping rewards are costly.

2. THE BASIC MODEL

We consider an agent with a set of actions $K = \{1, \ldots, n\}$. Let $K_{-i} = K \setminus \{i\}$. We use discrete time $t \in \{1, 2, \ldots\}$, and assume that the agent's belief about his value for an action at time t is dependent only on its state $x_i(t)$, which represents the agent's experience with action i prior to time t. Let $v_i(x_i(t))$ denote the agent's belief of the value of action i at time t if selected. At each time step t, the agent selects a particular action i, whose state transitions from $x_i(t)$ to $x_i(t+1)$, independently of time and the states of other actions. This transition can be stochastic, and for example can depend on the sequence of realized rewards from experiences with a particular action. The states of all other actions stay fixed, i.e., $x_j(t+1) = x_j(t), \forall j \neq i$. Throughout the paper, we find it notationally convenient to refer to the state of action i after it has been selected k times as x_i^k, and the agent's belief about its value as $v_i(x_i^k)$.

The agent's current belief can be an arbitrary function of the state, and thus can represent a range of adaptive agent behaviors. This includes, for example, an agent that selects an action according to the empirical average of rewards drawn so far, perhaps coupled with variance weighting to encourage exploration. To illustrate, let $r_1^i, \ldots, r_k^i$ denote the realized rewards received from each of the first k selections of action i. To encode an agent whose belief is the empirical average of rewards, let $v_i(x_i^k) = (\sum_{j=1}^{k} r_j^i)/k$ for $k \geq 1$.

[2]Cavallo et al. [5], Bergemann and Välimäki [2] and Babaioff et al. [1] study a distinct model of incentives in multi-armed bandit problems, from the mechanism design perspective. Each arm is associated with a different agent, and agents have private information about the rewards behind the arms. The goal is to design truthful mechanisms that elicit this information, and enable the center to utilize policies for selecting which arm to pull next to (approximately) maximize social welfare.

To encode the belief of an agent making explore and exploit tradeoffs, we can for example let

$$v_i(x_i^k) = d(x_i^k) + (\sum_{j=1}^{k} r_j^i)/k,$$

where $d(x_i^k)$ is the expected variance in rewards received from selecting action i and is decreasing in k. Similarly, Bayesian learning can also be directly modeled.

We consider a principal who wishes for the agent to select a target action g. The principal can provide incentives $\Delta(t) = (\Delta_1(t), \ldots, \Delta_n(t))$ at each time t, where $\Delta(t)$ can in general depend on any knowledge available to the principal, such as the incentives provided and actions selected prior to time t. The agent observes $\Delta(t)$ prior to selecting his action at time t, and the selected actions are observed by the principal. We assume that incentives are not incorporated into the agent's state, that is, the evolution of an agent's beliefs are independent of the incentives we offer, conditioned on the action the agent selects. We let $\Delta = (\Delta(1), \Delta(2), \ldots, \Delta(t), \ldots)$ denote a sequence of incentive decisions, which are induced by an *incentive policy*. Unless otherwise specified, we assume the principal has no knowledge of the agent's update process, and does not observe the realized rewards from the agent's selections.

In each time period, the agent selects the action with the maximal combined value using the following agent function:[3]

$$f(x(t), \Delta(t)) = \operatorname*{argmax}_{i \in K} [v_i(x_i(t)) + \Delta_i(t)]. \quad (1)$$

The agent is myopic with regard to the intervention of the principal, in that the agent selects the action with the highest combined value without considering the effect of its action on future incentive provisions. Equivalently, the agent adopts a belief that the external incentive is exogeneous and invariant to its own policy, and thus something that does not need to be modeled. While myopic with respect to future incentives, the agent's choice can still reflect explore vs. exploit tradeoffs in its intrinsic value as explained above. However, by assuming that incentives are not incorporated into agent's state, we preclude models of learning in which an agent 'internalizes' the incentives over time.

The online model. Our main analysis is carried out in an online model of computation (see, e.g., [3]); for our purposes an informal description suffices. An instance of our problem specifies a sequence of belief value updates $v_i(x_i^0), v_i(x_i^1), \ldots,$ for each action $i \in K$ and, optionally, a number of periods R. We assume that the principal has no knowledge of these values, and for the most part achieve incentive policies that could not be improved even with full knowledge. Our goal is to design algorithms with the same performance as the optimal offline algorithm with full knowledge of the input. As is usual, we will seek to compete in this sense with the offline algorithm even if the next value of each action is determined after each action of the algorithm in a way that is adversarial and dependent on the history. The performance is measured with respect to one of several objective criteria that we define in the sequel.

[3]For simplicity of exposition, we assume that the agent breaks ties in favor of the target action when there is a tie but otherwise in an arbitrary way. We can replace this assumption, which favors the target action, with any other tie-breaking rule, and all our results would continue to hold.

3. PER-PERIOD BUDGET

We consider first a principal that has a fixed budget at each time step. For example, consider a teacher with a limit of giving two gold stars per period, a coach with a fixed amount of time to demonstrate a preferred technique each period, or a marketer with a cap on the amount of discount that can be provided to a consumer across a set of products. For a per-period budget $B > 0$, we define the budget constraint on Δ as $\Delta_i(t) \leq B$ for all t and $i \in K$, and require further that incentives are non-negative, such that $\Delta_i(t) \geq 0$ for all actions i and times t. Note that the budget constraint formulation assumes that incentives are provided to the agent if and only if the agent selects the action with incentives applied to that action. This captures scenarios where incentives represent contracts (e.g., if you buy this then I give you this incentive), and not to the case where incentives are sunk costs (e.g., advertising dollars). Given this, the principal can in principle assign the entire budget to multiple arms if desired, in hopes that one of them is selected.

To see the power of effective incentives, note that incentives can sometimes induce an action to be selected forever that would otherwise never be selected. Consider a case with two actions, where initially the target action has value 2 and the non-target action has value 3. If either action is chosen, its value updates to 10. Assume $B = 2$. Without intervening in the first period the non-target action will be chosen, its value will update to 10, and it will be chosen forever even with incentives. However, by providing incentives on the target action in the first period it will be induced in that period and forever. The challenge is to design an incentive policy that is successful for all update models and even without knowledge of the update model. We consider two objective criteria.

3.1 Induce once

Consider a principal who wishes to induce action g once as soon as possible by providing effective incentives.

PROBLEM 1 (INDUCE-ONCE). *For a given instance and a budget B, provide incentives to minimize the time t such that $x_g(t) = x_g^1$.*

If a solution does not exist, the minimum is infinity. Note that for action g to be selected at time t it is necessary that $B \geq \max_{i \in K_{-g}} [v_i(x_i(t)) - v_g(x_g(t))]$, at which point it is sufficient to provide $\Delta_g(t) = B$. The INDUCE-ONCE problem is thus identical to finding incentives that most quickly lead the values of all other actions to drop below the *inducible threshold* $T_{\text{once}} = B + v_g(x_g^0)$. For any threshold value T, we define the following:

DEFINITION 1. *A threshold T for inducing action g is met at time t if and only if $v_i(x_i(t)) \leq T$ for all $i \in K_{-g}$.*

At first glance, it may appear that providing incentives to actions other than the target action g can be beneficial, by leading an action with value higher than the threshold to be selected and subsequently significantly drop in value, and in particular, to below the inducible threshold. This intuition turns out to be wrong! Any action above the inducible threshold will in any case be selected by the agent before action g until its value drops below the threshold, even without intervention. Getting such an action to be selected more quickly is possible through incentives, but this does not lead to action g being selected any sooner.

We formalize this observation as the 'threshold lemma,' which we will apply throughout this paper.

LEMMA 1 (THRESHOLD LEMMA). *Given a threshold T, let $k_i = \min\{k : v_i(x_i^k) \leq T\}$, for all $i \in K_{-g}$. Assume such k_i exist. Any incentive policy Δ that assigns $\Delta_i(t) = 0$ for all $i \in K_{-g}$ and $\Delta_g(t) \geq 0$ at every time t has the following properties:*

(a) At any time t before the threshold is first met, $x_i(t) = x_i^{m_i}$ satisfies $m_i \leq k_i$ for all $i \in K_{-g}$.

(b) If the threshold is first met at time t, then $x_i(t) = x_i^{k_i}$ for all $i \in K_{-g}$.

PROOF. Consider part (a). It suffices to show that at any time t before the threshold is first met, any action $i \in K_{-g}$ with $x_i(t) = x_i^{k_i}$ would not be selected at time t. Since the threshold is not yet met at such a time t, there exists $j \in K_{-g}$ such that $j \neq i$ and $v_j(x_j(t)) > T$. Under Δ, action i would not be selected at time t because $v_i(x_i(t)) + \Delta_i(t) = v_i(x_i^{k_i}) \leq T < v_j(x_j(t)) = v_j(x_j(t)) + \Delta_j(t)$, and so action j is strictly preferred.

Now consider part (b). If the threshold is first met at time t then exactly one action, say $\ell \in K_{-g}$, had been selected $k_\ell - 1$ times by period $t-1$ and was selected in period $t-1$ and every other action $j \in K_{-g}, j \neq \ell$ had already been selected at least k_j times by period $t-1$. By (a), these other actions had been selected exactly k_j times by period $t-1$ and hence $x_i(t) = x_i^{k_i}$ for all $i \in K_{-g}$ in period t. □

The threshold lemma shows that only providing incentives to the target action ensures that no other action is selected more times than needed before the threshold is met. Note that it does not guarantee the threshold will be met; that still needs to be shown for a particular incentive policy and corresponding threshold.

We next introduce a simple incentive policy that is central in our analysis. Its acronym hints at its guarantees.

DEFINITION 2. *The 'only provide to target' (OPT) incentive policy assigns $\Delta_g(t) = B$ and $\Delta_i(t) = 0$ for all $i \in K_{-g}$ for every time t.*

Note that in defining OPT we did not make any assumptions regarding its knowledge of current values or future updates.

THEOREM 1. *In the online model and under a per-period budget, OPT always provides the optimal offline solution to* INDUCE-ONCE.

PROOF. Consider $T_{\text{once}} = v_g(x_g^0) + B$ and define $k_i = \min\{k : v_i(x_i^k) \leq T_{\text{once}}\}$ for all $i \in K_{-g}$, and consider the interesting case in which this exists for every action so that a solution is not trivially precluded. The best possible solution will induce the agent to select the goal action after the necessary k_i activations of each action $i \in K_{-g}$. But actions $i \in K_{-g}$ can be selected no more than k_i times before the threshold is met by part (a) of the threshold lemma, and thus the threshold must be met under OPT. By applying part (b) of the threshold lemma, OPT makes the fewest selections of actions in K_{-g} necessary to meet the threshold, plus an additional necessary step to induce the target action. □

The key observation is that nothing the principal can do will speed up the agent's exploration of currently better actions. The principal can do worse than OPT however, e.g., by placing incentives on an action other than the target whose value is below the threshold and whose value in the state transitioned to is much higher.

3.2 Induce multiple times

In the motivating examples we consider, the principal may want the agent to select the target action (e.g. check answers, use a particular technique, or buy a product) more than once. This leads to the next objective criterion.

PROBLEM 2 (INDUCE-MULTI). *For a given instance, a budget B, and a number of rounds R, provide incentives to maximize m such that $x_g(R) = x_g^m$.*

Let us first tackle the related problem of minimizing the time to get m selections, for a given m. We know from Theorem 1 that OPT is the optimal incentive policy for $m = 1$. Furthermore, for $m \geq 2$, we know that OPT gets each subsequent selection of action g most quickly from *any* state configuration. However, this is not enough to conclude that OPT is the optimal incentive policy for getting m selections, because there may be other incentive policies that are slower than OPT at getting the first selection but faster in getting subsequent selections. While such incentive policies exist, we use the threshold lemma to show that they can do no better than OPT in minimizing the total amount of time needed to get m selections:

LEMMA 2. *In the online model and under a per-period budget, and for any fixed $m > 1$, OPT minimizes the time t such that $x_g(t) = x_g^m$.*

PROOF. Let $w = \text{argmin}_{0 \leq \ell < m} v_g(x_g^i)$ and let $T_{\text{multi}} = v_g(x_g^w) + B$. Let $k_i = \min\{k : v_i(x_i^k) \leq T_{\text{multi}}\}$ for all $i \in K_{-g}$, and consider the case in which this exists for every action so that a solution is not trivially precluded. The best possible solution will induce the agent to select the goal action the m-th time after the necessary k_i activations to each action $i \in K_{-g}$. But actions $i \in K_{-g}$ can be selected no more than k_i times before the threshold is met by part (a) of the threshold lemma, and thus the threshold must be met under OPT. Consider the period in which the threshold is first met. By applying part (b) of the threshold lemma, OPT makes the fewest selections of actions in K_{-g} necessary to meet the threshold, and since only the target item is selected thereafter until m selections are made, this completes the proof. □

By defining the threshold as the minimum value attained by action g before m selections we can apply the same idea as in the proof of Theorem 1. A fixed number of selections must necessarily occur on the other actions, and once they occur under OPT these actions will no longer be selected again.

THEOREM 2. *In the online model and under a per-period budget, OPT always provides the optimal offline solution to* INDUCE-MULTI.

PROOF. Let m denote the number of selections of the target action in time R under OPT. Assume for contradiction that there exist an incentive policy to induce the target $m' > m$ times in R steps. But by Lemma 2, OPT must also be able to induce the target action m' times in the same number or fewer time steps. This is a contradiction. □

4. FIXED ACROSS-PERIOD BUDGET

In this section, we consider a setting in which the principal has a budget that is fixed over time, and must decide on how to allocate that budget across time in order to induce the target action g once or multiple times. Formally we define the budget constraint on Δ as $\sum_{t=1}^{\infty} \Delta_{i(t)}(t) \leq B$, where $i(t)$ denotes the agent's selection at time t. We still require that incentives are non-negative, i.e., $\Delta_i(t) \geq 0$ for all actions i and times t.

This problem seems more difficult than the per-period budget problem because the principal must now decide how to split its budget across rounds. Providing too little in a particular round can miss an opportunity given the current state, whereas providing too much may make it difficult to induce future selections of the target action. As we will show, this turns out to be a nonissue if we wish to induce the target action once, but prevents any online algorithm from providing performance guarantees if we wish to induce the target action multiple times.

4.1 Induce once

We first return to INDUCE-ONCE, that is, we have a principal who wishes to induce action g once and as soon as possible. However, now the incentive policies under consideration have a fixed budget B across time.

Consider using OPT for this problem. OPT is optimal for the per-period budget case when B is available each period. Moreover, OPT in fact spends no money when the target is not selected, and so remains feasible even for a fixed budget across rounds and therefore optimal for this more constrained problem. The proof of this theorem is omitted as it is essentially identical to the proof of Theorem 1.

THEOREM 3. *In the online model and under a fixed across-period budget, OPT always provides the optimal offline solution to* INDUCE-ONCE.

4.2 Induce multiple times

Now consider the INDUCE-MULTI problem with a principal who wishes to induce the target action as many times as possible in a fixed number of rounds R. OPT is no longer optimal here because it may be beneficial to split the budget with the aim of getting more selections of the target action.

Consider a setting with two actions and a total budget of $B = 1$. Action 1 is the goal action. It may be that $v_2(x_2^0) = v_g(x_g^0) + B$ and v_2 increases in future states. By providing B on action g in the first period the goal is induced once, compared to zero successes with any other policy. On the other hand, suppose instead that $v_2(x_2^0) = v_g(x_g^0) + \epsilon$, some $0 < \epsilon < B$, and the value of both actions remains constant in all states. By providing B on g in the first period only one activation is achieved, whereas $\min(R, 1/\epsilon)$ could be achieved by providing ϵ on action g while budget remains (and this can be made arbitrarily large by increasing R and decreasing ϵ.)

In the online algorithms literature, an online algorithm for a maximization problem is α-competitive if the ratio between the optimal *offline* solution and the algorithm's solution is at most α for any given instance. Theorems 1 and 2 can be reformulated to state that under a per-period budget OPT is 1-competitive for INDUCE-ONCE and INDUCE-MULTI, respectively. On the other hand, the above argument implies that under an across-period budget there is no deterministic online algorithm that provides a bounded competitive ratio for the INDUCE-MULTI problem.

Our next formal result strengthens the above observation; we show that even a *randomized* algorithm cannot achieve a bounded approximation ratio. When the algorithm is randomized, the 'game' is as follows: we choose a randomized algorithm, then the adversary chooses an input; the input chosen by the adversary does not depend on the realization of the algorithm's randomness. The theorem holds even if the algorithm is allowed to know the current values of the actions at each time! In other words, this impossibility holds even for algorithms that are significantly more powerful than those we considered earlier.

THEOREM 4. *Under a fixed across-period budget there is no randomized algorithm that provides a bounded competitive ratio for* INDUCE-MULTI*, even if the algorithm can see the current values of the actions.*

The proof appears in the appendix. This result implies that it will be important to consider empirical performance or average case analysis, for particular agent models, in order to make progress.

4.3 Offline problem

As a counterpoint to Theorem 4, we consider the offline case, in which the principal knows the agent's value for any state of the actions the agent may reach and that state transitions are deterministic. This corresponds to a situation in which the agent is of known design, and that the principal has full understanding of the dynamics within the agent's decision environment.

The question of interest is whether there exists a tractable solution to this problem. An effective incentive policy would need to figure out when to provide incentives and how to split the budget across time periods, and a brute force computation of the optimal incentive to provide at each time step is too expensive.

THEOREM 5. *In the offline model and under a fixed across-period budget, an optimal solution to* INDUCE-MULTI *can be found in polynomial time.*

The proof involves the analysis of a nontrivial incentive policy; we give the outline here and relegate the proof of the key lemma to the appendix. To break down the problem, we first consider finding fixed budget incentive policies to solve the following subproblem.

PROBLEM 3. *Given $\bar{t} > 1$ and $m > 1$, and a budget B, find an incentive policy such that $x_g(\bar{t}) = x_g^{\ell}$ for $\ell \geq m$ when a solution exists.*

Essentially, if we can find an incentive policy that can get at least m selections in $\bar{t}$ rounds whenever possible for any $m > 1$, we can fix $\bar{t} = R$ and do a binary search over m to solve INDUCE-MULTI.

To get m selections within $\bar{t}$ time steps it is necessary that the agent selects the non-target actions no more than $\bar{t} - m$ times. An effective incentive policy should provide incentives on the target action when the other actions are least desirable, regardless of the value of the target action. This is the state in which it is cheapest to activate the target action.

We define the relevant activation threshold by simulating the agent function on actions K_{-g} only for $\bar{t} - m$ periods with no incentives, and computing

$$\underline{v} = \min_{1 \le t \le \bar{t}+1-m} \max_{i \in K_{-g}} v_i(x_i(t)). \quad (2)$$

It is easy to see that this threshold, $\underline{v}$, can be computed in polynomial time.

DEFINITION 3. *The 'only provide to target when cheap' (OPTc) incentive policy assigns $\Delta_i(t) = 0$ for all $i \in K$ until the threshold $T = \underline{v}$ is met, where $\underline{v}$ is defined as in Equation (2). Let t' denote the period in which the threshold is first met. OPTc provides $\Delta_g(t) = \max\{0, \underline{v} - v_g(x_g(t))\}$ for $t \ge t'$ while there is enough budget remaining and $\Delta_g(t) = 0$ otherwise. No incentives are ever provided to actions in K_{-g}.*

Now, by using binary search, the following lemma is sufficient to prove Theorem 5.

LEMMA 3. *In the offline model and under a fixed across-period budget, OPTc solves Problem 3.*

The proof of the lemma appears in the appendix. An interesting aspect of OPTc is that it has much of the same structure as OPT: the optimal incentive policy does not modify the agent's learning process until the point where the agent can best be incentivized to select the target action. The offline problem remains a problem about when to provide incentives on the target action and not how to intervene in the selection process on other actions.

4.4 Case study: induce a new action

Theorem 4 and Theorem 5 establish strong negative and positive results at opposing ends of the information spectrum. In most realistic scenarios we expect the principal to have some (but not full) knowledge of the agent's update process and reward distribution. Sticking with an across-period budget and the objective of maximizing the number of selections of the target arms within a fixed number of rounds, we demonstrate how to utilize such knowledge to find effective incentives for a particular problem of interest.

Consider a scenario in which there are two actions, one whose value to the agent is fixed and another whose realized rewards are drawn from a stationary distribution known to the principal. Moreoever, assume the agent's belief updates based on the empirical average of rewards, and is initialized by a draw from the same distribution (e.g., the agent gets an initial sample). This scenario models an agent's choice between an incumbent option (e.g., a known product, service, or technique) and a new entrant option, where the principal wishes to entice the agent toward the new option by providing appropriate incentives.

To be more concrete, let us consider a particular instance of the problem. There are 2 rounds. The value of the incumbent option is fixed at 1, and the reward from selecting the new option and his initial belief on its value are sampled uniformly from 0 to 1. The process reflects uncertainty about the new option's quality, and the agent's updating beliefs reflect his estimate of the new option's value. The principal has a budget of 1 to be spread across the rounds, and can observe which action the agent selects but not the realized rewards. The goal is to maximize the expected number of selections of the new action.

In analyzing this problem, note that the agent's decision in the first period provides some information on the agent's value at the time, and thus, the distribution of possible values following the update. Furthermore, if the agent does not select the new action in the first period, then no money is spent and we can guarantee a selection in the second period. Solving for the optimal incentive policy analytically, we get the following fact:

FACT 1. *The optimal incentive policy for this problem provides $\frac{4}{9}$ to the new action in the first period and the remaining budget in the second period. The expected number of selections is $\frac{25}{18}$.*

PROOF (SKETCH). Let T_i represent an indicator variable over whether the agent selects the new action in period i, such that $P(T_i)$ represents the probability of the selection given the principal's uncertainty over the agent's current value for the action and the provided incentive. Let α represent the incentive provided to the new action in the first period. We wish to maximize the expected number of selections:

$$\begin{aligned} & 2P(T_1)P(T_2|T_1) + P(T_1)P(\neg T_2|T_1) + P(\neg T_1)P(T_2|\neg T_1) \\ = \; & P(T_1)[P(T_2|T_1) + 1] + P(\neg T_1) \\ = \; & P(T_1, T_2) + 1 \end{aligned}$$

Here $P(T_1, T_2) = P(1 - \alpha \le r_0 \le 1, \frac{r_0 + r_1}{2} \ge \alpha)$, where r_0 represents the initial value on the new action and r_1 represents the value from the first selection of the new action. By integrating the probabillity density function of the uniform distribution over the valid regions based on the value of α, we have:

$$P(T_1, T_2) = \begin{cases} \alpha & \text{if } \alpha \le \frac{1}{3}. \\ -\frac{9}{2}\alpha^2 + 4\alpha - \frac{1}{2} & \text{if } \frac{1}{3} < \alpha \le \frac{1}{2}. \\ -\frac{5}{2}\alpha^2 + 2\alpha & \text{if } \frac{1}{2} < \alpha \le \frac{2}{3}. \\ 2\alpha^2 - 4\alpha + 2 & \text{if } \frac{2}{3} < \alpha \le 1. \end{cases}$$

It is easy to check from here that the maximum is attained in the second segment, with $\alpha = 4/9$. □

Intuitively, providing too much incentives in the first period misses out on possible selections in the second period, and providing too little in the first period may likewise miss out on selections in the first period. Given this, 4/9 seems like a good choice for α. However, it is surprising that the optimal solution is not $\alpha = 1/2$: it turns out that by providing slightly less incentives in the first period, there is a higher chance (7/8 vs. 3/4) of getting selections in the second period conditional on a selection in the first period, because the value of the first draw must have been higher and more incentives are left over. Even though the principal is slightly less likely to get a selection in the first period, this helps to maximize the probability of getting a selection in both rounds, which we have shown is equivalent to maximizing the expected number of selections.

We can also consider the same problem but with 3 rounds, where the optimal incentive policy (obtained via simulation) provides about 0.37 in the first period, 0.27 in the second period if the new action is selected and 0.54 otherwise. It is interesting to note the disparity in the amount of incentives provided in the second period based on what happened in

	INDUCE-ONCE	INDUCE-MULTI
Per-period	OPT optimal (Thm 1)	OPT optimal (Thm 2)
Fixed	OPT optimal (Thm 3)	Unbounded ratio (Thm 4)

Table 1: Summary of our results in the online model for the two objective criteria and different budget models for the principal.

the first period. Since success in the first period indicates a high draw in the first period and failure indicates a low draw, the observation serves as an informative signal about the amount of incentives required in the second period.

5. DISCUSSION

Table 1 summarizes our main results; rows correspond to the assumption made on the budget and columns to the optimization problem. The most striking aspect is the relation between the performance of OPT when one varies the assumption on the budget between per-period and fixed, and the problem between INDUCE-ONCE and INDUCE-MULTI. As long as one of these dimensions remains fixed OPT is still optimal, but when we consider the harder variation in both dimensions then even a randomized policy that knows the current values cannot provide a bounded ratio!

Our incentive policy for the per-period budget case requires no knowledge of the agent's values or value update process, nor the number of repetitions for which we wish to induce the target, nor the time horizon over which the target is to be induced; it is optimal even with this knowledge. In this setting, it is not necessary for the principal to learn about the agent, for example by drawing inferences about the agent's values and update process from observed behavior of selected options. It is interesting, also, that the principal is unable to usefully perturb the agent's learning process until the point at which his desired goal action can be induced, and even if he knew the agent's values or value update process.

Interestingly, this is quite different in the across-period budget setting, where progress will require knowledge of an agent's selection and learning process or learning by the principal about the agent. The analytical approach demonstrated for finding incentives to induce an agent to select a new action uses both knowledge about the value update process and inference on the distribution of current values based on past decisions, and can be applied for rewards drawn from different distributions and for different update processes. Future work should seek to obtain tractable algorithms for finding effective incentives given a known model of agent behavior but private agent beliefs, and seek to gain a better understanding of the structure of effective incentive policies, on particular classes of problems.

In addition to variations on the budget constraint, one can consider variations to the agent's selection policy. For example, the agent might select the action with highest value with probability $1-\epsilon$, and select a random action with probability ϵ. It is possible to show that in this case OPT is no longer guaranteed to be optimal for a per-period budget, even with respect to INDUCE-ONCE.[4] It is also of interest to relax the assumption on the independence of actions, and to consider models with long-term learning in which the agent learns to internalize external incentives and change its own intrinsic value for future actions. Other objective criteria are also of interest, for example a principal that wants to induce an action followed by another action, in immediate succession.

[4]Consider a case with three actions. The principal has a per-period budget of 2. Action 1 is the target and has its value fixed at 1. Action 2 is associated with the belief sequence $(v_2^0, v_2^1, \ldots) = (5, 0, 5, 5, \ldots)$, and action 3 is associated with the belief sequence $(v_3^0, v_3^1, \ldots) = (4, 0, 0, \ldots)$. Here action 1 can be induced by a non-random action if and only if action 2 is induced exactly one, and action 3 is induced at least once. For small ϵ, action 2 is most likely to be selected first under OPT. This is undesirable, however, since any random selection of action 2 henceforth will result in no future selections of action 1. It is better to instead provide incentives to action 3 in the first period (and apply OPT thereafter), so that we try to 'hold off' on selecting action 2 until the belief on the value of action 3 has dropped.

Acknowledgments

The last author acknowledges support from the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. In addition, this work is supported in part by NSF grant CCF 0915016, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

6. REFERENCES

[1] M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. In *ACM-EC*, pages 79–88, 2009.

[2] D. Bergemann and J. Välimäki. The dynamic pivot mechanism. *Econometrica*, 78:771–789, 2010.

[3] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[4] R. I. Brafman and M. Tennenholtz. On partially controlled multi-agent systems. *JAIR*, 4:477–507, 1996.

[5] R. Cavallo, D. C. Parkes, and S. Singh. Optimal coordinated planning amongst self-interested agents with private state. In *UAI*, pages 55–62, 2006.

[6] R. D. Kleinberg. *Online decision problems with large strategy sets*. PhD thesis, MIT, 2005.

[7] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *K-CAP*, pages 9–16, 2009.

[8] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, pages 278–287, 1999.

[9] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[10] P. Stone and S. Kraus. To teach or not to teach? decision making under uncertainty in ad hoc teams. In *AAMAS*, 2010.

[11] H. Zhang, Y. Chen, and D. C. Parkes. A general approach to environment design with one agent. In *IJCAI*, pages 2002–2008, 2009.

[12] H. Zhang and D. C. Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI*, pages 208–214, 2008.

[13] H. Zhang, D. C. Parkes, and Y. Chen. Policy teaching through reward function learning. In *ACM-EC*, pages 295–304, 2009.

APPENDIX

A. PROOFS

A.1 Proof of Theorem 4

We assume without loss of generality that the budget size is 1. Let $k \in \mathbb{N}$; assume for contradiction that there is a randomized online algorithm with a competitive ratio α (worst-case ratio of number of activations of g in offline optimal to expected number of activations of g in online algorithm over all instances) smaller than k. Set $\epsilon = 1/(10k)$.

We consider a setting where there are just two actions, the target action g and the non-target h. We design an infinite family of inputs $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_j, \ldots$ Note that an input simply specifies a number of rounds, and a sequence of values for g and h. For all $j \in \mathbb{N} \cup \{0\}$, the sequence of values that $\mathcal{I}_j$ assigns to g is all zeros, that is, $v_g(x_g^t) = 0$ for all t; the inputs only differ on their values h. The sequence of values $v_h(x_h^0), v_h(x_h^1), \ldots$ that $\mathcal{I}_j$ assigns to this action is

$$1, \epsilon, \epsilon^2, \ldots, \epsilon^j, 2, 2, \ldots, 2, \ldots$$

We do not specify the number of rounds, as we can choose it to be large enough for it not to be an issue.

Given a run of the algorithm on some input $\mathcal{I}_j$, we refer to the sequence of selections of action g while action h has a value ϵ^p as *phase* p. Once h is selected we move to phase $p+1$. Note that for each select of g in phase p the algorithm has to invest ϵ^p of its budget.

Let Z_p^j be a random variable that denotes the budget spent by the algorithm within phase p given the input $\mathcal{I}_j$, where the randomness comes from the algorithm's coin flips. The crux of the proof is the following lemma.

LEMMA 4. *For every $j \in \mathbb{N} \cup \{0\}$ and every $p \in \{0, \ldots, j\}$, if the randomized online algorithm has competitive ratio smaller than k then $\mathbb{E}[Z_p^j] \geq \epsilon$.*

PROOF. Assume for contradiction that this is not the case, i.e., there is some $j \in \mathbb{N} \cup \{0\}$ and $p \in \{0, \ldots, j\}$ such that $\mathbb{E}[Z_p^j] < \epsilon$. Up to phase p the algorithm cannot distinguish between $\mathcal{I}_j$ and $\mathcal{I}_p$ (due to the online nature of the model), hence it holds that $\mathbb{E}[Z_p^p] < \epsilon$, that is, the algorithm spends less than ϵ in expectation in phase p given the input $\mathcal{I}_p$. It follows that the expected number of times g is selected in phase p is smaller than $\epsilon/\epsilon^p = 10^{p-1}k^{p-1}$. Given $\mathcal{I}_p$, the algorithm will no longer be able to select g after phase p (since then the value of h is then 2). We derive an upper bound on the expected number of times the algorithm selects g on $\mathcal{I}_p$ by generously allowing the algorithm spend a budget of 1 in every phase $p' < p$. The upper bound is then

$$1 + 10k + \cdots + 10^{p-1}k^{p-1} + 10^{p-1}k^{p-1} \leq 3 \cdot 10^{p-1}k^{p-1}.$$

On the other hand, the optimal offline solution on $\mathcal{I}_p$ selects g $10^p k^p$ times, i.e., the ratio α is at least $(10/3)k$, in contradiction to the assumption that the algorithm's competitive ratio is smaller than k. □

Now, consider input $\mathcal{I}_{j^*}$ for some $j^* \in \mathbb{N}, j^* > 1/\epsilon$. By Lemma 4 we have that $\mathbb{E}[Z_p^{j^*}] \geq \epsilon$ for all $p \in \{0, \ldots, j^*\}$. It follows from the linearity of expectation that

$$\mathbb{E}\left[\sum_{p=0}^{j^*} Z_p^{j^*}\right] = \sum_{p=0}^{j^*} \mathbb{E}[Z_p^{j^*}] > \frac{1}{\epsilon} \cdot \epsilon = 1. \tag{3}$$

However, since the total budget size is 1 the random variable $\sum_{p=0}^{j^*} Z_p^{j^*}$ must take values in $[0, 1]$, and in particular $\mathbb{E}[\sum_{p=0}^{j^*} Z_p^{j^*}] \leq 1$. This is a contradiction to Equation (3). □

A.2 Proof of Lemma 3

We first establish that $\underline{v}$, as defined in Equation (2), is the threshold where it is cheapest to provide incentives to the target over at most $\bar{t} - m$ periods of selecting actions other than g. For this, let

$$v^* = \min_{m_{-g}} \max_{i \in K_{-g}} v_i(x_i^{m_i}) \tag{4}$$
$$\text{s.t.} \sum_{i \in K_{-g}} m_i \leq \bar{t} - m$$

represent the lowest value of the highest action with up to $\bar{t} - m$ selections of the non-target actions, and where $m_{-g} = (m_1, \ldots, m_{g-1}, m_{g+1}, \ldots, m_n)$.

We want to establish that $\underline{v} = v^*$. Indeed, clearly $v^* \leq \underline{v}$ by definition. Suppose for contradiction that $\underline{v} > v^*$. Let m^*_{-g} minimize the expression in Eq. (4). Consider running the simulation used to define $\underline{v}$ for $\sum_{i \in K_{-g}} m_i^*$ rounds, and let ℓ_i denote the number of times that each action $i \in K_{-g}$ is selected in this process.

If $\ell_i = m_i^*$ for all $i \in K_{-g}$ then clearly $\underline{v} = v^*$, since it is attained in the final period of the simulation, and this is a contradiction. Otherwise, and using the fact that $\sum_{i \in K_{-g}} \ell_i = \sum_{i \in K_{-g}} m_i^*$, then there exists some $j \in K_{-g}$ such that $m_j^* < \ell_j$. Note that,

$$v_j(x_j^{m_j^*}) \leq v^* < \underline{v}, \tag{5}$$

where the first inequality holds by definition of v^* and the second by assumption. Now there is some time t during the simulation where $x_j(t) = x_j^{m_j^*}$, and action j is selected. But by definition of $\underline{v}$ the value of the action that is selected by the agent must be at least $\underline{v}$, in contradiction to (5). This establishes $\underline{v} = v^*$.

In order to complete the proof of the lemma, we now know that OPTc uses $\underline{v} = v^*$ as the threshold T. Let $k_i = \min\{k : v_i(x_i^k) \leq T\}$ for all $i \in K_{-g}$. By definition of v^*, $\sum_{i \in K_{-g}} k_i \leq \bar{t} - m$. Note that OPTc satisfies the conditions of the threshold lemma. Proceed by case analysis. If the threshold is not met after $\bar{t}$ rounds then, by part (a) of the threshold lemma, action g must have been selected at least m times and the case is established. Otherwise, if the threshold is met, it is met after at most $\bar{t} - m$ selections of actions in K_{-g} by part (b) of the threshold lemma. For any incentive policy to get m selections in $\bar{t}$ rounds, it must have provided at least $\max\{0, v^* - v_g(x_g^\ell)\}$ to get selection number $\ell + 1$ of action g, for each of $\ell \in \{0, 1, \ldots, m-1\}$. Since OPTc spends no budget before the threshold is met and once it is met it provides exactly $\max\{0, v^* - v_g(x_g^\ell)\}$ for selection number $\ell+1$ of action g, for each of $\ell \in \{0, 1, \ldots, k-1\}$, then OPTc will get at least m selections of action g whenever this is possible under any incentive policy. This completes the case, and the proof. □

A Truth Serum for Sharing Rewards

Arthur Carvalho
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
a3carval@cs.uwaterloo.ca

Kate Larson
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
klarson@cs.uwaterloo.ca

ABSTRACT

We study a problem where a group of agents has to decide how a joint reward should be shared among them. We focus on settings where the share that each agent receives depends on the *subjective opinions* of its peers concerning that agent's contribution to the group. To this end, we introduce a mechanism to elicit and aggregate subjective opinions as well as for determining agents' shares. The intuition behind the proposed mechanism is that each agent who believes that the others are telling the truth has its expected share maximized to the extent that it is well-evaluated by its peers and that it is truthfully reporting its opinions. Under the assumptions that agents are Bayesian decision-makers and that the underlying population is sufficiently large, we show that our mechanism is incentive-compatible, budget-balanced, and tractable. We also present strategies to make this mechanism individually rational and fair.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*;
J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Economics, Theory

Keywords

Fair division, Bayesian Truth Serum, Mechanism Design

1. INTRODUCTION

Understanding how agents can work together in order to achieve some common goal is a central research topic in the field of multiagent systems [15]. Questions that are typically analyzed include how and which groups of agents should form [13], how agents should coordinate their actions once they have agreed to work together [4], how to ensure that the group, once formed, does not disintegrate [2], and how any joint rewards should be divided among the group members [9]. It is this last question that we address in this paper.

Cite as: A Truth Serum for Sharing Rewards, Arthur Carvalho and Kate Larson, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 635-642.

Commonly called *fair division*, the problem of dividing one or several goods among a set of agents, in a way that satisfies a suitable fairness criterion, has been studied in several literatures. In economics, the collective welfare approach is arguably the most influential application of the economic analysis to fair division. It uses the concepts of collective utility functions, in its cardinal interpretation, and social welfare orderings, in its ordinal interpretation, for deciding what makes a reasonable division [9]. In computer science and, more specifically, artificial intelligence, the fair division problem is traditionally studied in settings where the underlying agents not only have preferences over alternative allocations of goods, but also actively participate in computing an allocation [1].

In this work, we propose a novel game-theoretic model for sharing a joint, homogeneous reward based on the idea of *subjective opinions*. In detail, we consider scenarios where a group has been formed and has accomplished a task for which it is granted a reward, which must be shared among the group members. After observing the individual contributions of the peers in accomplishing the task, each agent is asked to *evaluate* the others. Agents also provide *predictions* about how their peers are evaluated. Thus, we consider two kinds of subjective opinions when sharing the joint reward: *evaluations* and *predictions*. These opinions are elicited and aggregated by a central, trusted entity called the *mechanism*, which is also responsible for sharing the reward based exclusively on the received opinions.

The share received by each agent from the proposed mechanism has two major components. The first one reflects the evaluations received by that agent. The second one is a truth-telling score used to encourage agents to truthfully report their opinions. For computing such scores, the mechanism uses the Bayesian truth serum method [12]. The intuition behind the proposed mechanism is that each agent who believes that the others are telling the truth has its expected share maximized to the extent that it is well-evaluated and that it is also telling the truth. Under the assumptions that agents are Bayesian decision-makers and that the underlying population is sufficiently large, we show that our mechanism is incentive-compatible, budget-balanced, and tractable. We also present strategies to make this mechanism individually rational and fair.

Besides this introductory section, the rest of this paper is organized as follows. In Section 2, we describe the model, concepts used throughout the paper, and properties that we wish our mechanism to exhibit. In Section 3, we introduce our mechanism and prove that it satisfies interesting prop-

erties. In Section 4, we empirically investigate the influence of the model and mechanism's parameters on agents' shares. In Section 5, we review the literature related to our work. Finally, we conclude in Section 6.

2. MODEL AND BACKGROUND

A set of agents $N = \{1, \ldots, n\}$, for $n \geq 3$, has accomplished a task for which it is granted a *reward* $V \in \Re^+$. Every agent is assumed to want more of the reward. Therefore, we can identify an agent's share with its welfare. We are interested in settings where the share of V that an agent receives depends on the *subjective opinions* of its peers concerning that agent's contribution to the group.

We model the private information of an agent as $n-1$ private signals that the agent receives from its peers. These signals are direct assessments of the peers' performance in accomplishing the joint task, and we call them *truthful evaluations*. Formally, given a positive integer parameter M, for $1 \leq M \leq V$, the signals observed by agent i are represented by the vector $\mathbf{t}_i = (t_i^1, \ldots, t_i^{i-1}, t_i^{i+1}, \ldots, t_i^n)$, where $t_i^j \in \{1, \ldots, M\}$ represents the signal observed by agent i coming from agent j. Thus, $\mathbf{t}_i$ is the vector with the truthful evaluations made by agent i regarding the contributions of its peers in accomplishing the task. In this way, the parameter M represents the top possible evaluation that an agent can give or receive, and we assume that its value is common knowledge. For each agent $j \in N$, let $\omega_j \in \Delta^M$ (unit simplex in $\Re^M$) be an unknown parameter representing the distribution of the truthful evaluations for agent j.

Based on their truthful evaluations, agents can make predictions about how their peers are evaluated. The predictions made by agent i are formally represented by the vector $\mathbf{r}_i = (r_i^1, \ldots, r_i^{i-1}, r_i^{i+1}, \ldots, r_i^n)$, where agent i's prediction about the empirical distribution of evaluations received by agent j is $r_i^j = (r_i^{j^1}, \ldots, r_i^{j^M}) \in \Delta^M$, *i.e.*, $0 \leq r_i^{j^k} \leq 1$ and $\sum_{k=1}^{M} r_i^{j^k} = 1$. Mathematically, r_i^j is the expected distribution of truthful evaluations for agent j given agent i's truthful evaluation, *i.e.*, $r_i^j = \mathbb{E}[\omega_j | t_i^j]$.

To avoid a biased self-judgment, agents are neither asked to make self-evaluations nor asked to make predictions about their received evaluations. They are requested to report their subjective opinions, namely, evaluations and predictions. We make the following assumptions in our model:

1. *Self-interestedness.* Agents act to maximize their expected shares.
2. *Common prior.* $\forall j \in N$, there exists a common prior distribution, $p(\omega_j)$, over ω_j.
3. *Rationality.* Every agent i, with truthful evaluation t_i^j, forms a posterior by applying Bayes' rule to the common prior $p(\omega_j)$, *i.e.*, $p(\omega_j | t_i^j)$.
4. *Stochastic relevance.* $\forall i, q, j \in N$, $p(\omega_j | t_i^j) = p(\omega_j | t_q^j)$ if and only if $t_i^j = t_q^j$.
5. *Large population.* The population of agents must be sufficiently large so that a single evaluation for an agent cannot significantly affect the empirical distribution of evaluations received by that agent.
6. *Independent signals.* The signals observed by an agent are independent of each other. Formally, given $i, j, k \in N$, and $x, y \in \{1, \ldots, M\}, p(t_i^j = x \,|\, t_i^k = y) = p(t_i^j = x)$.

The first assumption means that agents are *risk neutral* [7]. The second assumption means that agents have common prior distributions over the distributions of the truthful evaluations for their peers. The third assumption means that these priors are consistent with Bayesian updating. These first three assumptions are traditional in both game theory [11] and multiagent systems [15] literature, and they essentially mean that agents are Bayesian decision-makers. The fourth assumption means that different truthful evaluations imply different posterior distributions, and vice-versa. By far, the most stringent assumption is the requirement of a large population. Later in this paper, we discuss the implications of such assumption and how to circumvent it. Finally, the last assumption implies that the truthful evaluation of an agent for a peer does not influence that agent's truthful evaluation for other peer.

A consequence of self-interest is that agents may deliberately lie when reporting their evaluations and/or predictions. For example, an agent may intentionally give all other agents a low evaluation so that, in comparison, it looks good and receives a greater share of V. Therefore, we distinguish between the truthful evaluations made by each agent $i \in N$, $\mathbf{t}_i$, and the evaluations that agent i reports, $\mathbf{x}_i = (x_i^1, \ldots, x_i^{i-1}, x_i^{i+1}, \ldots, x_i^n)$. Similarly, we distinguish between the truthful predictions made by each agent $i \in N$, $\mathbf{r}_i$, and the predictions that agent i reports, $\mathbf{y}_i = (y_i^1, \ldots, y_i^{i-1}, y_i^{i+1}, \ldots, y_i^n)$.

We define the *strategy* of agent i, $\mathbf{s}_i = (\mathbf{x}_i, \mathbf{y}_i)$, to be its reported opinions. S_i is the set of strategies available to agent i, and $S = S_1 \times \ldots \times S_n$. We note that the parameter M fully determines the strategies available to the agents. Each vector $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_n) \in S$ is a *strategy profile*. As customary, let the subscript "$-i$" denote a vector without agent i's component, *e.g.*, $\mathbf{s}_{-i} = (\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}, \mathbf{s}_{i+1}, \ldots, \mathbf{s}_n)$. If the opinions reported by agent i are equal to its truthful opinions, *i.e.*, $\mathbf{x}_i = \mathbf{t}_i$ and $\mathbf{y}_i = \mathbf{r}_i$, then we say that agent i's strategy is *truthful*.

Opinions are elicited and aggregated by a central, trusted entity called the *mechanism*, which is also responsible for sharing the reward among the agents. This entity relies only on the reported opinions when determining agents' shares, and so it has no additional information. Formally:

DEFINITION 1 (MECHANISM). *A mechanism is a sharing function, $\Gamma : S \to \Re^n$, which maps each strategy profile to a vector of shares.*

We denote the share of V given to agent i, when all the reported opinions are $\mathbf{s}$, by $\Gamma_i(\mathbf{s})$. We use Γ_i when $\mathbf{s}$ is either irrelevant or clear from the context. Throughout this paper, we use the solution concept called *Bayes-Nash equilibrium*.

DEFINITION 2 (BAYES-NASH EQUILIBRIUM). *We say that the strategy profile $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_n)$ is a Bayes-Nash equilibrium if for each agent i, and strategy $\mathbf{s}'_i \neq \mathbf{s}_i \in S_i$, $\mathbb{E}\left[\Gamma_i(\mathbf{s}_i, \mathbf{s}_{-i}) | \mathbf{t}_i, \mathbf{r}_i\right] \geq \mathbb{E}\left[\Gamma_i(\mathbf{s}'_i, \mathbf{s}_{-i}) | \mathbf{t}_i, \mathbf{r}_i\right]$.*

In words, for each agent $i \in N$, $\mathbf{s}_i$ is the best response, in an expected sense, that agent i has to $\mathbf{s}_{-i}$ given its truthful opinions $(\mathbf{t}_i, \mathbf{r}_i)$. The expectation in taken with respect to the posterior distributions. When the inequality in Definition 2 holds strictly (with "$>$" instead of "$\geq$"), then the strategy profile $\mathbf{s}$ is called a *strict Bayes-Nash equilibrium*.

2.1 Properties

There are several key properties we wish mechanisms to have. We introduce them in this subsection.

DEFINITION 3 (FAIRNESS). *Consider a strategy profile $\mathbf{s} \in S$ in which the reported evaluation of every agent z for agent i is paired up with agent z's reported evaluation for agent j, for $i \neq j \neq z \in N$, so that $x_z^i > x_z^j$. Further, the evaluations of agent i and agent j for each other are paired up, so that $x_j^i > x_i^j$. Then, we say that a mechanism is fair if $\Gamma_i(\mathbf{s}) > \Gamma_j(\mathbf{s})$.*

In words, if an agent unanimously receives better evaluations than a peer, then that agent should also receive a greater share of the joint reward than its peer.

DEFINITION 4 (BUDGET BALANCE). *A mechanism is budget-balanced if $\forall \mathbf{s} \in S, \sum_{i=1}^{n} \Gamma_i(\mathbf{s}) = V$.*

In words, a budget-balanced mechanism allocates the entire reward V back to the agents. As stated, this is a strong definition because we do not put constraints on $\mathbf{s}$, *e.g.*, we do not require $\mathbf{s}$ to be an equilibrium strategy profile.

DEFINITION 5 (INDIVIDUAL RATIONALITY). *A mechanism is individually rational if $\forall i \in N, \forall \mathbf{s} \in S, \Gamma_i(\boldsymbol{s}) \geq 0$.*

This condition requires the share received by each agent to be greater than or equal to zero. In other words, all agents are weakly better off participating in the mechanism than not participating at all.

DEFINITION 6 (INCENTIVE COMPATIBILITY). *A mechanism is incentive-compatible if collective truth-telling is an equilibrium strategy profile.*

Since we are working with Bayes-Nash equilibrium, an incentive-compatible mechanism implies that it is best, in an expected sense, for each agent to tell the truth provided that the others are also doing so.

DEFINITION 7 (TRACTABILITY). *A mechanism is tractable if it computes agents' shares in polynomial time.*

By no means do we argue that the properties defined in this section are exhaustive. However, we believe that they are among the most desirable ones in practical applications.

2.2 The Bayesian Truth Serum Method

Prelec [12] proposes an incentive-compatible scoring method, called the *Bayesian Truth Serum* (BTS), which works on a single multiple-choice question with a finite number of alternatives. Each responder is requested to endorse the answer mostly likely to be true and to predict the empirical distribution of the endorsed answers.

Responders are evaluated by the accuracy of their predictions (how well they matched the empirical frequency) as well as how *surprisingly common* their answers are. For example, an answer endorsed by 50% of the population against a predicted frequency of 25% is surprisingly common. The responders who endorsed that answer should receive a high score. If predictions averaged 75%, an answer endorsed by 50% of the population would be *surprisingly uncommon* and, consequently, the responders who endorsed it would receive a lower score. The surprisingly common criterion exploits the *false consensus effect* to promote truthfulness, *i.e.*, the general tendency of responders to overestimate the degree of agreement that the others have with them [14].

In our work, the BTS method is used exclusively as a tool to promote truthfulness. This method is very convenient because it does not require objective answers to score opinions, *i.e.*, it is possible to work with subjective information, where an absolute truth is practically unknowable, and still be able to reward truthfulness. Questions that are considered in our work have the form: "What is the evaluation deserved by agent j?", where the possible answers are values inside the set $\{1, \ldots, M\}$. For illustration purpose, consider a question asking for the evaluation deserved by agent j. Using the notation previously defined, let $h(x_i^j, k)$ be a zero-one indicator function, *i.e.*,

$$h(x_i^j, k) = \begin{cases} 1 & \text{if } x_i^j = k, \\ 0 & \text{otherwise.} \end{cases}$$

The score returned by the BTS method to agent i, given its reported evaluation x_i^j and prediction y_i^j, is calculated as follows:

$$\mathbb{R}(i,j) = \sum_{k=1}^{M} h(x_i^j, k) \ln \frac{\bar{x}_k}{\bar{y}_k} + \sum_{k=1}^{M} \bar{x}_k \ln \frac{(1-\epsilon) {y_i^j}^k + \frac{\epsilon}{M}}{\bar{x}_k}, \quad (1)$$

where $\bar{x}_k$ is the average frequency of evaluation k, and $\bar{y}_k$ is the geometric average of the predicted frequencies of evaluation k:

$$\bar{x}_k = (1-\epsilon)\left(\frac{1}{n-1}\sum_{q \neq j} h(x_q^j, k)\right) + \frac{\epsilon}{M}$$

$$\bar{y}_k = \exp\left(\frac{1}{n-1}\sum_{q \neq j} \ln\left((1-\epsilon) {y_q^j}^k + \frac{\epsilon}{M}\right)\right)$$

and ϵ, for $0 < \epsilon < 1$, is a recalibration coefficient to adjust predictions and averages away from 0/1 extreme values.

The BTS method has two major components. The first one, called the *information score*, evaluates the evaluation given by agent i to agent j according to the log-ratio of its actual-to-predicted endorsement frequencies. An evaluation scores high to the extent that it is more common than collectively predicted. The second component, called the *prediction score*, is a penalty proportional to the relative entropy between the empirical distribution of evaluations for agent j and agent i's prediction of that distribution. For a small ϵ, the best prediction score is attained when a reported prediction matches the empirical distribution of evaluations.

It is interesting to note that Equation 1 is slightly different from the original BTS method, which uses $\epsilon = 0$. By using a small recalibration coefficient, we can avoid problems related to values that are not well-defined, *e.g.*, $\ln(0)$ and $\ln(0/0)$. Any distortion in incentives can be made arbitrarily small by making ϵ sufficiently small. Under the assumptions made in the beginning of this section, and using Equation 1 to compute agents' scores, the following theorems hold [12]:

THEOREM 1. *Collective truth-telling is a strict Bayes-Nash equilibrium.*

THEOREM 2. *The BTS method is zero-sum.*

Theorem 1 means that the strict best response of an agent, in an expected sense, when everyone else is telling the truth is also to tell the truth. Theorem 2 means that the sum of the scores received by the agents is equal to zero, *i.e.*, $\sum_{i \neq j} \mathbb{R}(i,j) = 0$. In what follows, we provide bounds for the scores returned by the BTS method.

LEMMA 1. $\forall i \neq j$, $\mathbb{R}(i,j) \in \left[-2\ln(\frac{M}{\epsilon}), \ln(\frac{M}{\epsilon})\right]$.

PROOF. We start by noting that:

$$0 < \frac{\epsilon}{M} \leq \bar{x}_k, \bar{y}_k \leq 1 - \epsilon + \frac{\epsilon}{M} < 1.$$

Focusing first on the lower-bound, we analyze each part of Equation 1 separately. Starting with the information score, we have:

$$\begin{aligned} \sum_{k=1}^{M} h(x_i^j, k) \ln \frac{\bar{x}_k}{\bar{y}_k} &\geq \sum_{k=1}^{M} h(x_i^j, k) \ln \bar{x}_k \\ &\geq \ln \frac{\epsilon}{M}, \end{aligned} \tag{2}$$

where the inequalities follow, respectively, from the facts that $0 < \bar{y}_k < 1$, and $\frac{\epsilon}{M} \leq \bar{x}_k < 1$. Moving to the prediction score, we have:

$$\begin{aligned} \sum_{k=1}^{M} \bar{x}_k \ln \frac{(1-\epsilon){y_i^j}^k + \frac{\epsilon}{M}}{\bar{x}_k} &\geq \sum_{k=1}^{M} \bar{x}_k \ln \frac{\epsilon}{M} \\ &= \left(1 - \epsilon + \frac{\epsilon}{M}\right) \ln \frac{\epsilon}{M} \\ &\geq \ln \frac{\epsilon}{M}, \end{aligned} \tag{3}$$

where the first inequality follows from the facts that $0 < \bar{x}_k < 1$ and $(1-\epsilon){y_i^j}^k \geq 0$. The second inequality follows from the facts that $\ln(\epsilon/M) < 0$, and $0 < \left(1 - \epsilon + \frac{\epsilon}{M}\right) < 1$. Joining (2) and (3), we have:

$$\begin{aligned} \mathbb{R}(i,j) &\geq 2 \ln \frac{\epsilon}{M} \\ &= -2 \ln \frac{M}{\epsilon}. \end{aligned}$$

Focusing now on the upper-bound of Equation 1, we start by analyzing the information score:

$$\begin{aligned} \sum_{k=1}^{M} h(x_i^j, k) \ln \frac{\bar{x}_k}{\bar{y}_k} &\leq \sum_{k=1}^{M} h(x_i^j, k) \ln \frac{1}{\bar{y}_k} \\ &\leq \ln \frac{1}{\frac{\epsilon}{M}} \\ &= \ln \frac{M}{\epsilon}, \end{aligned} \tag{4}$$

The inequalities follow from the fact that $\frac{\epsilon}{M} \leq \bar{x}_k, \bar{y}_k < 1$. Moving to the prediction score, we note that its value is always less than or equal to zero, because it can be seen as the negative of the Kullback-Leibler divergence, which is always greater than or equal to zero [3]. Thus, we have:

$$\mathbb{R}(i,j) \leq \ln \frac{M}{\epsilon}.$$

□

3. THE MECHANISM

In this section, we propose a mechanism for sharing rewards based on subjective opinions. It starts by requesting both evaluations and predictions from the agents. For each vector with evaluations, $\mathbf{x}_i$, the mechanism creates another vector, $\chi_i = (\chi_i^1, \ldots, \chi_i^{i-1}, \chi_i^{i+1}, \ldots, \chi_i^n)$, by scaling the elements of $\mathbf{x}_i$ so that they sum up to V. Mathematically,

$$\forall i, j,\ \chi_i^j = x_i^j \left(\frac{V}{\sum_{q \neq i} x_i^q} \right). \tag{5}$$

This simple pre-processing step ensures that the sum of the resulting shares is not orders of magnitude lower than the reward V. The share received by each agent $i \in N$ from the mechanism has two major components. The first one, $\bar{\chi}^i$, reflects agent i's received evaluations. It is calculated by summing the scaled evaluations received by agent i, and dividing the sum by n, *i.e.*,

$$\bar{\chi}^i = \frac{\sum_{j \neq i} \chi_j^i}{n}. \tag{6}$$

This simple idea of aggregating the scaled evaluations for an agent by summing them and dividing by n helps to ensure important properties for the mechanism. The second component of agent i's share is a truth-telling score. The intuition behind such scores is that agents who believe that the others are telling the truth maximize their expected scores by also telling the truth. The score of agent i, ζ_i, is calculated as follows:

$$\zeta_i = \frac{\sum_{j \neq i} \mathbb{R}(i,j)}{n-1}, \tag{7}$$

where $\mathbb{R}(i,j)$ is defined in Equation 1. Agent i's score is then the arithmetic mean of results returned by the Bayesian truth serum method, where each result is directly related to an evaluation and a prediction reported by agent i. Finally, the share of agent i is a linear combination of $\bar{\chi}^i$ and ζ_i, *i.e.*,

$$\Gamma_i = \bar{\chi}^i + \alpha\, \zeta_i, \tag{8}$$

where the constant α, for $\alpha > 0$, fine-tunes the weight given to the truth-telling score ζ_i. Its value has an important role in ensuring desirable properties for the mechanism.

The intuition behind the proposed mechanism is that agents who believe that the others are truthfully reporting have their expected shares maximized to the extent that they are well-evaluated and that they are also telling the truth. It is interesting to note that despite the assumptions of prior and posterior distributions, they are neither known nor requested by the mechanism, only evaluations and predictions are elicited from agents.

3.1 Numerical Example

A numerical example may clarify the mechanics of the proposed mechanism. Consider six agents indexed by the letters A, B, C, D, E, F, a joint reward $V = 1000$, and assume that $M = 2$. The reported predictions and evaluations can be seen, respectively, in Table 1 and Table 2.

In Table 1, each numeric cell can be interpreted as the prediction made by the agent in the row about the percentage of agents that give the evaluation in the second row of

Table 1: Numerical example: reported predictions.

	A		**B**		**C**		**D**		**E**		**F**	
	"1"	**"2"**	**"1"**	**"2"**	**"1"**	**"2"**	**"1"**	**"2"**	**"1"**	**"2"**	**"1"**	**"2"**
A	-	-	0	1	0.4	0.6	0.2	0.8	1	0	0.2	0.8
B	*0.8*	0.2	-	-	0.2	0.8	0.2	0.8	1	0	0.4	0.6
C	0.8	0.2	0	1	-	-	0.4	0.6	1	0	0.4	0.6
D	0.8	0.2	0.2	0.8	0.6	0.4	-	-	0.8	0.2	0.4	0.6
E	0.8	0.2	0	1	0.6	0.4	0.4	0.6	-	-	0.4	0.6
F	0.8	0.2	0.8	0.2	0.6	0.4	0.4	0.6	0.8	0.2	-	-

the cell's column (**"1"** or **"2"**) to the agent in the first row of the cell's column. For example, the emphasized number 0.8 means that agent B predicts that 80% of the population gives the evaluation 1 to agent A.

In Table 2, each numeric cell can be interpreted as the evaluation given by the agent in the row to the agent in the column. For example, the emphasized number 2 represents x_A^B, *i.e.*, the evaluation given by agent A to agent B.

Using these evaluations and predictions, and the parameters $\alpha = 100$ and $\epsilon = 0.01$, the mechanism returns the shares shown in the last column of Table 3. The major components of these shares are shown in the first columns. For illustration's sake, consider the share received by agent F. To compute the first component of Γ_F, the mechanism aggregates the scaled evaluations received by agent F (Equation 6):

$$\begin{aligned}\bar{\chi}^F &= \frac{142.86 + 250.00 + 285.71 + 250.00 + 222.22}{6} \\ &\approx 191.80.\end{aligned}$$

The second component of Γ_F is the arithmetic mean of results returned by the BTS method, where each result is directly related to an evaluation and a prediction submitted by agent F (Equation 7):

$$\begin{aligned}\zeta_F &= \frac{\mathbb{R}(F,A) + \mathbb{R}(F,B) + \mathbb{R}(F,C) + \mathbb{R}(F,D) + \mathbb{R}(F,E)}{5} \\ &\approx \frac{0.58 - 1.19 - 0.18 - 0.11 - 0.11}{5} \\ &\approx -0.21.\end{aligned}$$

Finally, the share received by agent F from the mechanism is a linear combination of $\bar{\chi}^F$ and ζ_F:

$$\begin{aligned}\Gamma_F &= \bar{\chi}^F + \alpha\,\zeta_F \\ &= 191.80 + 100 \times (-0.21) \\ &= 170.80.\end{aligned}$$

Table 2: Numerical example: reported evaluations.

	A	**B**	**C**	**D**	**E**	**F**
A	-	*2*	2	1	1	1
B	1	-	2	2	1	2
C	1	2	-	1	1	2
D	1	2	2	-	1	2
E	2	2	1	2	-	2
F	2	2	1	2	1	-

Table 3: Numerical example: resulting shares.

	$\bar{\chi}^i$	ζ_i	Γ_i
A	144.18	0.05	149.18
B	215.61	-0.06	209.61
C	170.30	0.09	179.30
D	167.99	-0.02	165.99
E	110.12	0.15	125.12
F	191.80	-0.21	170.80

3.2 Properties

In this subsection, we show that the proposed mechanism satisfies important properties.

PROPOSITION 1. *The mechanism is budget-balanced.*

PROOF. The sum of the shares received by the agents is equal to:

$$\begin{aligned}\sum_{i=1}^{n}\left(\bar{\chi}^i + \alpha\,\zeta_i\right) &= \sum_{i=1}^{n}\bar{\chi}^i + \alpha\sum_{i=1}^{n}\zeta_i \\ &= \sum_{i=1}^{n}\frac{\sum_{j\neq i}\chi_j^i}{n} + \alpha\sum_{i=1}^{n}\frac{\sum_{j\neq i}\mathbb{R}(i,j)}{n-1} \\ &= \sum_{j=1}^{n}\frac{\sum_{i\neq j}\chi_j^i}{n} + \alpha\sum_{j=1}^{n}\frac{\sum_{i\neq j}\mathbb{R}(i,j)}{n-1} \\ &= n\left(\frac{V}{n}\right) + \frac{\alpha}{n-1}\left(\sum_{j=1}^{n}\sum_{i\neq j}\mathbb{R}(i,j)\right).\end{aligned}$$

The last equality follows from the fact that the scaled evaluations sum up to V (Equation 5). From Theorem 2, we know that $\sum_{i\neq j}\mathbb{R}(i,j) = 0$, thus completing the proof. □

PROPOSITION 2. *The mechanism is incentive-compatible.*

PROOF (SKETCH). Due to space limitations, we only provide a sketch of the proof. Suppose that every peer of an agent $i \in N$ is truthfully reporting its opinions. We prove that the strict best response for agent i, in an expected sense, is also to tell the truth. We start by observing that the share received by agent i (Equation 8) can be written as $c_1 + c_2\sum_{j\neq i}\mathbb{R}(i,j)$, where c_1 and c_2 are positive constants, from agent i's point of view, because they do not depend on the opinions reported by agent i. Due to the assumption of independent signals (Assumption 6, Section 2), we can restrict ourselves to find the strategy of agent i that maximizes $\mathbb{E}\left[c_1 + c_2\mathbb{R}(i,j)\right] = c_1 + c_2\mathbb{E}\left[\mathbb{R}(i,j)\right]$, which in turn is strictly maximized when agent i tells the truth (Theorem 1). Thus, the mechanism is incentive-compatible. □

PROPOSITION 3. *The mechanism is tractable.*

PROOF. The pre-processing step (Equation 5) is computed in $O(n^2)$. Thereafter, for each agent $i \in N$, Equation 6 is computed in $O(n)$, Equation 7 is computed in $O(n^2M)$ (since Equation 1 can be computed in $O(nM)$), and Equation 8 is computed in $O(1)$. Thus, the mechanism runs in $O(n^3M)$ time. □

PROPOSITION 4. *If* $M \leq \sqrt{n-2}$ *and* $\alpha \leq \frac{V}{3Mn^2 \ln\left(\frac{M}{\epsilon}\right)}$, *then the mechanism is fair.*

PROOF. Consider a pair of agents $i, j \in N$ and a strategy profile $\mathbf{s} \in S$ where $x_j^i > x_i^j$ and, for every other agent $z \neq i, j$, $x_z^i > x_z^j$. For the mechanism to be considered fair, its resulting shares must satisfy the following inequality:

$$\begin{aligned} \Gamma_i(\mathbf{s}) > \Gamma_j(\mathbf{s}) &\equiv \bar{\chi}^i + \alpha\,\zeta_i > \bar{\chi}^j + \alpha\,\zeta_j \\ &\equiv \alpha < \frac{\bar{\chi}^i - \bar{\chi}^j}{\zeta_j - \zeta_i}. \end{aligned} \tag{9}$$

In what follows, we compute a lower-bound for the above fraction. Starting with the numerator, we have:

$$\begin{aligned} \bar{\chi}^i - \bar{\chi}^j &= \frac{\sum_{z\neq i,j}\left(x_z^i - x_z^j\right)\left(\frac{V}{\sum_{q\neq z} x_z^q}\right) + x_j^i\left(\frac{V}{\sum_{q\neq j} x_j^q}\right)}{n} \\ &\quad - \frac{x_i^j\left(\frac{V}{\sum_{q\neq i} x_i^q}\right)}{n} \\ &\geq \frac{V}{n}\left(\frac{n-2}{(n-1)M} + \frac{1}{(n-1)M} - \frac{M}{(n-1)}\right) \\ &= \frac{V}{n}\left(\frac{n-2+1-M^2}{(n-1)M}\right) \\ &\geq \frac{V}{n(n-1)M} \\ &\geq \frac{V}{n^2M}. \end{aligned}$$

The first inequality follows from the facts that for every agent $z \neq i, j$, $x_z^i > x_z^j$ and $\forall i, j$, $x_i^j \in \{1, \ldots, M\}$. The second inequality follows from the assumption that $M \leq \sqrt{n-2}$. Focusing on the denominator of the fraction in (9), since $\forall q \in N, \zeta_q$ is the average of $n-1$ results from the BTS method, then the difference between ζ_j and ζ_i is always less than or equal to the difference between the highest and the lowest scores that can be returned by the BTS method (Equation 1), which is equal to $3\ln\left(\frac{M}{\epsilon}\right)$ according to Lemma 1. Thus, we conclude that if:

$$\alpha \leq \frac{V}{3Mn^2 \ln\left(\frac{M}{\epsilon}\right)},$$

and $M \leq \sqrt{n-2}$, then the proposed mechanism is fair. □

Intuitively, this proposition means that the proposed mechanism can be made fair by reducing the influence of the truth-telling scores on agents' shares, so that these shares will depend almost entirely on the reported evaluations.

PROPOSITION 5. *If* $\alpha \leq \frac{V}{2Mn\ln\left(\frac{M}{\epsilon}\right)}$, *then the mechanism is individually rational.*

PROOF. We start the proof by observing that $\forall i \in N$, $\bar{\chi}^i \geq 0$ (Equation 6). Consequently, if agents' scores are positive, then their shares will also be positive. So, we restrict ourselves to the scenario where truth-telling scores are negative. Thus, for every agent $i \in N$, the following inequality must be true when $\zeta_i < 0$:

$$\bar{\chi}^i + \alpha\,\zeta_i \geq 0 \equiv \frac{\bar{\chi}^i}{-\zeta_i} \geq \alpha. \tag{10}$$

In what follows, we compute a lower-bound for the fraction in (10). Starting with the numerator, we have:

$$\begin{aligned} \bar{\chi}^i &= \frac{\sum_{j\neq i} x_j^i\left(\frac{V}{\sum_{q\neq j} x_j^q}\right)}{n} \\ &\geq \frac{\sum_{j\neq i} x_j^i\left(\frac{V}{M\,(n-1)}\right)}{n} \\ &\geq \frac{V(n-1)}{M\,n(n-1)} \end{aligned}$$

The inequalities follow from the fact $\forall i, j$, $x_i^j \in \{1, \ldots, M\}$. Focusing on the denominator of the fraction in (10), since ζ_i is the average of $n-1$ results from the BTS method, we can restrict ourselves to find the lowest negative score that can be returned by the BTS method. From Lemma 1, we know that this value is $-2\ln\left(\frac{M}{\epsilon}\right)$. Thus, we conclude that if:

$$\alpha \leq \frac{V}{2Mn\ln\left(\frac{M}{\epsilon}\right)},$$

then the proposed mechanism is individually rational. □

Since agents' scores can be negative, the above proposition means that the resulting shares can always be positive, regardless the reported evaluations and predictions, if we reduce the influence of these scores on agents' shares.

4. EMPIRICAL EVALUATION

In this section, we report an empirical investigation of the influence of the model and mechanism's parameters on agents' shares. In all experiments reported here, agents' truthful evaluations are drawn from the probability distribution of the random variable $\mathbb{H} = \lceil M\mathbb{B} \rceil$, where $\mathbb{B}$ is Beta-distributed with parameters $\alpha = \beta = 0.5$, *i.e.*, $\mathbb{B}$ has a symmetric, U-shaped distribution. For creating a random prediction, we use the empirical distribution of $n-1$ evaluations drawn from the probability distribution of $\mathbb{H}$. Thus, the experiments reflect scenarios where most of the agents have extreme opinions. Lastly, agents always report their opinions truthfully.

4.1 Parameter M

The parameter M defines the range of possible evaluations that an agent can give or receive. To better understand the influence of different values of M on agents' shares, we performed the following experiment. We shared the reward $V = 1000$ among 100 agents using the proposed mechanism and the following values for M: $2, 5, 7, 10, 25, 50, 75, 100$. We used the parameters $\alpha = 10$ and $\epsilon = 10^{-4}$, and we observed the mean and the standard deviation of the resulting shares for different values of M. Figure 1 shows the results.

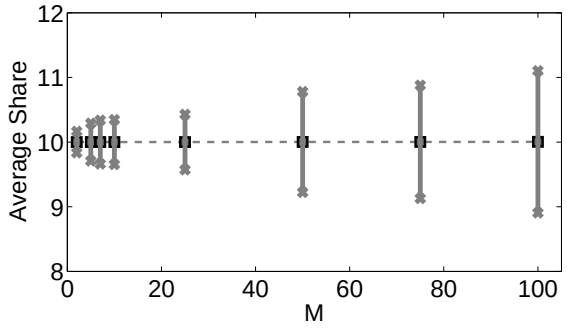


Figure 1: Results of the experiment with different values for M. Average shares are represented by black squares, and standard deviations by gray lines. The dotted line is used to facilitate visualization.

As can be seen in Figure 1, as M increases, the standard deviation of the resulting shares also increases. Intuitively, this happens because the reported evaluations become more fine-grained, in that small differences between agents are recognized and specified by their peers, thus resulting in more diverse shares. It is important to note that this increased expressivity may be burdensome for the agents since they will have more possibilities to evaluate their peers, thus making the evaluation process more challenging. We argue that the underlying application may help to determine appropriate settings for M. Since the mechanism is budget-balanced and we used a fairly large population in this experiment, the average share stayed constant for different values of M.

4.2 Parameter α

The parameter α of the proposed mechanism fine-tunes the weight given to the truth-telling scores. To better understand its influence on agents' shares, we performed the following experiment. We shared the reward $V = 1000$ among 100 agents using the parameters $M = 10$, $\epsilon = 10^{-4}$, and $\alpha \in \{0.1, 1, 5, 10, 25, 50, 100, 500\}$. We ran this experiment 100 times. We observed the total number of unfair shares and the total number of negative shares returned by the mechanism for different values of α. An agent's share is considered unfair if that agent unanimously receives better evaluations than a peer, but its share is smaller than the peer's share. Thus, a mechanism is fair if it does not return unfair shares (see Definition 3). To compute the number of unfair shares, we made a pairwise comparison in each simulation step in which each returned share was compared to each other for determining whether the former was unfair or not. Table 4 presents the results of this experiment.

Table 4: Results of the experiment with different values for α.

α	Unfair shares	Negative shares
0.1	0	0
1	0	0
5	0	0
10	0	0
25	0	0
50	0	0
100	0	8
500	0	2543

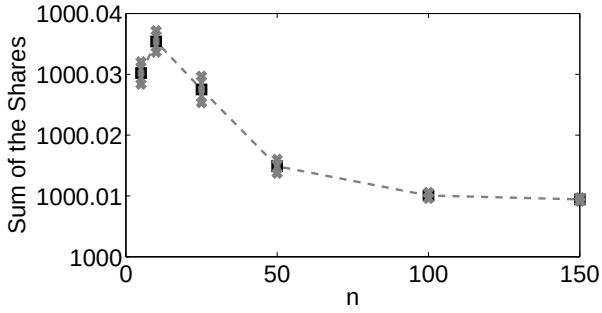


Figure 2: Results of the experiment with different values for n. Black squares represent averages of the sum of the shares, and gray lines represent standard deviations. The dotted line is used to facilitate visualization.

According to Proposition 4 and 5, we need to set $\alpha < 2.9 \times 10^{-4}$ to mathematically ensure that the mechanism will be fair, and $\alpha < 0.044$ to mathematically ensure that the returned shares will always be greater than or equal to zero. From Table 4, we note that even with much higher values for α, the mechanism did not return a single unfair share in this experiment. Further, the mechanism did not return a single negative share for $\alpha \leq 50$. This discrepancy between experiment and theory can be ascribed to the fact that the bounds for α are calculated based on worst-case scenarios, which are very unlikely to happen in practical applications. This implies that it is possible to promote truthfulness by using high values for α and still be able to obtain individual rationality and fairness.

4.3 Parameter n

The most stringent assumption made in this work is that the population of agents is large. This assumption is necessary for the proposed mechanism to be able to use the BTS method. We performed an experiment to investigate how this mechanism behaves when dealing with populations of different sizes. In detail, we studied how the size of the population affects the budget of the mechanism. We shared the reward $V = 1000$ using the parameters $M = 10$, $\alpha = 10$, $\epsilon = 10^{-4}$, and $n \in \{5, 10, 25, 50, 100, 150\}$. We executed the experiment 100 times. At the end of each simulation step, we computed the sum of the returned shares for each value of n. At the end of the experiment, we computed the averages and the standard deviations of these sums. Figure 2 shows the results.

As can be seen in Figure 2, the mechanism loses more when $n \leq M$. Intuitively, since there are few agents to endorse a larger number of possible evaluations, the reported evaluations are very often surprisingly common. This implies higher truth-telling scores for the agents and, consequently, greater shares. Alternatively, agents' scores are more balanced when $n > M$. Since there are more agents than evaluations to be endorsed, the reported evaluations are not very often surprisingly common. Consequently, the average truth-telling score is not so high, and the mechanism's loss gradually decreases. An ANOVA test confirms that n does indeed influence the resulting shares ($\rho < 0.0001$). The standard deviation of the sum of the shares also decreases when n increases, thus supporting our claim that the scores are more balanced.

In conclusion, we note that a possible way to circumvent the assumption of a large population is to reduce the number of possible evaluations, *i.e.*, to reduce the value of the parameter M. In this way, the influence of a single agent on the empirical distributions of evaluations may be reduced since these distributions will probably (but not necessarily) be more balanced. We suggest that a good rule of thumb is to use a value for the parameter $M \leq \sqrt{n-2}$, because at this point the number of different evaluations seems to be sufficiently smaller than the number of agents. Also, a value for M satisfying this inequality helps to mathematically ensure fairness (Proposition 4). This rule has a strong empirical support in our experiment because the loss taken by the mechanism is negligible when the inequality is satisfied, *i.e.*, for $n = 100$ and $n = 150$.

5. RELATED WORK

Fair division has long been studied in cooperative game theory. The *Shapley value* [11] is a key concept used in this field to distribute a joint surplus (or cost) among a set of agents. Roughly speaking, the Shapley value assigns a share to each agent equal to that agent's marginal contribution to the group. We note that sharing schemes based on marginal contributions, like the Shapley value, are not appropriate in our setting. The idea of marginal contribution is not objectively defined in our model because individual contributions are subjective information.

In the context of cooperative learning, Oakley *et al.* [10] propose some guidelines to the effective design and management of teams of students. Slightly different from our model, each team member receives a common grade as the result of a joint academic work. These grades are adjusted through peer ratings (evaluations) to account for individual performance. In detail, a team grade is weighted by the average evaluation that a student receives to determine his or her final grade. A total of 9 verbal evaluations are used, which are later converted to values inside the set $\{0, 12.5, 25, ...100\}$. Differently from our work, this rating scheme allows agents to make self-evaluations. Further, it does not promote truthfulness. Kaufman *et al.* [6] discuss the problems that may arise when using this rating system, *e.g.*, inflated self-evaluations and gender and racial bias. We believe that these problems may be even worse in our scenario because there is a joint reward to be shared, and not a common team grade.

Hence, the BTS method is an important component of our mechanism. We note that similar incentive-compatible methods which would require less information from the agents (*i.e.*, only evaluations) could have been used (*e.g.*, [8, 5]). However, most of these methods are not budget-balanced, which we believe is an important property in our setting.

6. CONCLUSION

In this paper, we proposed a game-theoretic model for sharing a joint, homogeneous reward based on the idea of subjective opinions. Each agent is asked to evaluate its peers as well as to predict how they will be evaluated. We introduced a mechanism to aggregate and use such opinions for determining agents' shares. The intuition behind the proposed mechanism is that each agent who believes that the others are telling the truth has its expected share maximized to the extent that it is well-evaluated and that it is truthfully reporting its opinions. Under the assumptions that agents are Bayesian decision-makers and that the underlying population of agents is sufficiently large, we showed that the proposed mechanism is incentive-compatible, budget-balanced, and tractable. We also presented strategies to make this mechanism individually rational and fair.

We implicitly assumed that agents are not participating in collusive agreements. However, there are many reasons why an agent may lie to benefit a peer. For example, in exchange for misreporting its evaluation, which may lead to a lower share for itself, a liar agent may receive a side-payment from the agent who benefits from the misreporting. Thus, an exciting direction for future research work is to study which kinds of collusive behavior may arise and how to avoid them.

7. REFERENCES

[1] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[2] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6):607–619, 2006.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

[4] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[5] R. Jurca and B. Faltings. Incentives for expressing opinions in online polls. In *Proceedings of the 2008 ACM Conference on Electronic Commerce*, pages 119–128. ACM, July 2008.

[6] D. B. Kaufman, R. M. Felder, and H. Fuller. Accounting for individual effort in cooperative learning teams. *Journal of Engineering Education*, 89(2):133–140, 2000.

[7] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[8] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[9] H. Moulin. *Fair Division and Collective Welfare*. The MIT Press, 2004.

[10] B. Oakley, R. M. Felder, R. Brent, and I. Elhajj. Turning student groups into effective teams. *Journal of Student Centered Learning*, 2(1):8–33, 2004.

[11] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[12] D. Prelec. A Bayesian truth serum for subjective data. *Science*, 306(5695):462–466, 2004.

[13] T. Rahwan, S. D. Ramchurn, N. R. Jennings, and A. Giovannucci. An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research*, 34(1):521–567, 2009.

[14] L. Ross, D. Greene, and P. House. The "false consensus effect": An egocentric bias in social perception and attribution processes. *Journal of Experimental Social Psychology*, 13(3):279–301, 1977.

[15] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

Capability-Aligned Matching: Improving Quality of Games with a Purpose

Che-Liang Chiou
Department of Computer Science and Information Engineering
National Taiwan University
clchiou@gmail.com

Jane Yung-Jen Hsu
Department of Computer Science and Information Engineering
National Taiwan University
yjhsu@csie.ntu.edu.tw

ABSTRACT

So far computer cannot satisfyingly solve many tasks that are extremely easy for human, such as image recognition or common sense reasoning. A partial solution is to delegate algorithmically difficult computation task to human, called human computation. The Game with a Purpose (GWAP), in which computational task is transformed into a game, is perhaps the most popular form of human computation. A simplified adverse selection model for output-agreement/ simultaneous-verification GWAP was built, using the ESP Game as example. The experiment results favored an adverse selection model over an moral hazard model. We were particularly interested in output quality of a GWAP affected by how players are matched with each other, and proposed capability-aligned matching (CAM) versus commonly-used random matching. The analysis showed that when compared with random mathcing, the CAM improved output quality. The experiment confirmed conclusions drawed from the analysis, and further pointed out that task-human matching scheme was as important as human-human matching scheme studied in this paper. The main contribution of this paper is the analysis and empirical evaluation of human-human matching scheme, showing that capability-aligned matching can improve quality of GWAP.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Economics, Experimentation

Keywords

Game with a purpose, Adverse selection, Mechanism design

1. INTRODUCTION

The Game with a Purpose (GWAP) is a computer game designed to perform computation tasks as a by-product [12]. It is targeted for algorithmically difficult problems that are easy for human. Generally the GWAP are used for two purposes: (1) Solve algorithmically difficult problems. (2) Generate and/or annotate datasets for further research.

The ESP Game [11] is used as the primary example due to two reasons. (1) Because the ESP Game is the first GWAP, much of its design is widely used in many GWAP, such as output agreement, simultaneous verification, and random player matching. (2) From a game theory perspective, the ESP Game presents a fundamental type of game: static game. The analysis of the ESP Game is the basic for more complicated games, say, repeated games.

The ESP Game is used to illustrate the poor quality problem of a class of GWAPs, the output-agreement/simultaneous-verification games [12, 5].

1.1 Motivation

There are two orthogonal properties of outputs generated by a GWAP: correctness and quality. This paper addressed the quality of outputs. What is "correct" or "of good quality" depends on the nature of computation task; this paper defined them in the context of the ESP Game.

The ESP Game is designed to annotate images, and outputs are labels. A label is correct to an image if it describes the image. This paper defines a label is of good quality relative to another label based on their specificness. That is, labels are ordered by an "is-a" relation. For example, "red" is of better quality than "color" because red is a color but not vice-versa.

Figure 1 shows the relationship between correctness and quality. It is possible that a label is of good quality but incorrect to an image (the quadrant II). For example, "Lincoln" is of better quality than "man" but incorrect to a photo of Washington. Note: The correctness is bounded to an image, but the quality is a relative relation among labels.

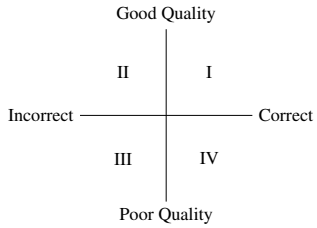


Figure 1: The four quadrants of a label.

In the original paper of the ESP Game [11], it has been shown that output labels are descriptions of the images, *i.e.,* correct. In a later study [10], the Google's implementation

Cite as: Capability-Aligned Matching: Improving Quality of Games with a Purpose, Che-Liang Chiou and Jane Yung-Jen Hsu, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 643-650.

of the ESP Game, Google Image Labeler, is examined. It is observed that players tend to answer generic labels such "building" as opposed to "terraced house", *i.e.,* correct but of poor quality.

Even worse, it has been found that output labels are predictable by a low entropy distribution. This means that a computer *without looking at the image* can guess what labels a player will output (this is exactly the cheating defined in [11]). Given that image annotation is algorithmically difficult, the predictability of outputs suggests that players are not properly motivated to outperform computers. Besides, if output labels are computer predictable, why do we need human in the course of computation?

To summarize, prior experiments have shown that output labels are correct but of poor quality (quadrant IV). Google apparently noticed this and implemented a variable scoring scheme according to specificity of labels, rather than a flat-rate scheme used in the original ESP Game. Many GWAP might also suffer from the poor quality problem because they share much of the same design of the ESP Game.

From a game theory perspective, the poor quality problem stems from the use of coordination game in the ESP Game. The focal points, also called Schelling point, of coordination GWAP are "generic label" of the ESP Game. The poor quality problem is equivalent to the existence of focal points.

The original ESP Game has implemented taboo words as an instrument to improve output quality. This is how taboo words work: When the ESP Game verifies a label, it is listed on the taboo list. Eventually all labels in the quadrant IV are in taboo list, and players can only output labels in the quadrant I—correct and of good quality.

The main shortcoming of taboo words is inefficiency: Why not simply motivate players to output quadrant I labels?

1.2 Overview of Proposed Solution

The *capability-align matching* (CAM) is proposed for solving the poor quality problem. Two types of of matching are in the ESP Game. (1) Matching a player with another. (2) Matching an image to a pair of players. The CAM is the former.

The CAM matches players with similar if not identical capability. On the other hand, the current implementation of the ESP Game (intentionally) use random matching.

Note: A critical game-theoretical requirement of the CAM is that which matching scheme is used is common knowledge among players.

The CAM is implemented in a small-scale experiment. The implemented method is called the *Segmentation* method, which extracts capability information from demographic data.

2. RELATED WORK

The ESP Game, the first GWAP, is designed to annotate images and is shown to be effective on generating label-descriptions of an image [11]. This simple game demonstrates that designing a game to use human to perform computation task is possible. Since then, many of GWAP follow its design.

Three designs of the ESP Game are relevant to modeling. They are also widely used in many GWAP.

- *Random player matching.* When compared with the CAM, it incurs poor quality output.
- *Isolated players.* This has a great strategical consequence: the ESP Game is a static game.
- *Output agreement/Simultaneous Verification.* (For its definition, see [12, 5].) It is equivalent to coordination game.

In theory, the ESP Game is a static coordination game.

The following are overview of previous approaches dealt that could be used to improve output quality.

Incentive provision. This approach tries to "manipulate" players through incentive [10, 6], either money or score. Its goal is to implement a designer-chosen good quality outcome. Nevertheless, in some cases incentive-provision along might be ineffective. As shown by experiments [9], increased financial incentive does not necessary increase quality. The CAM can reduce the amount of incentive required for implementing a good quality outcome.

Competition. This approach is based a game structure called zero-sum games. The Search War [8] is a two-player zero-sum game. The KKB [4] adds a zero-sum sub-game to the ESP Game. Nevertheless, in theory competition does not improve quality, at least in the sense of specificness, but it does diversify output—when the equilibrium is mixed. Zero-sum games often have only mixed equilibrium due to their strictly competitive nature. The zero-sum game that the Search War and the KKB use is called matching pennies, which has only a unique mixed equilibrium.

Note: This paper uses "competition" in a strict game-theoretic sense. There are GWAP [3] that are competitive (in ordinary sense) but not zero-sum. These games are coordination games, and players only compete for first proposing the will-be-agreed output.

Community. This approach is loosely defined by the use of social network or demographic data. It could be used for drawing players from Facebook[1], for annotating your friends [1], or for improving output quality [7]. The CAM may use communities for extract player's capability information, called the Segmentation method.

3. A SPECIAL THEORY OF CAPABILITY-ALIGNED MATCHING

Here we present a special version of the theory of the CAM. The general version of the theory is published in another venue due to page limits [2].

The analysis of the one-shot ESP Game is carried out by comparing the CAM in a hypothetical, ideal scenario to the random matching so that the theoretical maximal improvement of the CAM is derived. The hypothetical, ideal scenario is referred to as the first-best model, in which a computer has complete information of players' capability. The scenario of the random matching is referred to as the second-best model, in which a computer has only incomplete information of players' capability.

The performance of an outcome is defined in three aspects. These are used for comparing the first- to the second-best model.

The first aspect considers the quality, or the "revenue". It is possible that a best quality outcome is too costly to implement. The following results all condition on that there is a sufficient "margin of profit" (revenue minus cost) so that implementing a best quality outcome is in equilibrium.

[1] `http://apps.new.facebook.com/fb_gwap/`

The second aspect considers the amount of incentive provision, or the "cost". Given a particular outcome to be implemented, this amount should be as little as possible.

The last aspect considers the agreement rate, or the "risk of the business". This probability is a measure of efficiency of a game. The higher the agreement rate, the faster a correct label is produced. And why is that? The ESP Game is like a Las Vegas algorithm due to the verifying by agreement nature of it. It always produce correct labels, but its computation time varies randomly (we do not know when a pair of agents will agree).

3.1 The One-Shot ESP Game Model

We formulate a direct mechanism of the ESP Game, called one-shot ESP Game, in a special setting. This model is used for demonstrating the theory we will examine in experiments.

The strategic interaction in GWAP between a computer and players, in economics terminology, is a *principal-agent* relation. A computer (as a principal) hires players (as agents) to perform computation tasks. The following is the basic setup:

Number of labels/types. $n+1$.

Index of labels/types. $0 \le k, l \le n$. Note: Because k and l may index either label or type, we use superscript for labels, and subscript for types.

Set of labels. $W = \{w^k \mid 0 \le k \le n\}$.

Qualities of labels. $q^k = \alpha k + \beta$ for label w^k where α and β are real constants.

Agent (player). There are two agents, 1 and 2, indexed by $i \ne j \in \{1, 2\}$. The term "agent" and "player" are used interchangeably.

Agent's output. $w_{k,i}$ denote output label of a type-V_k, index i agent. The index of player is often dropped because the ESP Game is symmetric, that is, w_k.

Utility of agents. $u(p) = \sqrt{p}$. Let $v = u^{-1}$ for the ease of notation. Agents are assumed to be homogeneous so that we will not be distracted from minor issues like private information of agent's utility function.

Reservation utility $\underline{u} > 0$. Let $\underline{v} = v(\underline{u})$ for the ease of notation.

Capability of an agent (type). $V_k = \{w^0, \ldots, w^k\}$. In the ESP Game, the capability of an agent is his vocabulary of words he can use. The term "capability" and "type" are used interchangeably.

Type space. $\mathbf{V} = \{V_k \mid 0 \le k \le n\}$. The set of capabilities, also referred to as "the type space".

Distribution of types. $\mu_k = \frac{1}{2^n} \mathrm{C}_k^n$ for type V_k. μ_k is the proportion of type-V_k players. It is a binomial distribution so that most agents have moderate capability and few agents are at extreme.

Payoffs. The principal chooses the payoffs. In the first-best model, payoffs may contingent on both output label and type, denoted by $p(\cdot)$. In the second-best model, payoffs are contingent on output label only, denoted by p^k. Let $u^k = u(p^k)$ for the ease of notation.

The one-shot ESP Game is a ESP Game that a player only outputs one label. It is played as follows:

0. The quality function q is given to the principal.
1. The principal chooses a payoff function p, and matches two agents from a pool of agents.
2. The agents observe the payoff function, and then decide whether to play (note that at this point, the agents know what matching scheme is in charge).
   (a) If any agent decides not to play, then the game terminates; the principal receives 0, and the agents both receive $\underline{u}$.
   (b) Otherwise, the game proceeds to the next step.
3. The agents simultaneously output a label w_i.
4. (a) If the agents agree on w, *i.e.,* $w = w_1 = w_2$, then the agents win; the principal receives $q(w) - p(w)$, and the agents both receive $u(p(w))$.
   (b) Otherwise, the agents lose; the principal and the agents all receive 0.

Note: For the ease of notation, the payoff to the agents is also written as

$$p(w_{k,1}, w_{l,2}) = \begin{cases} p(w) & w_{k,1} = w_{l,2} = w \\ 0 & w_{k,1} \ne w_{l,2}, \end{cases}$$

or in the unit of utility $u(w_{k,1}, w_{l,2}) = u(p(w_{k,1}, w_{l,2}))$, and the payoff to the principal

$$\pi(w_{k,1}, w_{l,2}) = \begin{cases} q(w) - p(w) & w_{k,1} = w_{l,2} = w \\ 0 & w_{k,1} \ne w_{l,2}. \end{cases}$$

A cautious reader might wonder why the payoff of the principal is not $q(w) - 2p(w)$. The reason is the ease of notation. Because only the relative order of q-value matters in this thesis, q can be linearly scaled up arbitrarily, and whether the principal receives $q - p$ or $q - 2p$ does not matter.

3.2 The First-Best Model

The first-best model is our benchmark; it is the best possible performance the CAM can achieve. In the first-best model, the principal has complete information of capabilities, and players are perfectly aligned, that is, $k = l$.

The payoff function p is subjected to two constraints due to the rationality of agents.

Individual rationality.

$$u(w_{k,1}, w_{k,2}) \ge \underline{u}. \tag{IR1}$$

Incentive compatibility.

$$w_{k,i} \in \arg\max_{w \in V_{k,i}} u(w, w_{k,j}). \tag{IC1}$$

The principal maximizes the average payoff

$$\max_{p, \{w_0, \ldots, w_n\}} \sum_{0 \le k \le n} \mu_k \pi(w_k, w_k) \tag{P1}$$

subjected to (IR1) and (IC1).

Obviously, the maximization program is solved by

$$p(w_k) = \begin{cases} \underline{v} & w_k = w^k \\ 0 & w_k \ne w^k. \end{cases} \tag{1}$$

That is, the agent is not paid unless he outputs a best quality label he can think of. Under this payoff function, the output label w_k is, not surprisingly, the best quality output w^k. It is easy to check if this outcome satisfies (IR1) and (IC1). And agents always agree in equilibrium because it is a symmetric game.

So in the first-best model, the principal implements

- Best quality outcome $w_k = w^k$,
- Using minimal incentive provision $\underline{v}$,
- With perfect agreement rate equals to 1.

This too-good-to-be-true performance of first-best equilibrium shows the power of capability information in an idealize scenario. Rarely can a real world game have 100% complete capability information; there is always uncertainty in practice.

3.3 The Second-Best Model

In the second-best model, the principal has incomplete information; capabilities are private information of agents. The principal thus can at best randomly match agents. Note: This is usually called adverse selection in economics literature.

The Individual Rationality and Incentive Compatibility constraints are rewritten to reflect the uncertainty of the agents. Let $\Pr[w = w_{*,j}]$ denote agent i's belief that agent j's output is w

$$\Pr[w = w_{*,j}] \stackrel{\text{def}}{=} \sum_{0 \le l \le n} \mu_l \mathbf{1}\{w = w_{l,j}\}. \tag{B}$$

Note: Agent inherits uncertainty from principal, who implements random matching.

Individual rationality.

$$\Pr[w_k = w_*]u(w_k) \ge \underline{u}. \tag{IR2}$$

Incentive compatibility.

$$\Pr[w_k = w_*]u(w_k) \ge \Pr[w^l = w_*]u^l \tag{IC2}$$

where $0 \le l < k$.

Collusion proofness. Here is one more constraint in the second-best model than the first-best model to prevent players collude.

$$\Pr[w_k = w_*]u(w_k) \ge \Pr[w^l = w_*]u^l + \mu_k u^l \tag{CP}$$

where $0 \le l < k$.

Note: For simplicity this paper assumes that only the same type of agents can collude.

Note: Collusion is not the same as cheat defined in the original paper of the ESP Game [11] which is the attempts to fast agree on many images without looking at images. Collusion means some players lower the output quality together, but they still look at images. In other words, when players cheat, the output is incorrect (because they even not look at images). When players collude, the output is still correct, but of poor quality.

As in the first-best model, the principal maximizes its average payoff

$$\max_{p,\{w_0,\ldots,w_n\}} \sum_{0 \le k,l \le n} \mu_k \mu_l \pi(w_k, w_l) \tag{P2}$$

subjected to (IR2), (IC2) and (CP).

The three constraints are divided into two groups: (IR2) and (IC2), and (CP) alone. The best quality outcome is $w_k = w^k$, and so $\Pr[w_k = w_{*,j}] = \mu_k$. Plug them into constraint groups. The first constraint group is solved by

$$\frac{\underline{u}}{\mu_k}. \tag{2}$$

The second constraint group is solved by

$$u^{k-1} + \frac{\mu_{k-1}}{\mu_k} u^{k-1}. \tag{3}$$

The maximization program (P2) is constrained by the maximum of the two

$$u^k = \max\left\{ \frac{\underline{u}}{\mu_k}, u^{k-1} + \frac{\mu_{k-1}}{\mu_k} u^{k-1} \right\}.$$

The constraint groups are not chosen arbitrarily. They correspond to the information rent and collusion-proof rent.

So in the second-best model, the principal implements

- Best quality outcome $w_k = w^k$,
- Using amount of incentives higher than that of the first-best $u^k > \underline{u}$,
- With less than perfect agreement rate

$$\sum_{0 \le k \le n} \mu_k \Pr[w_k = w_*] < 1.$$

Here the components of second-best "cost" are analyzed, including information rent and collusion-proof rent.

Information rent. Observe that in the ESP Game, an agent outputs a best quality label is equivalent to an agent reveals his private information, type. Consider the first-best cost $\underline{u}$; the positive rent $\underline{u}/\mu_k - \underline{u}$ paid by the principal for acquiring agent's private information is called "information rent" in economic literature.

Collusion-proof rent. When μ is non-decreasing, such as when $0 \le l < k \le \lfloor n/2 \rfloor$, we have information rent inversion $\underline{u}/\mu_l > \underline{u}/\mu_k$. Does this mean a good quality label w^k is paid less than a poor quality label w^l? In fact, no. The constraint group (3) implies that $u^k > u^{k-1}$, that is, the principal always has to pay more to a good quality label. We call this rent to maintain (CP) "collusion-proof rent".

How much profit margin is it enough? Now we calculate values of α, β for reference. For a best quality equilibrium to be existed, we must have positive profit margin

$$q^k - p^k > 0$$

where

$$q^k = \alpha k + \beta,$$

and

$$p^k = v\left(\max\left\{ \frac{\underline{u}}{\mu_k}, u^{k-1} + \frac{\mu_{k-1}}{\mu_k} u^{k-1} \right\} \right).$$

Let $n = 4$ and $\underline{v} = \$0.01$, then at least $\alpha \approx \$433.40$, and $\beta \approx \$2.57$. This means given that the agent's reservation utility equals to 1 cent, the qualities of labels must be worthy of tens to thousands of dollars so that a best quality outcome is still profitable after paying information rent and collusion-proof rent (see table 1). For example, the quality of label w^4, $q(w^4)$, must be worthy of at least \$1736.11 dollars to the principal.

On the other hand, the agreement rate is so low (roughly 27.3%) that the expected cost of the principal, the money which he actually pays, is approximately \$12.94 dollars.

This example shows us how expansive and how inefficient (in terms of agreement rate) to implement a best quality equilibrium when the principal does not have capability information.

Example of signals. Here we prepare results for the experiments. We considers two types of signal, "narrow" and

k	μ_k	p^k
0	6.25%	2.56
1	25.00%	4.00
2	37.50%	11.11
3	25.00%	69.44
4	6.25%	1736.11

Table 1: Payoff of labels in unit of dollar.

"lift". Put loosely, the narrow signal is like reducing population variance, and the lift signal is like increasing population mean toward higher type. Note: Here we use the $n = 4, \underline{v} = \$0.01, \alpha = \$1000, \beta = \$10$ setup.

k	μ_k	$\mu_k \vert \theta_{\text{narrow}}$	$\mu_k \vert \theta_{\text{lift}}$
0	6.25%	0.00%	5.25%
1	25.00%	25.00%	25.00%
2	37.50%	50.00%	37.50%
3	25.00%	25.00%	25.00%
4	6.25%	0.00%	7.25%

Table 2: The narrow and lift signal.

Consider a signal θ_{narrow} that shrinks the population by only drawing from type-V_1 to type-V_3 (see the middle column of table 2). The expected payoff to the principal is increased by about \$217 dollars, from \$536.66 dollars to \$753.45 dollars.

Consider a signal θ_{lift} that simply add 1% to μ_4 and subtract 1% from μ_0 (see the rightmost column of table 2). The expected payoff to the principal is increased by about 58 cents, from \$536.66 dollars to \$537.24 dollars.

3.4 Summary

The principal is facing an adverse selection problem when the capability information is private, and have to pay information rent and collusion-proof rent for a good quality outcome. In addition to these rents, the principal suffers from lower agreement rate, that is, slower verification speed. We can perceive these troubles when the principal lacks the capability information as the "cost" of the random matching.

4. EXPERIMENT

A preliminary, small-scale experiment was conducted to test the core concepts of the theory. The experiment also demonstrated how the Segmentation method with narrow and lift signal can be implemented use online communities.

The Segmentation method extracts capability information from demographic data. The border of a demographic group is very flexible, which could be as broad as a university, or as tight as a zealous fan group. The CAM does not necessarily have to ask players to fill in annoying survey forms; the demographic data can be automatically crawled from social network websites or online forums,

Note: There is another implementation of the CAM, called the Bootstrapping method, detailed in [2].

4.1 Experiment Design

On choosing demographic data, the lessons will be learned from the experiments are:

- The demographic group should be related to the content of the images, and
- The deeper the participation of an agent in this group, the higher the capability he might have.

The experiment design featured:

- The one-shot ESP Game was played without any time limit.
- Subjects were not rewarded by scores or any other incentives.
- Problem sets of images that were assumed to be associated with signals were chosen.
- Subjects were actually played with robots (for reasons stated below).
- The control and treatment group differed in:
  - The matching scheme.
  - The equilibrium strategy played by robots.
- The experiment only had between-subjects effect, but no within-subjects effect (because each subject participated only once).

In brief, the experiment design features: the one-shot ESP Game, robots, and the Segmentation.

The detailed experimental process was:

1. Subjects were randomly put into either the control or treatment group.
2. Subjects were asked to report their participation level of online communities.
3. Subjects were informed the matching scheme (but actually played with a robot).

   Control: Random matching.
   Treatment: The CAM.
4. Subjects played 5 training images from each problem set, in the same order. The robot's output label was displayed when subjects lost.
5. Subjects played 20 testing images, every four from each problem set, in the same order. The robot's output label was *not* displayed when subjects lost.
6. Subjects filled in a post-hoc survey to assess the difficulty of all problem set in absolute and relative scale.

Note: In training games and test games, no scores or any other incentives were awarded when subjects won, and there was no time limit.

4.2 Comments on the Experiment Design

To eliminate the effect of time preference, the one-shot ESP Game, rather than the original ESP Game, was used in the experiment. The time preference should be eliminated because it has been shown to affect the output quality [6].

In addition to time preference, anything that might affect subjects was eliminated, such as time limit and scores, so that any difference in outcomes could only be explained by matching schemes.

The use of robots, a necessary evil in small-scale experiment, was because:

- It was unlikely to have aligned-capability subjects at the same time, especially when the scale was small.
- To eliminate human variations as much as possible.

The robots played equilibrium strategy, that is, pooling when put into the control group, and separating when put into the treatment group.

Only high type of robot was implemented. Otherwise, one more factor (robot's type) had to be added, along with matching scheme. This would further divide subjects, resulting in smaller groups that could not yield anything statistically significant.

In testing games, the robot's labels were not displayed to subjects when subjects lost as the original ESP Game, but in testing games, in order to teach subjects who had had never played the ESP Game, the robot's labels were displayed.

The implemented one-shot ESP Game did not compare labels literally. Instead, a list of synonyms and common misspellings was built in for label comparison.

4.3 Signals and Problem Sets

Two narrow-and-lift signals were chosen. One was a subject's participation level in a online community, and the other was locality (the college where subjects were recruited). Note: Although the experiment extracted these signals by asking subjects, it was easy to crawl these signals automatically.

For the online community signals, it was assumed that the population was narrower when the community was smaller, and the population was lifted higher when the participation was deeper (assuming that participation level was positively correlated to capability).

The online communities were Bulletin Board System (BBS) boards:

- WoW / Exchange information of the World of Warcraft.
- Baseball / Discussions about baseball.
- OnePiece / Discussions about the manga One Piece.

For each online community, the participation of a subject was categorized into 4 levels.

Level 0. None of the below.

Level 1. Had played the World of Warcraft, watched any of Major League Baseball games, or read the One Piece, respectively.

Level 2. Had read the respective BBS board.

Level 3. Had added the respective BBS board to his My Favorite.

Three problem sets positively associated with online community signals were chosen, namely, WoW, MLB, and OP. The MLB problem set were pictures of game characters of the World of Warcraft. The MLB problem set were pictures of Major League Baseball players. The OP problem set were pictures of manga characters of the One Piece.

Two problem sets (positively and negatively) associated with locality signal were chosen, namely, LO and FO. The LO and FO problem set were images of local and foreign celebrities and landmarks, respectively. It was assumed that subjects were more capable to the LO problem set than FO; this assumption would be verified.

The images of a problem set were carefully chosen that their difficulty to subjects was assumed uniform, and so variation of output quality within one problem set by one subject was assumed random normal.

4.4 Subjects

In total, 26 subjects were recruited from National Taiwan University (that means 104 labels per problem set). Table 3 shows the distribution of subject's gender, age, and group.

		Distribution
Gender	Female	10
	Male	16
Age	18–21	8
	22–25	14
	26–29	2
	30–33	2
Group	Control	12
	Treatment	14

Table 3: The distribution of subject's gender, age, and group.

Table 4 shows the distribution of participation level. The OnePiece board had the most dedicated subjects (level 2 and level 3).

To our surprise, the WoW board was "very unpopular" among our subjects; 24 out of 26 subjects had had never played the World of Warcraft. In fact, the "unpopularity" of the World of Warcraft among subjects would cause the regression to fail because only zero or one subject was in levels above 1.

Participation Level	BBS Board		
	WoW	Baseball	OnePiece
#0	24	17	7
#1	1	6	11
#2	0	2	4
#3	1	1	4

Table 4: The participation levels of online communities.

Table 5 shows the post-hoc survey result. The survey asked subjects to evaluate the difficulty of each problem set in absolute and relative (to other problem sets) scale.

No matter sorted by mean or median, the difficulty of problem sets were: LO (easiest), OP, FO, MLB, and WoW (hardest).

The fact that subjects felt LO was easier than FO verified our assumption that locality was a good measure of capability to the LO and FO problem set.

		WoW	MLB	OP	LO	FO
Absolute	Median	5.00	5.00	3.00	2.00	4.00
	Mean	4.70	4.35	2.96	2.61	3.91
Relative	Median	5.00	4.00	2.00	2.00	3.00
	Mean	4.48	3.70	2.13	1.70	3.00

Table 5: The absolute and relative difficulty of problem sets. The difficulty scales from 1 (easiest) to 5 (hardest).

Table 5 also helped us verify that participation levels were indeed, as assumed to be, good measures of capability. Why was that? The participation level was an objective measure of capability, whereas the post-hoc survey was a subjective assessment of difficulty. Although different by nature, they demonstrated the same tendency: OP (most capable or easiest), MLB, and WoW (least capable or hardest) no matter

sorted by participation level or by subjective difficulty assessment.

4.5 Experiment Results

For each problem set, 104 labels were collected, and manually annotated. A label was annotated "of good quality" if it was the name (including synonyms and common misspellings) of a person, object, or building in the image, *i.e.,* correct and specific.

Table 6 shows numbers of label annotated as of good quality, divided by numbers of label of that category. Note: There were empty categories in the WoW problem set due to the "unpopularity" of the World of Warcraft among subjects.

A first observation was the trend that the ratio of good quality labels increased when the "Group" or "Participation Level" increased.

Problem Set	Group	Participation Level			
		#0	#1	#2	#3
WoW	0	0/44	0/ 4	0/ 0	0/ 0
	1	2/52	0/ 0	0/ 0	4/ 4
MLB	0	6/28	4/12	0/ 4	2/ 4
	1	12/36	7/12	4/ 4	1/ 4
OP	0	0/12	11/28	0/ 4	0/ 4
	1	4/16	9/16	10/12	11/12
LO	0	30/48			
	1	44/56			
FO	0	1/48			
	1	10/56			

Table 6: The contingency table of output labels. In the "Group" column, 0 is for the control and 1 for the treatment. Note: The LO and FO problem set did not have related participation levels.

The ratios of good quality labels were regressed against matching scheme and participation level in a logit model. Let i index over problem sets { WoW, MLB, OP, LO, FO }. Let $\Pr[Y_i = 1]$ denote the ratio of good quality labels, and X_i the group (0 is for the control and 1 for the treatment), and Z_i the participation level. The logit model was

$$\text{logit}\Pr[Y_i = 1] = \beta_{i0} + \beta_{i1}X_i + \text{DUMMY}_i\beta_{i2}Z_i \quad (4)$$

where DUMMY_i was a dummy variable that equaled to 1 when i equaled to WoW, MLB, or OP; and 0 otherwise.

Table 7 shows the p-values of logit regressions. All were statistically significant at least at the 0.05 significance level; the null hypotheses $\beta_{i1} = \beta_{i2} = 0$ were rejected. That is, matching schemes and capabilities (X_i, Z_i) indeed affected the good quality ratios $\Pr[Y_i = 1]$.

	p-value
WoW	0.0000***
MLB	0.0060**
OP	0.0000***
LO	0.0434*
FO	0.0027**

Table 7: The p-values of logit regressions. Significance codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.**

Table 8 shows the predictive power of the logit models; the rightmost column are p-values (β-values would be explained later). For the MLB, OP, and FO problem set, matching scheme X_i and capability Z_i were statistically significant predictors.

Note: The WoW and LO problem set presented interesting results. See paragraphs below.

		Estimate	Std. Error	z value	Pr[> |z|]
WoW	β_{i0}	-26.6047	4380.2597	-0.01	0.9952
	β_{i1}	23.3858	4380.2598	0.01	0.9957
	β_{i2}	7.7057	2116.2887	0.00	0.9971
MLB	β_{i0}	-1.5753	0.4201	-3.75	0.0002***
	β_{i1}	1.0668	0.4638	2.30	0.0215*
	β_{i2}	0.6083	0.2714	2.24	0.0250*
OP	β_{i0}	-2.0770	0.4621	-4.50	0.0000***
	β_{i1}	1.5524	0.4643	3.34	0.0008***
	β_{i2}	0.7692	0.2393	3.22	0.0013**
LO	β_{i0}	0.5108	0.2981	1.71	0.0866•
	β_{i1}	0.7885	0.4415	1.79	0.0741•
FO	β_{i0}	-3.8501	1.0105	-3.81	0.0001***
	β_{i1}	2.3241	1.0691	2.17	0.0297*

Table 8: The summary of logit regression results. Significance codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.**

WoW. Although the logit regression failed partially due to the "unpopularity" effect (empty categories), the main reason was that the WoW problem set was the hardest. In fact, when the ratios were regressed against only capability Z_i,

$$\text{logit}\Pr[Y_{\text{WoW}} = 1] = \hat{\beta}_0 + \hat{\beta}_2 Z_{\text{WoW}},$$

the p-value was statistically significant, and capability was statistically significant predictor (table 9). In other words, the WoW problem set was so hard that the capability itself dominated the outcomes, and so the matching scheme had little effect on output quality.

		Estimate	Std. Error	z value	Pr[> |z|]
WoW	$\hat{\beta}_0$	-4.0772	0.7562	-5.39	0.0000***
	$\hat{\beta}_2$	2.3410	0.7657	3.06	0.0022**

Table 9: The logit regression on the WoW problem set with only capability. Significance codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.**

LO. On the contrary to the WoW problem set, the problem of the LO problem set was too easy. The control-LO category had good quality labels that were 30 times more than the control-FO category (table 6). Given that control group subjects should output poor quality labels (and they did in all problem sets except LO), the huge disparity between the control-LO and control-FO category may be explained by that subjects just could not think of any poor quality label; the LO problem set was just too easy.

4.6 Predicted Good Quality Ratios

Table 10 shows the predicted ratios $\Pr[Y_i = 1]$ from the fitted β-values. Note: The predicted ratios of the WoW and LO problem set were not predicted by statistically significant predictors, and were listed only for reference.

Consistent trends in the MLB, OP, and FO (and also WoW and LO) emerged:

- The ratios were higher when the participation levels were higher.

- The ratios of the treatment group (the CAM was used) were higher than the ratios of the control group (the random matching was used).
- The effect of matching scheme was greater than the effect of capability (participation level). Lower participation level treatment categories had higher ratios than higher participation level control categories.

The last trend was particularly interesting: A less-capable but properly-motivated player could output better quality than a more-capable but less-motivated player.

	Group	Participation Level			
		#0	#1	#2	#3
WoW	0	0.0000	0.0000	0.0000	0.0297
	1	0.0385	0.9889	1.0000	1.0000
MLB	0	0.1715	0.2755	0.4113	0.5621
	1	0.3755	0.5249	0.6700	0.7886
OP	0	0.1114	0.2129	0.3685	0.5574
	1	0.3718	0.5608	0.7338	0.8561
LO	0	0.6250			
	1	0.7857			
FO	0	0.0208			
	1	0.1786			

Table 10: The predicted ratios of good quality labels. In the "Group" column, 0 is for the control and 1 for the treatment.

4.7 Discussion

We had had observed:

- Potentially, a more capable player was more likely to generate good quality labels.
- The CAM had improved quality of labels, given that players had moderate capability.
- The effect of matching scheme on output quality was greater than the effect of capability, for tasks that players had moderate capability.

From the observations, a limitation of the CAM was: When difficulty of a task was extremely high or low, the capability of players dominated the output quality, and the effect of the CAM was negligible.

This limitation pointed out that matching the right task to the right player was as important as matching the right pair of players.

The experiment *per se* brought its own limitation. Subjects were interacted with robots, not other subjects, and there was only one type of robots. This experiment design restricted what we could conclude from data. The experiment was more like testing if subjects would learn and play the equilibrium strategy, and less like a user study of the CAM. Despite the methodological imperfectness, the promising results of this preliminary experiment showed that the CAM is worthy of further investigation in larger-scale experiments.

5. CONCLUSION

This paper proposes the capability-aligned matching (CAM) for solving the poor quality problem that the output-agreement/ simultaneous-verification Games with a Purpose (GWAP) would suffer from.

The analysis of an adverse selection model shows that the CAM has two advantages over random matching. On cost aspect, the information and collusion-proof rent, which are used for increasing output quality, are reduced. On informational aspect, the agreement rate, which is the bounding factor of verification speed, is increased.

This paper implements the Segmentation method, whose source of capability information is demographic data, and tests it in the experiments. The experiments suggest that task-human matching is as important as human-human matching.

All in all, the CAM is orthogonal to game rules, and so could be seamlessly integrated into existing and future output-agreement/simultaneous-verification GWAP for improving output quality.

6. REFERENCES

[1] M. Bernstein, D. Tan, G. Smith, M. Czerwinski, and E. Horvitz. Collabio: a game for annotating people within social networks. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 97–100. ACM, 2009.

[2] C.-L. Chiou. Capability-aligned matching: Improving quality of games of a purpose. Master's thesis, National Taiwan University, January 2011.

[3] C.-J. Ho, T.-H. Chang, and J. Y.-J. Hsu. Photoslap: A multi-player online game for semantic annotation. In *In Twenty-Second Conf. on Artificial Intelligence*, 2007.

[4] C.-J. Ho, T.-H. Chang, J.-C. Lee, J. Y.-J. Hsu, and K.-T. Chen. Kisskissban: A competitive human computation game for image annotation. In *Proceedings of the First Human Computation Workshop (HCOMP 2009)*, June 2009.

[5] C.-J. Ho and K.-T. Chen. On formal models for social verification. In *Proceedings of the First Human Computation Workshop (HCOMP 2009)*, June 2009.

[6] S. Jain and D. C. Parkes. A game theoretic analysis of games with a purpose. In *In Proc. 4th Intl. Workshop on Internet and Network Economics*, 2008.

[7] Y.-L. Kuo, K.-Y. Chiang, C.-W. Chan, J.-C. Lee, R. Wang, E. Y.-T. Shen, and J. Y.-J. Hsu. Community-based game design: Experiments on social games for commonsense data collection. In *Proceedings of the First Human Computation Workshop (HCOMP 2009)*, June 2009.

[8] E. Law, L. von Ahn, and T. Mitchell. Search war: A game for improving web search. In *Proceedings of the First Human Computation Workshop (HCOMP 2009)*, June 2009.

[9] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". In *Proceedings of the First Human Computation Workshop (HCOMP 2009)*, June 2009.

[10] S. Robertson, M. Vojnovic, and I. Weber. Rethinking the ESP game. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 3937–3942, 2009.

[11] L. von Ahn and L. Dabbish. Labeling images with a computer games. In *In Proc. SIGCHI Conf. on Human Factors in Computing Systems*, 2004.

[12] L. von Ahn and L. Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

False-name-proof Mechanism Design without Money

Taiki Todo, Atsushi Iwasaki, and Makoto Yokoo
Department of Informatics, Kyushu University
Motooka 744, Fukuoka, Japan
{todo@agent., iwasaki@, yokoo@}is.kyushu-u.ac.jp

ABSTRACT

Mechanism design studies how to design mechanisms that result in good outcomes even when agents strategically report their preferences. In traditional settings, it is assumed that a mechanism can enforce payments to give an incentive for agents to act honestly. However, in many Internet application domains, introducing monetary transfers is impossible or undesirable. Also, in such highly anonymous settings as the Internet, declaring preferences dishonestly is not the only way to manipulate the mechanism. Often, it is possible for an agent to pretend to be multiple agents and submit multiple reports under different identifiers, e.g., by creating different e-mail addresses. The effect of such false-name manipulations can be more serious in a mechanism without monetary transfers, since submitting multiple reports would have no risk.

In this paper, we present a case study in *false-name-proof mechanism design without money*. In our basic setting, agents are located on a real line, and the mechanism must select the location of a public facility; the cost of an agent is its distance to the facility. This setting is called the *facility location problem* and can represent various situations where an agent's preference is *single-peaked*. First, we fully characterize the deterministic false-name-proof facility location mechanisms in this basic setting. By utilizing this characterization, we show the tight bounds of the approximation ratios for two objective functions: social cost and maximum cost. We then extend the results in two natural directions: a domain where a mechanism can be randomized and a domain where agents are located in a tree. Furthermore, we clarify the connections between false-name-proofness and other related properties.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics, Theory

Cite as: False-name-proof Mechanism Design without Money, Taiki Todo, Atsushi Iwasaki, and Makoto Yokoo, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 651-658.

Keywords

Auction and mechanism design, social choice theory, facility location problems

1. INTRODUCTION

1.1 Background

Mechanism design has become an integral part of electronic commerce and a promising field for applying AI and agent technologies. In particular, the celebrated Vickrey-Clarke-Groves (VCG) mechanism for combinatorial auctions, which is considered one crucial contribution of mechanism design, has been applied to several domains. One of its advantages is that it satisfies a property called *strategy-proofness*; no agent ever benefits from misreporting her preference, regardless of the other agents' strategies. The VCG mechanism achieves this property by collecting an appropriate amount of payment from each winner of the auction.

In several domains such as the Internet, however, implementing payments is sometimes impossible mainly due to security/banking issues. Moreover, there are several application fields in which monetary transfers should not be introduced due to ethical/legal considerations, including political decision making or kidney exchanges. Thus, mechanisms must be developed that satisfy strategy-proofness without involving monetary transfers. Such *mechanism design without money* is quite challenging and has attracted considerable attention among computer scientists (see [5, 10]).

Meanwhile, in such highly anonymous settings as the Internet, reporting preference insincerely is not the only way to manipulate a mechanism. Often, it is possible for an agent to pretend to be multiple agents and participate in a mechanism multiple times by using different identifiers, e.g., by creating different e-mail accounts. Since many Web applications require a valid e-mail address only, an agent can create multiple e-mail address at practically no cost. Such strategic behaviors called *false-name manipulations* have been discussed so far in the mechanism design field.

In environments in which payments can be made securely, authenticating each identifier and collecting a participation fee might discourage agents from using multiple identifiers. Furthermore, in mechanisms with monetary transfers, adding false identifiers is risky. For example, in an auction, the manipulator might have to pay a lot of money or buy unnecessary items by such false-name manipulation.

In contrast, such manipulations are more likely to occur in a mechanism without monetary transfers, since submitting multiple reports is less risky. For example, in voting,

casting additional votes is unlikely to create disadvantages for the manipulator. To the best of our knowledge, there exist very few works on false-name manipulations in mechanism design without money. One notable exception is a work by Conitzer [4], which characterized anonymity-proof voting rules (i.e., rules that satisfy false-name-proofness and voluntary participation). The obtained result is rather negative. In essence, an anonymity-proof voting rule can take into account voters' preferences only when voters unanimously prefers one candidate within two candidates that are chosen at random. Furthermore, even if we only require strategy-proofness, the Gibbard-Satterthwaite theorem states that it is impossible to make a mechanism strategy-proof when agents' preferences are general (see [7]).

1.2 Our Results

In this paper, we present a case study in false-name-proof mechanism design without money. Assuming that agents' preferences are highly structured, we avoid falling into the negative result in Conitzer [4], or in more general sense, the Gibbard-Satterthwaite Theorem. We focus on *facility location problems* and discuss how difficult it is to incentivize agents to behave sincerely, even though they can use false identifiers. This is the first work to deal with false-name manipulations in facility location problems.

We discuss a facility location problem on a real line as a basic setting and characterize deterministic false-name-proof facility location mechanisms. Our characterization is inspired by Moulin's characterization of strategy-proofness [6]. To simplify expositions and notations, we define the cost of an agent as the distance between her location and a facility. It is straightforward to extend this characterization to a domain with general single-peaked preferences. Additionally, we establish the tight bounds of the approximation ratios achieved by deterministic false-name-proof mechanisms for two objective functions: social cost and maximum cost.

We then extend the results of the basic case in two further directions. One is a domain of randomized mechanisms, and the other is a facility location problem on a tree. For randomized mechanisms, we show in Section 4.1 that the left-right-middle mechanism, which was originally proposed in Procaccia and Tennenholtz [8], satisfies false-name-proofness. Furthermore, we show a lower bound of the approximation ratio for the social cost. On the other hand, for the facility location problem on a tree, we characterize deterministic false-name-proof mechanisms in Section 4.2. Our characterization can be considered a refinement of the result by Schummer and Vohra [9], in which they characterized deterministic strategy-proof mechanisms on a tree.

Furthermore, in Section 5, we clarified the connections between false-name-proofness and other related properties in a facility location problem on a tree. We focused on *population monotonicity*, *group-strategyproofness*, and *anonymity-proofness*, which have been discussed in the literature of social choice and mechanism design. By utilizing our characterization, we show that both population monotonicity and anonymity-proofness are equivalent to false-name-proofness. We also show that there exists a group-strategyproof mechanism which is not false-name-proof.

1.3 Related Works

Facility location problems have also been considered an important famework of social choice due to the highly structured preferences of agents in the setting: *single-peaked preferences*. There exist many application domains with such single-peaked preferences. For example, in political decision making, an agent's peak is her most preferred alternative. Moulin [6] characterized strategy-proof, Pareto efficient, and anonymous facility location mechanisms on a real line. Schummer and Vohra [9] extended Moulin's results to facility location problems on graphs.

Procaccia and Tennenholtz [8] presented a case study in approximate mechanism design without money and established tight bounds for the approximation ratio achieved by strategy-proof facility location mechanisms on a real line. They also proposed two extensions of facility location problems: a domain where two facilities must be located and a domain where each agent owns multiple locations. Alon et al. [1] discussed the maximum cost of strategy-proof facility location mechanisms on several network topologies. Guo and Conitzer [5] is one of the most recent development of approximate mechanism design without money for strategy-proof resource allocations.

False-name manipulations have also been widely studied in combinatorial auctions. Yokoo et al. [12] proposed a condition where VCG becomes false-name-proof. Todo et al. [11] characterized false-name-proof combinatorial auction mechanisms. Besides combinatorial auctions, false-name-proofness and its relatives have been discussed in other mechanism design fields, such as voting [4] and coalitional games [2]. In particular, Conitzer [4] proposed an extended property called *anonymity-proofness* in voting and characterized anonymity-proof voting rules.

2. PRELIMINARIES

2.1 Basic Model

In this paper, we deal with facility location problems in which a mechanism locates one facility. Let n denote the number of agents (identifiers) joining a mechanism and N ($|N| = n$) the set of agents. Note that the number of agents n is defined to be variable in $\mathbb{N}$ to discuss the change of the number of agents joining a mechanism. Each agent $i \in N$ has a true location (or the most preferred location) x_i on a graph G. In this paper, we restrict our attention to *peak-only* mechanisms, i.e., each agent reports only her most preferred location. In a more general setting (e.g., voting), this can be a quite strong restriction. However, in our setting, we can assume any strategy-proof mechanism is peak-only, since the peak location of each agent is her only private information. The cost of an agent i is defined by the distance $d(\cdot, \cdot)$ between her true location and the location of a facility: if the facility is located at y, the cost of agent i with location x_i is $\mathrm{cost}(x_i, y) = d(x_i, y)$. If a graph G is a real line, the distance is defined as $|x_i - y|$.

A (direct revelation, deterministic) *facility location mechanism* (or simply *mechanism*) is a function that maps a reported location profile $x = (x_1, \ldots, x_n)$ by the set of agents to a location of a facility y on a graph G. A mechanism must locate a facility with respect to any number of agents n, since we consider an environment where each agent may use multiple identifiers (formally defined in Section 2.2). For this reason, we define a mechanism f as a set of functions $(f^n)_{n \in \mathbb{N}}$, where each function f^n is a mapping from a set of location profiles reported by n identifiers to the graph. For simplicity, we assume that a mechanism is *anonymous*,

meaning that the obtained results are invariant under the permutation of identifiers.

Definition 1 (Facility Location Mechanism). *For any natural number $n \in \mathbb{N}$, a* facility location mechanism f *assigns an outcome $f^n(x)$ to any reported location profile $x = (x_1, \ldots, x_n) \in G^n$:*

$$f = (f^n)_{n\in\mathbb{N}}, f^n : G^n \to G.$$

In facility location problems, each agent reports her location x_i', which is not necessarily her true location x_i, to the mechanism. However, in a *strategy-proof* mechanism, it is guaranteed that each agent reports her true location x_i to the mechanism if she behaves to minimize her cost.

Definition 2 (Strategy-proofness). *A mechanism f is* strategy-proof *if $\forall n \in \mathbb{N}$, $\forall i \in N$, $\forall x_{-i}$, $\forall x_i$, $\forall x_i'$, $\mathrm{cost}(x_i, f(x_i, x_{-i})) \leq \mathrm{cost}(x_i, f(x_i', x_{-i}))$.*

Here x_{-i} denotes the reported location profile by agents except i. That is, $f(x_i', x_{-i})$ is the location of a facility when agent i reports x_i' and other agents report x_{-i}. Definition 2 means that a mechanism is strategy-proof if for each agent, reporting her true location is a dominant strategy; it minimizes her cost regardless of the strategies of other agents.

Several strategy-proof mechanisms have been developed for facility location problems. For a real line, one well-known strategy-proof mechanism is the *median* mechanism, which chooses the median location among the reported locations (if the number of agents n is even, locates at the $n/2$-th smallest location). To simplify the exposition and the notations, we define a function $\mathrm{med}(\cdot)$ that returns the median point for a given profile of real numbers.

For the facility location problem on a real line, Moulin [6] characterized strategy-proof mechanisms.

Theorem 1 (Moulin, 1980). *A mechanism f is strategy-proof, Pareto efficient, and anonymous if and only if for all $n \in \mathbb{N}$, there exist $n-1$ real numbers $\alpha_1^n, \alpha_2^n, \ldots, \alpha_{n-1}^n$ such that for all reported location profile $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$,*

$$f(x) = \mathrm{med}(x_1, \ldots, x_n, \alpha_1^n, \ldots, \alpha_{n-1}^n). \quad (1)$$

In the case of a real line, *Pareto efficiency* requires that a facility be located at a point between the leftmost and rightmost locations among the reported locations. Theorem 1 means that any Pareto efficient, anonymous, and strategy-proof mechanism can be represented by appropriately setting the parameters in Eq. (1). Indeed, the median mechanism is represented by setting these parameters as follows:

$$\forall n \in \mathbb{N}, \forall m \in \{1, \ldots, n-1\}, \alpha_m^n = \begin{cases} -\infty & \text{if } m \text{ is odd} \\ \infty & \text{if } m \text{ is even.} \end{cases}$$

Also, the leftmost mechanism, which locates a facility at the smallest location among the reported locations, is represented by setting all parameters to $-\infty$.

We focus on a worst case analysis to consider the performance of the mechanisms. This analysis is commonly used in the literature of (algorithmic) mechanism design, especially by computer scientists. We introduce two objective functions: *social cost* and *maximum cost*. The social cost is the sum of the costs of all agents. A solution minimizing the social cost is also called a *minisum* solution. On the other hand, the maximum cost is defined by the cost of the agent whose cost is the highest among all agents. A solution minimizing the maximum cost is also called a *minimax* solution, which achieves an equitable location. We now define the *approximation ratios* of a mechanism.

Definition 3 (Approximation Ratio). *The* approximation ratios *of a mechanism f for the social cost and the maximum cost are defined as follows:*

$$\max_x \frac{\sum_{i\in N} \mathrm{cost}(x_i, f(x))}{\min_{y\in G} \sum_{i\in N} \mathrm{cost}(x_i, y)},$$

$$\max_x \frac{\max_{i\in N} \mathrm{cost}(x_i, f(x))}{\min_{y\in G} \max_{i\in N} \mathrm{cost}(x_i, y)}.$$

2.2 False-name-proofness

In this subsection, we formalize false-name-proofness in facility location problems. First, we introduce some notations for discussing false-name manipulations.

Let ϕ_i denote the set of identifiers used by agent i. This is also the private information of agent i. Let x_{ϕ_i} denote a location profile reported by a set of identifiers ϕ_i, and let $x_{-\phi_i}$ denote a location profile reported by identifiers except for ϕ_i. In this definition, x_{ϕ_i} is considered a false-name manipulation by i.

Definition 4 (False-name-proofness). *A mechanism f is* false-name-proof *if $\forall n \in \mathbb{N}$, $\forall i \in N$, $\forall x_{-\phi_i}$, $\forall x_i$, $\forall \phi_i$, $\forall x_{\phi_i}$, $\mathrm{cost}(x_i, f(x_i, x_{-\phi_i})) \leq \mathrm{cost}(x_i, f(x_{\phi_i}, x_{-\phi_i}))$.*

In other words, a mechanism is false-name-proof if for each agent, reporting her true location by using a single identifier is a dominant strategy, even though she can use multiple identifiers. The following example shows that the median mechanism on a real line is not false-name-proof: an agent can reduce her cost by using multiple identifiers.

Example 1. Consider the median mechanism on a real line and $N = \{1, 2, 3\}$. Assume that $x_1 = 1$, $x_2 = 2$, and $x_3 = 3$. If they report their locations truthfully, the mechanism locates a facility at 2. However, if agent 1 adds two false identifiers and reports $x_{\phi_1} = (1, 1, 1)$, the mechanism locates a facility at 1. By this false-name manipulation, agent 1 can strictly reduce her cost.

3. BASIC RESULTS

3.1 Characterization Theorem

Now we are ready to show our characterization theorem of false-name-proof mechanisms on a real line. More precisely, we provide a necessary and sufficient condition for a mechanism to be false-name-proof, Pareto efficient, and anonymous. Lemmas 1 and 2 prove the theorem.

Theorem 2. *A mechanism f is false-name-proof, Pareto efficient, and anonymous if and only if there exists a real number α such that for all $n \in \mathbb{N}$ and for all reported location profiles $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$,*

$$f(x) = \mathrm{med}(x_1, \ldots, x_n, \underbrace{\alpha, \ldots, \alpha}_{n-1}). \quad (2)$$

Lemma 1. *If a mechanism f satisfies Eq. (2), f is false-name-proof, Pareto efficient, and anonymous.*

PROOF. If f satisfies Eq. (2), it also satisfies Eq. (1). Thus, f is Pareto efficient and anonymous. Therefore, we now show that f is false-name-proof if it satisfies Eq. (2).

Let us discuss false-name manipulations by agent i and show that no false-name manipulation reduces her cost. Let $\mathrm{lt}(x_{-i})$ denote the leftmost location in a location profile x_{-i} reported by agents except i, and let $\mathrm{rt}(x_{-i})$ denote the rightmost location. If $\mathrm{lt}(x_{-i}) \leq \alpha \leq \mathrm{rt}(x_{-i})$ holds, f always locates a facility at α regardless of i's strategy.

We prove that f is false-name-proof if $\alpha < \mathrm{lt}(x_{-i})$. The same argument can be applied if $\mathrm{rt}(x_{-i}) < \alpha$ from the symmetry. We show that agent i cannot reduce her cost by false-name manipulations in each of the following three cases: (i) $x_i \leq \alpha$, (ii) $\alpha < x_i \leq \mathrm{lt}(x_{-i})$, and (iii) $\mathrm{lt}(x_{-i}) < x_i$.

Case (i) If i's true location x_i satisfies $x_i \leq \alpha$, f locates a facility at α if i reports truthfully. In this situation, reporting $x_{i'} > \alpha$ by all false identifiers $i' \in \phi_i$ are the only false-name manipulations that affect the outcome. However, by these manipulations, the outcome becomes strictly further away from x_i.

Case (ii) If x_i satisfies $\alpha < x_i \leq \mathrm{lt}(x_{-i})$, f locates a facility at x_i when agent i reports her true location. In this case, agent i has no incentive to use false identifiers.

Case (iii) If x_i satisfies $\mathrm{lt}(x_{-i}) < x_i$, f locates a facility at $\mathrm{lt}(x_{-i})$ if i reports truthfully. In this situation, reporting $x_{i'} < \mathrm{lt}(x_{-i})$ by all false identifiers $i' \in \phi_i$ are the only false-name manipulations that affect the outcome. However, by these manipulations, the outcome moves further away from x_i. □

LEMMA 2. *If a mechanism f is false-name-proof, Pareto efficient, and anonymous, f satisfies Eq.* (2).

PROOF. Since false-name-proofness is a generalization of strategy-proofness, if f is false-name-proof, Pareto efficient, and anonymous, then for all $n \in \mathbb{N}$, f has $n-1$ parameters $\alpha_1^n, \alpha_2^n, \ldots, \alpha_{n-1}^n$ satisfying $\alpha_1^n \leq \ldots \leq \alpha_{n-1}^n$ and locates a facility at the median point defined by Eq. (1) (from Theorem 1). To prove this lemma, it suffices to show that there exists $\alpha \in \mathbb{R}$ such that for all $n \geq 2$,

$$\alpha_1^n = \ldots = \alpha_{n-1}^n = \alpha. \tag{3}$$

We prove this lemma by induction on n. For $n = 2$, Eq. (3) obviously holds since there exists only one parameter α_1^2.

We suppose that Eq. (3) holds for all $n \leq k$ and show that it also holds for $n = k+1$. Assuming $\alpha_1^k = \ldots = \alpha_{k-1}^k = \alpha$ holds, we prove $\alpha_1^{k+1} = \ldots = \alpha_k^{k+1} = \alpha$ also holds.

First, assume that $\alpha < \alpha_1^{k+1}$ holds and derive a contradiction. Now consider that a location profile $x = (x_1, \ldots, x_k)$ such that $\alpha < x_1 < \ldots < x_k < \alpha_1^{k+1}$ holds. In this case, the outcome of the mechanism f is $f(x) = x_1$. If an agent k whose location is the largest among all k agents adds another identifier k' and reports $x_{\phi_k} = (x_k, x_k)$, then the outcome changes to $f(x_{\phi_k}, x_{-k}) = x_k$. By this manipulation, k's cost decreases from $x_k - x_1$ to 0. This contradicts the assumption of false-name-proofness. From symmetry, the same argument can be applied to $\alpha_k^{k+1} < \alpha$.

Next, assume that there exists $j \in \{1, \ldots, k-1\}$ such that $\alpha_j^{k+1} \leq \alpha < \alpha_{j+1}^{k+1}$ holds and derive a contradiction. Consider a location profile $x = (x_1, \ldots, x_k)$ such that $\alpha < x_1 < \ldots < x_k < \alpha_{j+1}^{k+1}$. In this case, we have $f(x) = x_1$. If agent $l = k - j$ (whose location is the $(k-j)$-th smallest among the k agents) reports $x_{\phi_l} = (x_l, x_l)$, the outcome becomes $f(x_{\phi_l}, x_{-l}) = x_l$. By this manipulation, l's cost decreases from $x_l - x_1$ to 0. This contradicts false-name-proofness. From symmetry, we can apply the same argument to $\alpha_j^{k+1} < \alpha \leq \alpha_{j+1}^{k+1}$. □

Theorem 2 means that f is false-name-proof if and only if it has a fixed parameter α regardless of the number of agents and locates a facility based on the following rule. Given a reported location profile x, f locates a facility at α if α is between the smallest and largest locations among the reported locations; otherwise, it locates at the closest location to α among the reported locations.

Since we can obtain the leftmost and the rightmost mechanism by setting the parameter α to $-\infty$ and ∞, respectively, both mechanisms satisfy false-name-proofness. However, since the median mechanism cannot be represented in this form, it does not satisfy false-name-proofness. In this way, Theorem 2 allows us to easily verify if a mechanism satisfies false-name-proofness.

One might think that a mechanism, which always locates a facility at a pre-defined point regardless of the agents' reports, satisfies false-name-proofness. This is true; there is no incentive for agents to participate at all. Even though it is false-name-proof, we cannot represent it in the form of Theorem 2 because it is not Pareto efficient. However, it is straightforward to obtain the following corollary that can deal with such non-efficient mechanisms.

COROLLARY 1. *A mechanism f is false-name-proof and anonymous if and only if there exist three real numbers $\alpha_L, \alpha, \alpha_R (\alpha_L \leq \alpha \leq \alpha_R)$ such that for all $n \in \mathbb{N}$ and for all reported location profiles $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$,*

$$f(x) = \mathrm{med}(x_1, \ldots, x_n, \alpha_L, \overbrace{\alpha, \ldots, \alpha}^{n-1}, \alpha_R). \tag{4}$$

The additional parameters α_L and α_R define the range of the mechanism; the mechanism described in Corollary 1 always locates a facility in the range $[\alpha_L, \alpha_R]$. Indeed, Eq. (4) can describe the above mechanism by defining the parameters as $\alpha_L = \alpha_R = \alpha$. Clearly, if we set these two parameters as $-\infty$ and ∞, respectively, we obtain Theorem 2.

Procaccia and Tennenholtz [8] extended the facility location problem on a real line to a domain where each agent i owns ω_i locations $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,\omega_i})$. The domain is still single-peaked; when an agent hopes to minimize the sum of distances to her locations, her peak is the median point of her locations. As stated in Section 1.2, we can easily apply Theorem 2 to general single-peaked domains. Thus, we obtain the following corollary.

COROLLARY 2. *A mechanism f for the multiple locations setting is false-name-proof, Pareto efficient, and anonymous if and only if there exists a real number α such that for all $n \in \mathbb{N}$, for all $\omega_i|_{i \in N}$, and for all reported location profiles $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^n$,*

$$f(\mathbf{x}) = \mathrm{med}(\mathrm{med}(\mathbf{x}_1), \ldots, \mathrm{med}(\mathbf{x}_n), \overbrace{\alpha, \ldots, \alpha}^{n-1}). \tag{5}$$

Procaccia and Tennenholtz [8] developed a valuable mechanism, which creates ω_i copies of the median of each agent i and returns the median point among all copies. In their setting, the number of locations ω_i owned by agent i is public.

Table 1: Summary of the approximation ratios achieved by deterministic strategy-proof/false-name-proof mechanisms on a real line. UB and LB indicate upper and lower bounds. SP and FNP indicate strategy-proof and false-name-proof.

	SP	FNP
Social Cost	UB: 1 LB: 1	UB: $O(n)$ (Thm. 4) LB: $\Omega(n)$ (Thm. 3)
Maximum Cost	UB: 2 ([8]) LB: 2 ([8])	UB: 2 LB: 2 (Thm. 5)

This means that the domain is not anonymous; a change of agents' location profiles affects the outcome. In contrast, we deal with anonymous mechanisms in the above corollary by assuming that ω_i is the private information of i.

3.2 Approximation Ratios

In this subsection, we analyze the performance of false-name-proof mechanisms from the viewpoint of *approximate mechanism design without money* [8]. Table 1 summarizes the results of this section. Our results, shown in the rightmost column, provide tight bounds of the approximation ratios for both social and maximum costs.

3.2.1 Social Cost

We first consider social cost as an objective function and show the lower bound of the approximation ratio achieved by deterministic false-name-proof mechanisms for it in Theorem 3. Theorem 4 shows that the lower bound is tight.

THEOREM 3. *Any deterministic false-name-proof mechanism has an approximation ratio of $\Omega(n)$ for social cost.*

PROOF. First, consider a location profile $x = (0, 1)$ and assume that $f(x) = y \in \mathbb{R}$. If $y \neq 0$, then consider another location profile $x' = (0, \ldots, 0, 1)$ where $|x'| = n$. From false-name-proofness, $y' = f(x')$ must satisfy $|y'| \geq |y| = \mathrm{cost}(0, f(x))$. In this case, the social cost with respect to x' becomes $(n-1)|y'| + |y' - 1| \geq (n-1)|y| = (n-1) \cdot \mathrm{cost}(0, f(x))$, which depends on the number of agents n. If $y = 0$, we can apply a similar argument by considering a location profile $x' = (0, 1, \ldots, 1)$. □

THEOREM 4. *The leftmost mechanism has an approximation ratio of $n - 1$ for the social cost.*

PROOF. The leftmost mechanism is a false-name-proof mechanism whose parameter α is defined as $-\infty$. For any reported location profile x, the approximation ratio of the leftmost mechanism for the social cost with respect to x is defined as $\sum_{i \neq 1} |x_i - x_1| / \sum_{i \neq \lceil n/2 \rceil} |x_i - x_{\lceil n/2 \rceil}|$. Here the denominator is the social cost of the median mechanism that has an approximation ratio of 1 for social cost. This ratio is at most

$$\frac{\sum_{i \neq 1} |x_n - x_1|}{\sum_{i=1,n} |x_i - x_{\lceil n/2 \rceil}|} = \frac{(n-1)(x_n - x_1)}{x_n - x_1} = n - 1,$$

and we have equality if x satisfies $x_1 < x_2 = \ldots = x_n$. □

3.2.2 Maximum Cost

In contrast to Section 3.2.1, we consider maximum cost as an objective function in this subsection. First, we show the lower bound of an approximation ratio for maximum cost.

THEOREM 5. *Any deterministic false-name-proof mechanism has an approximation ratio of at least 2 for maximum cost.*

The proof is straightforward. It was shown by Procaccia and Tennenholtz [8] that any deterministic strategy-proof mechanism has an approximation ratio of at least 2 for maximum cost. Since false-name-proofness implies strategy-proofness, the lower bound does not decrease if we require false-name-proofness.

As stated in Section 3.1, the leftmost mechanism is false-name-proof. Furthermore, the leftmost mechanism has an approximation ratio of 2 for maximum cost by [8]. This implies that the bound obtained in Theorem 5 is tight.

4. EXTENDED RESULTS

In Section 3, we showed a basic result on false-name-proof mechanisms on a real line. We then extend the result in two directions. In Section 4.1, we discuss the bound of approximation ratios achieved by randomized false-name-proof mechanisms. In Section 4.2, we characterize deterministic false-name-proof mechanisms on a tree.

4.1 Randomized Mechanisms

Our results shown in Section 3.2 suggest the difficulty of designing a deterministic false-name-proof mechanism that achieves good approximation ratios, even if the domain is a real line. One natural approach to this problem is to use *randomized* mechanisms, which return a probability distribution over a real line for a given location profile. In this subsection, we discuss whether allowing randomization enables mechanisms to achieve better approximation ratios.

First, let us introduce a mechanism called *left-right-middle*, which was developed by Procaccia and Tennenholtz [8].

MECHANISM 1 (LEFT-RIGHT-MIDDLE). *Given a location profile $x = (x_1, \ldots, x_n)$, the* left-right-middle *mechanism locates a facility at x_1 with probability 1/4, x_n with probability 1/4, and $(x_1 + x_n)/2$ with probability 1/2.*

Note that the cost of an agent is defined as the expected distance from the location. Also, approximation ratios can be redefined over a distribution. Now we confirm that the left-right-middle mechanism is false-name-proof and calculate the approximation ratio for social cost.

THEOREM 6. *The left-right-middle mechanism is false-name-proof and has an approximation ratio of $n/2$ for social cost.*

PROOF. First we show that the left-right-middle mechanism is false-name-proof. The mechanism defines the outcome depending only on the leftmost and rightmost locations. Thus, from a similar argument for strategy-proofness, no agent can be better off by any false-name manipulations.

We now turn to proving that the left-right-middle mechanism is $n/2$-approximation. For any reported location profile x, the approximation ratio of the left-right-middle mechanism for social cost with respect to x is defined as

$$\frac{\frac{1}{4}\sum_i |x_i - x_1| + \frac{1}{4}\sum_i |x_i - x_n| + \frac{1}{2}\sum_i |x_i - \frac{x_1 + x_n}{2}|}{\sum_i |x_i - x_{\lceil n/2 \rceil}|}.$$

This is at most

$$\frac{\frac{1}{4}\sum_{i \in N}(x_i - x_1) + \frac{1}{4}\sum_{i \in N}(x_n - x_i) + \frac{1}{2}\sum_{i \in N}(x_n - \frac{x_1 + x_n}{2})}{\sum_{i=1,n} |x_i - x_{\lceil n/2 \rceil}|}$$
$$= \frac{\frac{n}{2}(x_n - x_1)}{x_n - x_1} = \frac{n}{2},$$

and equality holds if x satisfies $x_1 < x_2 = \ldots = x_n$. □

This shows us that with randomization, we can slightly improve the social cost than with deterministic mechanisms, e.g., the leftmost mechanism, when the number of agent n is large. However, from an algorithmic point of view, the performances of these mechanisms are essentially the same: both have an approximation ratio of $O(n)$ for social cost. Thus, we next discuss if there exist randomized false-name-proof mechanisms which have an essentially better approximation ratio for social cost. We show a lower bound of the approximation ratio for social cost and answer the question.

THEOREM 7. *Any randomized false-name-proof mechanism has an approximation ratio of $\Omega(n)$ for social cost.*

PROOF. Consider arbitrary randomized false-name-proof mechanism f. Let $x = (0, 1)$ be a location profile when there are two agents and let $P = f(x)$ be the outcome distribution over $\mathbb{R}$. Intuitively, $\mathrm{cost}(0, P) + \mathrm{cost}(1, P) \geq 1$ holds (formally proved in [8], Lemma 2.6). Thus, we assume $\mathrm{cost}(1, P) \geq 1/2$ without loss of generality.

Then, we consider the case with n agents $1, \ldots, n$ and the reported location profile $x' = (0, 1, \ldots, 1)$. Let $P' = f(x')$ be the outcome distribution. Since f is false-name-proof, $\mathrm{cost}(1, P') \geq \mathrm{cost}(1, P)$. Thus, the social cost is at least $(n-1)/2$. On the other hand, the optimal solution with respect to the profile x' is to locate a facility at 1, in which the social cost is 1. Thus, the ratio is $(n-1)/2$. □

That is, even if randomization is allowed, the approximation ratio of a false-name-proof mechanism for social cost ever depends on the number of agents n.

In contrast to social cost, we can obtain a tight bound for maximum cost from the result of Procaccia and Tennenholtz [8]. They showed that the left-right-middle mechanism achieves an optimal approximation ratio of 3/2 for maximum cost. Furthermore, as shown in Theorem 6, the left-right-middle is false-name-proof. Thus, the tight bound of the approximation ratio for maximum cost is 3/2.

4.2 Location on a Tree

Several application domains of facility locations have much more complicated structures, i.e., graph structure, than a simple line. Thus, facility location problems on a graph are natural extensions of the 1-dimensional case to such application domains, as discussed in Section 3. One simple structure of graphs is a tree [1, 9]. Therefore, we characterize deterministic false-name-proof mechanisms on a tree.

First, let us introduce additional notations. Let G be a *tree*, which is a finite connected graph composed of the union of a finite number of curves of finite length and contains no cycle. Let $L(G) \subset G$ be a set of leaves of G. For any two points $p, q \in G$, let $[p, q]$ denote the *path* between p, q. Note that we can define a unique path $[p, q]$ for all p, q since G contains no cycle. Also, let $d(p, q)$ denote the *distance* between two points p, q, which is defined as the path-length between the two points. When a facility is located at $y \in G$, the cost of an agent i with true location $x_i \in G$ is defined as $\mathrm{cost}(x_i, y) = d(x_i, y)$.

Although each agent still has a peak on the tree G, this setting is no longer a single-peaked domain because we cannot order all the points on the tree G according to any linear order in which every agent has a single-peaked preference. Thus, we cannot straightforwardly apply our result obtained in Section 3.1 to this setting.

Let us introduce a well-known (group) strategy-proof mechanism, which is an generalization of the median mechanism on a real line. We refer to it as the *tree-median* mechanism.

MECHANISM 2 (TREE-MEDIAN). *A* tree-median *mechanism on a tree G has a fixed parameter (root) $\beta \in G$ and, for all n and for all reported location profiles, starts from β. As long as the current point has a subtree that contains at least $n/2$ locations, it smoothly moves down this subtree. When it reaches a point that does not have such a subtree, locates a facility at this point.*

As stated in Alon et al. [1], the tree-median mechanism achieves the optimal approximation ratio for social cost. However, obviously it is not false-name-proof, since it behaves in the same manner as the original median mechanism when all agents are on a single path.

We then characterize false-name-proof mechanisms on a tree. First, to simplify notations, let us define a *Pareto efficient set* with respect to a given location profile.

DEFINITION 5 (PARETO EFFICIENT SET). *For a tree G and for a location profile $x = (x_1, \ldots, x_n) \in G^n$, a set of points $PE(x) \subseteq G$ is said to be* Pareto-efficient *for x if $\forall y \in PE(x)$, $\forall y' \in G$, y' does not dominate y for x.*

Here, we say $y' \in G$ *dominates* $y \in G$ for a location profile x if $\forall i \in N, d(x_i, y) \leq d(x_i, y')$ and $\exists j \in N, d(x_j, y) < d(x_j, y')$. By using this notation, we define a class of mechanisms on a tree called *Pareto-improving relocation* rules.

MECHANISM 3 (PARETO-IMPROVING RELOCATION). *A mechanism f on a tree G is a* Pareto-improving relocation *rule if it has a fixed point $\beta \in G$ such that for all $n \in \mathbb{N}$ and for all reported location profiles $x = (x_1, \ldots, x_n) \in G^n$,*

$$f(x) = \arg \min_{z \in \mathrm{PE}(x)} d(z, \beta). \quad (6)$$

Now we show our characterization theorem; a class of false-name-proof, Pareto efficient, and anonymous mechanisms consists exactly of Pareto-improving relocation rules. It is shown separately in Lemmas 3 and 4.

THEOREM 8. *For any tree G, a mechanism f is false-name-proof, Pareto efficient, and anonymous if and only if it is a Pareto-improving relocation rule.*

LEMMA 3. *If a mechanism f for a tree G is false-name-proof, Pareto efficient, and anonymous, then it is a Pareto-improving relocation rule.*

PROOF. Consider a deterministic mechanism f that is false-name-proof, Pareto efficient, and anonymous. Since f is deterministic, it returns a point with probability 1 for a reported location profile. Now choose a location profile x^L that exactly contains every leaf $L(G)$ of the tree G. We then prove that f is a Pareto-improving relocation rule with a parameter $\beta = f(x^L)$. More precisely, we show that $f(x) = \arg\min_{z \in \mathrm{PE}(x)} d(z, \beta)$ holds for the above $\beta = f(x^L)$ and any location profile x.

First, we show that $f(x^L, x) = \beta$ holds for all x. Suppose not; there exists at least one location profile x such that $f(x^L, x) \neq \beta$. Here, let $f(x^L, x)$ indicate an outcome

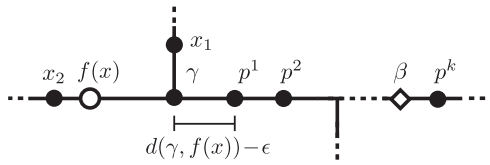


Figure 1: Sequence $p^1, p^2, \ldots, p^k$ in the proof of Lemma 3. Note that $x = (x_1, x_2)$. False-name-proofness implies $f(x, p_1, p_2, \ldots, p_k) = f(x)$. However, it contradicts Eq. (7).

location for an input location profile that is a joint of two location profiles, x^L and x. From Pareto efficiency, there exists at least one location x_i^L in the profile x^L such that $d(x_i^L, f(x^L, x)) < d(x_i^L, \beta)$. Thus, when x^L is a true location of the agents, agent i at x_i^L can strictly reduce her cost by reporting (x_i^L, x) under false identifiers. This contradicts the assumption that f is false-name-proof.

Next, let us show that

$$\forall x \text{ such that } \beta \in \mathrm{PE}(x), f(x) = \beta. \tag{7}$$

Suppose not; there exists at least one profile x such that $\beta \in \mathrm{PE}(x) \wedge f(x) \neq \beta$. From Pareto efficiency, there exists at least one location x_i in the profile x such that $d(x_i, \beta) < d(x_i, f(x))$. Thus, when x is a true location profile, agent i at x_i can strictly reduce her cost by reporting (x_i, x^L). This contradicts the assumption that f is false-name-proof.

Finally, let us show that

$$\forall x \text{ such that } \beta \notin \mathrm{PE}(x), f(x) = \gamma \tag{8}$$

where $\gamma = \arg\min_{z \in \mathrm{PE}(x)} d(z, \beta)$. Suppose not; there exists at least one profile x such that $\beta \notin \mathrm{PE}(x) \wedge f(x) \neq \gamma$. From Pareto efficiency, $f(x) \in \mathrm{PE}(x)$ holds. Then we can move from γ toward β with distance $d(\gamma, f(x)) - \epsilon$ and refer to the point as p^1. For this p^1, there exists at least one location x_i such that $\forall z \in [\gamma, p^1], d(x_i, z) < d(x_i, f(x))$. If $f(x, p^1) \in \mathrm{PE}(x) \setminus f(x)$, there exists at least one agent who strictly prefers $f(x, p^1)$ to $f(x)$. She can reduce her cost by adding a location p^1 under a false identifier. Also, if $f(x, p^1) \in (\gamma, p^1]$, agent i at x_i has an incentive to report (x_i, p^1). Note that $(\gamma, p^1]$ indicates a half-open interval. Thus, from false-name-proofness, $f(x, p^1) = f(x)$ must hold.

If $\beta \in (\gamma, p^1]$, the above equation contradicts Eq. (7). If $\beta \notin (\gamma, p^1]$, we can construct a finite sequence of points $p_1, p_2, \ldots, p_k$ by applying the same argument (see Fig. 1) and obtain $\beta \in (\gamma, p_k] \wedge f(x, p_1, \ldots, p_k) = f(x) \neq \beta$. This contradicts Eq. (7). □

LEMMA 4. *If a mechanism f for a tree G is a Pareto-improving relocation rule, then it is false-name-proof, Pareto efficient, and anonymous.*

PROOF. Clearly, a Pareto-improving relocation rule is Pareto efficient and anonymous. To prove this lemma, it suffices to show that f is false-name-proof if it is a Pareto-improving relocation rule.

From the definition, a Pareto-improving relocation rule f has a fixed parameter $\beta \in G$. Let us fix a location profile x_{-i} and consider false-name manipulations by i. If $\beta \in \mathrm{PE}(x_{-i})$ holds, f always locates a facility at β regardless of i's strategy and satisfies false-name-proofness. Then we focus on showing that no false-name manipulation strictly reduces her cost when $\beta \notin \mathrm{PE}(x_{-i})$ holds.

Let us define a point $\gamma = \arg\min_{z \in \mathrm{PE}(x_{-i})} d(z, \beta)$. From the definition of a Pareto-improving relocation rule, we have $f(x_{\phi_i}, x_{-i}) \in [\gamma, \beta]$ for any x_{ϕ_i} and $f(x_i, x_{-i}) = \arg\min_{z \in [\gamma, x_i]} d(z, \beta)$ for any x_i. This means that $f(x_i, x_{-i})$ is the closest point to x_i in the range $[\gamma, \beta]$. Thus, we obtain $d(x_i, f(x_i, x_{-i})) \leq d(x_i, f(x_{\phi_i}, x_{-i}))$ for any x_i and x_{ϕ_i}. □

Theorem 8 can be considered an extension of the result by Schummer and Vohra [9], which characterized the class of strategy-proof and Pareto efficient mechanisms on a tree. Now let us introduce the relationship between these two characterizations. We assume in this paper that mechanisms are anonymous, while [9] did not. With the assumption of anonymity, mechanisms characterized in [9] behave in the same manner as those described in Eq. (1), when all agents are on a single path. To achieve false-name-proofness when all agents are on a single path, each "partial" mechanism defined on each single path must be described in Eq. (2); for any two leaves $l_1, l_2 \in L(G)$, the path $[l_1, l_2]$ has a fixed parameter $\alpha_{l_1, l_2} \in [l_1, l_2]$. Here, as discussed in [9], these partial mechanisms must be self-consistent in some way; they must agree on the intersection of their paths. This consistency requires that each parameter of each partial mechanism must be defined as the closest point to a fixed $\beta \in G$ on the path, which is identical to Mechanism 3.

5. DISCUSSIONS

In the literature of social choice and mechanism design, several properties have been introduced. In this section, we clarify the connections between false-name-proofness and three other properties in facility location problems on a tree.

5.1 Population Monotonicity

Population monotonicity in public goods environments was originally identified in Ching and Thomson [3] Informally, population monotonicity requires that the arrival of a new agents affects all agents initially present in the same direction. However, with the assumption of Pareto efficiency, we can define the property in a more restricted way:

DEFINITION 6 (POPULATION MONOTONICITY). *A mechanism f is* population monotonic *if $\forall n \in \mathbb{N}$, $\forall x$, $\forall j \in N$, $\forall i \neq j$, $\mathrm{cost}(x_i, f(x)) \geq \mathrm{cost}(x_i, f(x_{-j}))$.*

This is somehow reminiscent of false-name-proofness. Both deal with the change of the number of agents. The following theorem shows the equivalence of these two properties.

THEOREM 9. *Under the assumptions of Pareto efficiency and anonymity, a mechanism f is population monotonic if and only if it is false-name-proof.*

PROOF. Ching and Thomson [3] gave a characterization of population monotonic mechanisms under the assumption of Pareto efficiency. Their characterization is identical to our characterization of false-name-proofness (Theorem 8). □

Note that without the assumption of Pareto efficiency, false-name-proofness and population monotonicity do not coincide even in the case of a real line. Consider the following mechanism for a real line: if $n < 3$, then locate a facility at the point that is slightly smaller than the leftmost reported location, otherwise use the leftmost mechanism. This mechanism is population monotonic, although it is not false-name-proof (not even strategy-proof).

5.2 Group-strategyproofness

Group-strategyproofness has been widely discussed in economics. A mechanism is *group-strategyproof* if for any location profile and any coalition of agents, there is no joint deviation of the coalition such that every agent in the coalition strictly reduces her cost. For the connection to false-name-proofness, we show the next theorem. For space reasons, we omit the proof.

THEOREM 10. *Under the assumptions of Pareto efficiency and anonymity, any false-name-proof mechanism f is group-strategyproof.*

It has been known that the tree-median mechanism is group-strategyproof. However, as stated in Section 4.2, it is not false-name-proof. In other words, the class of false-name-proof mechanisms is a strict subset of the class of group-strategyproof mechanisms under the assumptions of Pareto efficiency and anonymity.

5.3 Anonymity-proofness

Anonymity-proofness, which was first proposed by Conitzer [4], is an extension of false-name-proofness. First, to define anonymity-proofness, we introduce the notion of *participation*. A mechanism f satisfies participation if $\forall n \in \mathbb{N}, \forall i \in N, \forall x_{-i}, \forall x_i, \mathrm{cost}(x_i, f(x_i, x_{-i})) \leq \mathrm{cost}(x_i, f(x_{-i}))$. That is, for each agent, it never hurts her to join the mechanism as long as she behaves sincerely.

DEFINITION 7 (ANONYMITY-PROOFNESS). *A mechanism f is* anonymity-proof *if it is false-name-proof and satisfies participation.*

The next theorem shows the equivalence of anonymity-proofness and false-name-proofness.

THEOREM 11. *Under the assumptions of Pareto efficiency and anonymity, a mechanism f is anonymity-proof if and only if f is false-name-proof.*

PROOF. From the definition of anonymity-proofness, f is false-name-proof if it is anonymity-proof. To prove this theorem, it suffices to show that f satisfies participation if it is false-name-proof. From Theorem 8, a false-name-proof f is a Pareto-improving relocation rule; it has a parameter β. Now let us fix the location profile x_{-i} reported by agents except i and focus on i's strategy.

Let us define $\gamma = \arg\min_{z \in \mathrm{PE}(x_{-i})} d(z, \beta)$. Note that $f(x_{-i}) = \gamma$ holds from the definition of the Pareto-improving relocation rule. If $\mathrm{PE}(x_i, x_{-i}) \cap (\gamma, \beta] = \emptyset$ holds, f always locates a facility at γ regardless whether agent i participates. Thus, f satisfies participation.

If $\mathrm{PE}(x_i, x_{-i}) \cap (\gamma, \beta] \neq \emptyset$ holds, we can find a point γ' such that $\gamma' = \arg\min_{z \in \mathrm{PE}(x_i, x_{-i})} d(z, \beta) \wedge \gamma' \in (\gamma, x_i]$. Here $f(x_i, x_{-i}) = \gamma'$ holds from the definition of Pareto-improving relocation rule. Thus, we obtain $\mathrm{cost}(x_i, f(x_{-i})) = d(x_i, \gamma) > d(x_i, \gamma') > \mathrm{cost}(x_i, f(x_i, x_{-i}))$, and f satisfies participation. □

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a case study of false-name-proof mechanism design without money by dealing with facility location problems. We first characterized deterministic false-name-proof mechanisms on a real line and established the tight bounds of approximation ratios. We then discussed the approximation ratios achieved by randomized false-name-proof mechanisms. Also, we characterized deterministic false-name-proof mechanisms on a tree. Furthermore, we clarified the connections between false-name-proofness and other related properties.

We outline our future direction of false-name-proof mechanism design without money. To the best of our knowledge, there exists no work that discussed the effect of false-name manipulations in private goods environments without monetary transfers, e.g., resource allocations [5]. Intuitively, false-name manipulations must become much more powerful strategic behaviors in such environments. We would like to find a solution to prevent false-name manipulations, evaluate it using techniques of approximate mechanism design without money, and characterize the solutions.

7. ACKNOWLEDGMENTS

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (A), 20240015, 2008, and Grant-in-Aid for JSPS research fellows. The authors would like to thank anonymous reviewers of AAMAS'11 and attendees of WINE'10.

8. REFERENCES

[1] N. Alon, M. Feldman, A. D. Procaccia, and M. Tennenholtz. Strategyproof approximation of the minimax on networks. *Mathematics of Operations Research*, 35(3):513–526, 2010.

[2] Y. Bachrach and E. Elkind. Divide and conquer: false-name manipulations in weighted voting games. In *AAMAS'08*.

[3] S. Ching and W. Thomson. Population-monotonic solutions in public good economies with single-peaked preferences. *Social Choice and Welfare*, forthcoming.

[4] V. Conitzer. Anonymity-proof voting rules. In *WINE'08*.

[5] M. Guo and V. Conitzer. Strategy-proof allocation of multiple items between two agents without payments or priors. In *AAMAS'10*.

[6] H. Moulin. On strategy-proofness and single peakedness. *Public Choice*, 35(4):437–455, 1980.

[7] B. Peleg. Game-theoretic analysis of voting in committees. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, chapter 8, pages 395–423, 2002.

[8] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. In *ACM-EC'09*.

[9] J. Schummer and R. V. Vohra. Strategy-proof location on a network. *Journal of Economic Theory*, 104(2):405–428, 2004.

[10] J. Schummer and R. V. Vohra. Mechanism design without money. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 10, 2007.

[11] T. Todo, A. Iwasaki, M. Yokoo, and Y. Sakurai. Characterizing false-name-proof allocation rules in combinatorial auctions. In *AAMAS'09*.

[12] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.

Majority-rule-based preference aggregation on multi-attribute domains with CP-nets

Minyi Li
Swinburne University of Technology
myli@swin.edu.au

Quoc Bao Vo
Swinburne University of Technology
BVO@swin.edu.au

Ryszard Kowalczyk
Swinburne University of Technology
RKowalczyk@swin.edu.au

ABSTRACT

This paper studies the problem of majority-rule-based collective decision-making where the agents' preferences are represented by CP-nets (Conditional Preference Networks). As there are exponentially many alternatives, it is impractical to reason about the individual full rankings over the alternative space and apply majority rule directly. Most existing works either do not consider computational requirements, or depend on a strong assumption that the agents have acyclic CP-nets that are compatible with a common order on the variables. To this end, this paper proposes an efficient SAT-based approach, called `MajCP` (Majority-rule-based collective decision-making with CP-nets), to compute the majority winning alternatives. Our proposed approach only requires that each agent submit a CP-net; the CP-net can be cyclic, and it does not need to be any common structures among the agents' CP-nets. The experimental results presented in this paper demonstrate that the proposed approach is computationally efficient. It offers several orders of magnitude improvement in performance over a Brute-force algorithm for large numbers of variables.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Design

Keywords

CP-nets; Voting; Preference aggregation; Majority rule

1. INTRODUCTION

Group decision making where a collective decision needs to be derived from individual preferences has been an active area of research [1]. In particular, various aggregation rules and voting procedures have been developed as group decision-making mechanisms [9]. However, the decision-making process tends to become much more complex when the attributes of the domain are interdependent. As an example, a research group plan to order several PCs and the group members need to decide on a standard group PC configuration. The decisions are not independent, because, perhaps, the preferred operating systems may depend on the given processor type. For instance, "I prefer to choose WinXP operating system rather than Linux if an Intel processor is given." Hence, we cannot decide on the issues separately. Moreover, in many real world decision-making problems, the number of alternatives is exponential in the number of domain variables. The prohibitive size of such combinatorial domain makes it impractical to represent preference relations explicitly.

In this paper, we investigate the theory of CP-nets as a formal model for representing and reasoning with the agents' preferences. There are some preference relations can not be modelized by CP-nets and its variants. For instance, Domshlak *et al.* [5] compare compare the expressive power of soft constraints and CP-nets and study several examples in which the preference relations can not be represented by CP-nets. However, CP-nets are quite commonly used and to some extent, representative of a variety of languages. Moreover, CP-nets and its variants can be used to specify individual preference relations in a relatively compact, intuitive, and structured manner, making it easier to encode human preferences and supports the decision-making systems in real world applications.

In this paper, given that the individual preferences have been elicited and represented as CP-nets, the problem of majority-rule-based preference aggregation will be addressed. Recent work on the complexity of computing dominance relations shows that dominance testing[1] for an arbitrary CP-net is PSPACE-complete [6]. However, computing the majority winning alternatives with multiple agents' CP-nets may furthermore require dominance testing on each pair of alternatives on each individual CP-net. For example, having 10 binary variables, each involved agent would need to compare $\binom{2^{10}}{2} = 523776$ pairs of alternatives. This problem is likely to be even harder than NP or coNP problems. The problem of computing aggregation rules from a collection of CP-nets has been studied in the literature, e.g.,[8, 10]. In particular, Lang and Xia [8] consider decomposition with voting rules assuming that the agents' preferences can be represented with acyclic CP-nets being compatible with a common order on the variables. However, such an assumption is unlikely to be applicable in most real world applications [11]. Xia *et al.* [12] partially addressed this shortcoming by introducing an *order-independent* sequential composition of voting rules. In their framework, the profile is still required to be compatible with *some* order on the variables, but this order is not specified in the definition of the rule. Nevertheless, the domain restriction by this order-independent sequential composition of voting rules is still severe: there must exist some (unspecified) directed acyclic graph that the profile is compatible with. Xia *et al.* [11] generalize the earlier, more restrictive method, proposing an aggregation methodology that does not require any relationship among the

[1] A dominance testing, given an individual CP-net and two alternatives o and o', tests whether o is preferred to o' according to the preferences induced by that CP-net.

Cite as: Majority-rule-based preference aggregation on multi-attribute domains with CP-nets, Minyi Li, Quoc Bao Vo and Ryszard Kowalczyk, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 659-666.

agents' CP-net structures. However, the performance of their algorithm also depends on the consistency among the structures of the agents' CP-nets.

To this end, our paper addresses the above drawbacks, proposing an efficient SAT-based approach, called MajCP (Majority-rule-based collective decision-making with CP-nets), to compute the majority winning alternatives. The proposed approach allows the agents to have different preferential independence structures, and enables us to aggregate preferences when the agents' CP-nets are cyclic. With multiple agents' CP-nets as input, it first reduces the problem into an extended SAT (Boolean satisfiability problem) for cardinality constraints, such that the set of possible winners can be obtained by computing the models of the corresponding SAT. Then the set of majority winners is the subset of the possible winners after filtering out those that are majority-dominated by some alternative. The proposed approach reduces the search space and is computationally efficient. According to the experimental evaluation, it offers several orders of magnitude improvement in performance over a brute-force algorithm for large numbers of variables.

The paper is structured as follows. We provide background information about CP-nets and majority rule in Section 2. In Section 3, we study a hypercube-wise composition of majority rule and analyze its incompatibility with the original majority preferences by several examples. After that, we present our proposed approach for computing the winning alternatives in Section 4 and the experimental results in Section 5. Finally, we discuss about the concluding remarks in Section 6.

2. BACKGROUND

2.1 CP-nets overview

Let $\mathbf{V} = \{X_1, \ldots, X_n\}$ be a set of n variables. For each $X_i \in \mathbf{V}$, $D(X_i)$ is the *value domain* of X_i. A variable X_i is *binary* if $D(X_i) = \{x_i, \bar{x}_i\}$. If $\{x_i, \bar{x}_i\}$ is the binary domain of X_i, then $x_i = \neg\bar{x}_i$; $\bar{x}_i = \neg x_i$. If $\mathbf{X} = \{X_{i_1}, \ldots, X_{i_p}\} \subseteq \mathbf{V}$, with $i_1 < \cdots < i_p$, then $D(\mathbf{X})$ denotes $D(X_{i_1}) \times \cdots \times D(X_{i_p})$. The assignments of variable values to $\mathbf{X}$ are denoted by $\mathbf{x}$, $\mathbf{x}'$ etc., and represented by concatenating the values of the variables. For instance, if $\mathbf{X} = \{X_1, X_2, X_3\}$, an assignment $\mathbf{x} = x_1\bar{x}_2x_3$ assigns x_1 to X_1, $\bar{x}_2$ to X_2 and x_3 to X_3. If $\mathbf{X} = \mathbf{V}$, $\mathbf{x}$ is a *complete assignment*; otherwise $\mathbf{x}$ is called a *partial assignment*. For an assignment $\mathbf{x}$, we denote by $\mathbf{x}[X_i]$ the value $x_i \in D(X_i)$ assigned to variable X_i by that assignment; and $\mathbf{x}[\mathbf{W}]$ denotes the assignment of the variable values $\mathbf{w} \in D(\mathbf{W})$ assigned to the set of variables $\mathbf{W} \subseteq \mathbf{X}$ by that assignment. We also allow logical operations between the value assignments to binary variables. For instance, $x_1\bar{x}_2 = x_1 \wedge \bar{x}_2 = (X_1 = x_1) \wedge (X_2 = \bar{x}_2)$. That is, x_1 is $True$ and x_2 is $False$. If $\mathbf{p} = x_1\bar{x}_2$ and $\mathbf{q} = x_3$, then $\mathbf{p} \vee \mathbf{q} = (x_1\bar{x}_2) \vee x_3 = ((X_1 = x_1) \wedge (X_2 = \bar{x}_2)) \vee (X_3 = x_3)$.

Let $\{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ be a partition of the set of variables $\mathbf{V}$ and $\succ$ a preference relation over $D(\mathbf{V})$. $\mathbf{X}$ is *conditionally preferentially independent* of $\mathbf{Y}$ given $\mathbf{Z}$ if and only if, for all $\mathbf{x}, \mathbf{x}' \in D(\mathbf{X})$, $\mathbf{y}, \mathbf{y}' \in D(\mathbf{Y})$ and $\mathbf{z} \in D(\mathbf{Z})$:

$$\mathbf{xyz} \succ \mathbf{x}'\mathbf{yz} \text{ iff } \mathbf{xy}'\mathbf{z} \succ \mathbf{x}'\mathbf{y}'\mathbf{z}$$

A CP-net $\mathcal{N}$ [3] over a set of variables $\mathbf{V} = \{X_1, \ldots, X_n\}$ is an annotated directed graph G over $X_1, \ldots, X_n$, in which nodes stand for the problem variables. Each node X_i is annotated with a conditional preference table $CPT(X_i)$, which associates a total order $\succ^{X_i|\mathbf{u}}$ with each instantiation $\mathbf{u}$ of X_i's parents $Pa(X_i)$. For instance, let $\mathbf{V} = \{X_1, X_2, X_3\}$, all three variables are binary-valued. Assume that the preference of a given agent over $2^{\mathbf{V}}$ can be defined by a CP-net, whose structural part is the directed graph $G = \{(X_1, X_2), (X_2, X_3), (X_1, X_3)\}$. Then the agent's preference over the values of X_1 is unconditional, preference over the values of X_2 (resp. X_3) is conditioned on the value of X_1 (resp. the context of X_1 and X_2). The conditional preference statements contained in the CPTs are written with the following notation, e.g. $x_1\bar{x}_2 : x_3 \succ \bar{x}_3$ means that if x_1 is $True$ and x_2 is $False$, then the agent prefers $X_3 = x_3$ to $X_3 = \bar{x}_3$.

In this paper, we assume that each agent A_j's preference is captured by a binary-valued (possibly cyclic) CP-net $\mathcal{N}_j$ and the ordering $\succ_{A_j}^{X_i|\mathbf{u}}$, $\mathbf{u} \in D(Pa_j(X_i))$, expressed in the CPTs of the network is total. As such, conditional expressions of indifference are not allowed, and an agent will not be indifferent between two alternatives. However, as the preference relation induced from a CP-net is generally not complete, two alternatives can be incomparable for an agent.

2.2 Majority rule

In classical social choice theory, majority rule is one of the most well known aggregation rule for collective decision-making. It is a binary decision rule that selects one of two alternatives, based on which has more than half of the votes. The semantics of majority voting in the context of CP-nets has been provided by Rossi *et al.* [10]:

DEFINITION 1 (MAJORITY SEMANTICS). *Given two alternatives o and o', let $\boldsymbol{S}_\succ$, $\boldsymbol{S}_\prec$, $\boldsymbol{S}_\bowtie$ be the sets of agents who say, respectively, that $o \succ o'$, $o \prec o'$, and $o \bowtie o'$ (incomparable). We say that o majority-dominates o' (written as $o \succ_{maj} o'$) if and only if there is a majority of agents who prefer o to o' (i.e., $|\boldsymbol{S}_\succ| > |\boldsymbol{S}_\prec| + |\boldsymbol{S}_\bowtie|$). Two alternatives o and o' are majority-incomparable (written as $o \bowtie_{maj} o'$) if they are not ordered in either way.*

In order to determine the winning alternatives according to majority rule, the Condorcet method has usually been used [2]. The following definitions of the Condorcet winner and weak Condorcet winner are adapted from the standard social choice literature [1]:

DEFINITION 2 (CONDORCET WINNER). *An alternative o is a Condorcet winner if and only if it majority-dominates every other alternative in a pair-wise matchup: $\forall o' \in O$ and $o' \neq o$, $o \succ_{maj} o'$.*

DEFINITION 3 (WEAK CONDORCET WINNER). *An alternative o is a* weak *Condorcet winner if and only if it majority-dominates or is incomparable to every other alternative in a pair-wise matchup: $\forall o' \in O$ and $o' \neq o$, $o \succ_{maj} o'$ or $o \bowtie_{maj} o'$.*

When the Condorcet winner exists, it is unique. A Condorcet winner is also a weak Condorcet winner, while the reverse does not hold: a weak Condorcet winner is not necessarily a Condorcet winner. In majority-rule based group decision-making, it is possible for a paradox to form, in which collective preferences can be cyclic (i.e. not transitive), even if the preferences of individual agents are not. For instance, it is possible that there are alternatives o_1, o_2, and o_3 such that a majority prefers o_1 to o_2, another majority prefers o_2 to o_3, and yet another majority prefers o_3 to o_1. The requirement of majority rule then provides no Condorcet winner. Consequently, the set of majority winning alternatives can be empty. Also, there can be more than one weak Condorcet winner when the number of agents is even or the individual preferences are incomplete (i.e.

[2]There are also some other aggregation methods which do not comply with the Condorcet criterion, e.g., approval voting, Borda count, plurality voting, etc..

partial order). Note that the set of weak Condorcet winners are majority-incomparable to each other.

Rossi *et al.* [10] study the computational complexity of a brute-force algorithm for aggregating preference based on majority rule. Suppose that there are a set of m agents making decisions over a set of n binary variables. To test whether an alternative is a winner we need to compare the given alternative with all other alternatives (2^n) in all CP-nets (m). Recall that computing the majority-dominance relation between a pair of alternatives require individual dominance testing on each agent's CP-net, which is PSPACE-complete. Moreover, finding the set of majority winners is even more challenging. We need to compare all alternatives (2^n) to all other alternatives (2^n) in all CP-nets(m). Consequently, it is impractical to use pair-wise comparison over the alternative space directly.

3. H-COMPOSITION OF MAJORITY RULES

Instead of applying voting directly over the alternative space, Xia *et al.* [11] propose a *hypercube-wise composition* (*H-composition*) of local voting rules. An H-composition of local rules is defined as the following two steps. First, the set of all possible alternatives are represented as a hypercube, and alternatives that differ on only one variable are neighbours on this hypercube as discussed in [4]. Then an induced graph is generated by applying local rules to each pair of neighbours on this hypercube. In the second step, a *choice set* is selected based on the induced graph as the set of winners. According to the representation in [11], we apply majority rule between each pair of neighbours and obtain the following majority induced graph:

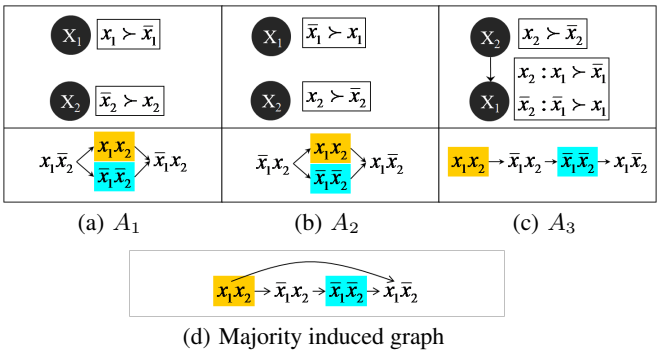
DEFINITION 4 (MAJORITY INDUCED GRAPH). *Given a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, the majority induced graph, denoted by $\mathcal{G} = (O, E)$, is defined by the following edges between alternatives. For each variable X_i, any two alternatives o, $o' \in O$ that differ only on the value of X_i, let there be a directed edge $o \to o'$ if a majority of agents prefer o to o'; there be a directed edge $o' \to o$ if a majority of agents prefer o' to o. If o and o' are majority-incomparable, $\mathcal{G}$ does not contain any edge between o and o'.*

For any two alternative o, $o' \in O$ that differ only on the value of X_i, $o[X_i] = x_i$ and $o'[X_i] = \bar{x}_i$, let $\mathbf{W} = \mathbf{V} - \{X_i\}$ and $\mathbf{w} = o[\mathbf{W}] (= o'[\mathbf{W}])$. Whether or not there is a directed edge $o \to o'$ (resp. $o' \to o$) can be computed directly from the conditional preference table $CPT_j(X_i)$ of each agent A_j's CP-net $\mathcal{N}_j$. Because for each agent A_j, $\mathcal{N}_j \models o \succ o'$ (resp. $\mathcal{N}_j \models o' \succ o$) if and only if $x_i \succ_{A_j}^{X_i|\mathbf{w}} \bar{x}_i$ (resp. $\bar{x}_i \succ_{A_j}^{X_i|\mathbf{w}} x_i$). Note that a pair of neighbours o and o' are incomparable if and only if the number of agents is even and the number of agents who prefer o to o' is equal to the number of agents who prefer o' to o.

The dominance relations in $\mathcal{G}$ are then induced by the directed paths between alternatives [11]:

DEFINITION 5 (GRAPH DOMINANCE). *Given a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, let $\mathcal{G} = (O, E)$ be the majority induced graph. For any $o, o' \in O$, we say that o dominates o' in $\mathcal{G}$, denoted by $o \succ_{\mathcal{G}} o'$ if and only if: i) there is a directed path from o to o', and ii) there is no directed path from o' to o.*

According to Xia *et al.* [11], the *transitive closure* $\succeq_{\mathcal{G}}$ of E specifies the minimum preorder such that if there is a directed path from o to o' in $\mathcal{G}$ then $o \succeq_{\mathcal{G}} o'$. $\succ_{\mathcal{G}}$ is the strict order induced by $\succeq_{\mathcal{G}}$: $o \succ_{\mathcal{G}} o'$ if and only if $o \succeq_{\mathcal{G}} o'$ and $o' \not\succeq_{\mathcal{G}} o$. Based on the induced graph, a choice set function is then defined, which always chooses the following alternatives as winners.



(a) A_1 (b) A_2 (c) A_3

(d) Majority induced graph

Figure 1: Illustration for Proposition 1

DEFINITION 6 (GRAPH WINNER). *Let $\mathcal{G} = (O, E)$ be the majority induced graph for a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, we say,*

- *an alternative is a* global Condorcet winner *(GCW), if it dominates all other alternatives in $\mathcal{G}$;*
- *an alternative is a* local Condorcet winner *(LCW), if it dominates all its neighbours in $\mathcal{G}$;*
- *an alternative is a* weak local Condorcet winner *(wLCW), if it dominates or is incomparable to all its neighbours in $\mathcal{G}$.*

When the global Condorcet winner (GCW) exists, it is unique. A GCW is also a local Condorcet winner (LCW), while the reverse does not hold: a LCW is not necessarily a GCW. Similarily, a LCW is also a weak local Condorcet winner (wLCW), while a wLCW is not necessarily a LCW.

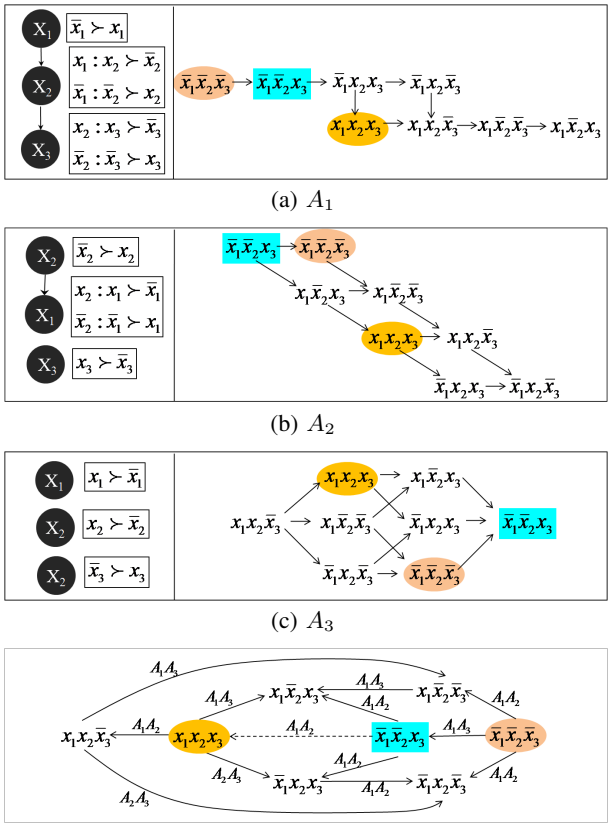
However, we emphise here that GCW, LCW and wLCW in $\mathcal{G}$ are different from the meaning of a (weak) Condorcet winner (Definition 2 and 3), which refers to a majority winner in pair-wise election. In the following section, we analyze the relation between the preferences derived from a majority induced graph $\mathcal{G}$ and the original majority preferences among the agents.

PROPOSITION 1. *Majority-domination $\succ_{maj}$ does not follow from graph domination $\succ_{\mathcal{G}}$.*

PROOF. To prove this proposition, we need to prove that given a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, the majority induced graph $\mathcal{G} = (O, E)$ and a pair of alternatives $o, o' \in O$ and $o \neq o'$, it may be the case that $o \succ_{\mathcal{G}} o'$ but $o \not\succ_{maj} o'$. Consider an example of 3 agents making decision over 2 binary domain variables. The agents' CP-nets, their partial order over the alternative space and the majority induced graph are depicted in Figure 1. According to the majority induced graph (see Figure 1(d)), there is a directed path from outcome x_1x_2 to $\bar{x}_1\bar{x}_2$ and no directed path from $\bar{x}_1\bar{x}_2$ to x_1x_2, i.e. $x_1x_2 \succ_{\mathcal{G}} \bar{x}_1\bar{x}_2$. However, for both A_1 and A_2, these two alternatives are incomparable (see Figure 1(a) and Figure 1(b)), and thus, $\bar{x}_1\bar{x}_2$ and x_1x_2 are majority-incomparable, i.e. $\bar{x}_1\bar{x}_2 \bowtie_{maj} x_1x_2$. Consequently, in this example, $x_1x_2 \succ_{\mathcal{G}} \bar{x}_1\bar{x}_2$ but $x_1x_2 \not\succ_{maj} \bar{x}_1\bar{x}_2$. □

PROPOSITION 2. *The preference relation $\succ_{\mathcal{G}}$ derived from the majority induced graph does not preserve the strict majority preference relation $\succ_{maj}$.*

PROOF. To prove this proposition, we need to prove that given a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, the majority induced



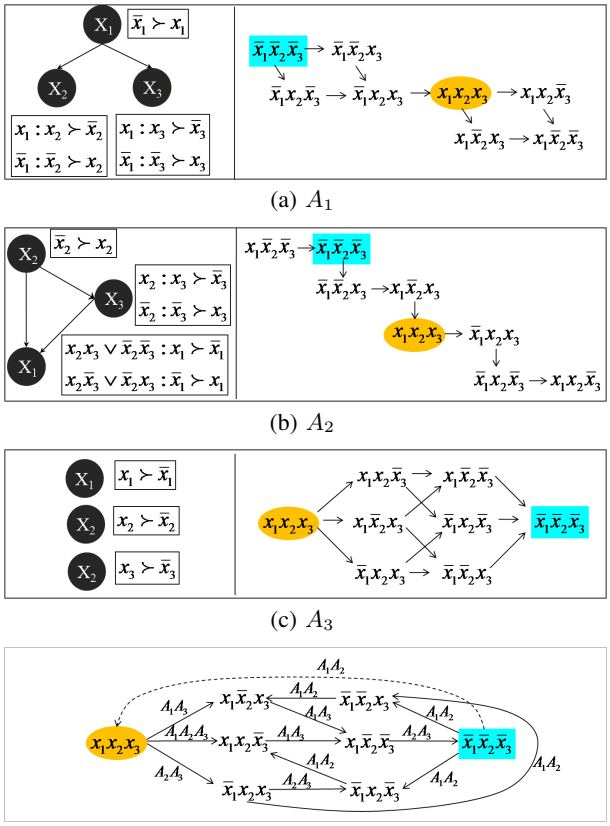
(a) A_1

(b) A_2

(c) A_3

(d) Majority induced graph

Figure 2: Illustration for Propositions 2 and 3



(a) A_1

(b) A_2

(c) A_3

(d) Majority induced graph

Figure 3: Illustration for Proposition 4

graph $\mathcal{G} = (O, E)$ and a pair of alternatives $o, o' \in O$ and $o \neq o'$, it may be the case that $o \succ_{maj} o'$ but $o \not\succ_{\mathcal{G}} o'$. Consider the agents' CP-nets, their partial order over the alternative space and the corresponding majority induced graph depicted in Figure 2. According to the majority induced graph Figure. 2(d), there is no directed path from alternative $\bar{x}_1\bar{x}_2x_3$ to alternative $x_1x_2x_3$, i.e. $\bar{x}_1\bar{x}_2x_3 \not\succ_{\mathcal{G}} x_1x_2x_3$. However, both A_1 and A_2 preferred $\bar{x}_1\bar{x}_2x_3$ to $x_1x_2x_3$ (see Figure 2(a) and Figure 2(b)), and thus $\bar{x}_1\bar{x}_2x_3 \succ_{maj} x_1x_2x_3$. Consequently, in this example, $\bar{x}_1\bar{x}_2x_3 \succ_{maj} x_1x_2x_3$ but $\bar{x}_1\bar{x}_2x_3 \not\succ_{\mathcal{G}} x_1x_2x_3$. □

As $\succ_{\mathcal{G}}$ does not preserve the strict majority preference $\succ_{maj}$, a (weak) local Condorcet winner that dominates or is incomparable to all it neighbours may still be majority-dominated by some alternative, and thus is not guaranteed to be a weak Condorcet winner.

COROLLARY 1. *A (weak) local Condorcet winner is not necessarily a weak Condorcet winner.*

Consider the example in Figure 2. Alternative $x_1x_2x_3$ is a LCW as it dominates all its neighbours ($x_1x_2\bar{x}_3$, $x_1\bar{x}_2x_3$ and $\bar{x}_1x_2x_3$) in the majority induced graph (see Figure 2(d)). However, it is majority-dominated by another alternative $\bar{x}_1\bar{x}_2x_3$ because both A_1 (Figure 2(a)) and A_2 (Figure 2(b)) preferred $\bar{x}_1\bar{x}_2x_3$ to $x_1x_2x_3$ and thus is not a (weak) Condorcet winner.

Now we are interested in whether or not the (weak) local Condorcet winners set is guaranteed to be a non-majority-dominated set, i.e. the alternatives in this set can only be majority-dominated by some alternative in this set but not by any other alternatives outside this set. Unfortunately, the following proposition gives a negative answer to this question.

PROPOSITION 3. *A (weak) local Condorcet winner can be majority-dominated by an alternative outside the set of (weak) local Condorcet winners, even though it is not majority-dominated by any other (weak) local Condorcet winner.*

PROOF. Consider the example in Figure 2, there are only two LCWs $x_1x_2x_3$ and $\bar{x}_1\bar{x}_2\bar{x}_3$ and $x_1x_2x_3 \bowtie_{maj} \bar{x}_1\bar{x}_2\bar{x}_3$: they are incomparable for both A_2 (see Figure 2(b)) and A_3 (see Figure 2(c)). However, as we mentioned before, $x_1x_2x_3$ is majority-dominated by alternative $\bar{x}_1\bar{x}_2x_3$, which is not a LCW or wLCW. □

Finally, we are interested in the following question: whether a global Condorcet winner that dominates every other alternative in the majority induced graph, is guaranteed to be non-majority-dominated, i.e. a (weak) Condorcet winner.

PROPOSITION 4. *A global Condorcet winner is not necessarily a (weak) Condorcet winner.*

PROOF. Consider the agents' CP-nets, their preference ordering over the alternative space and the corresponding majority induced graph in Figure 3. In this example, there is a unique global Condorcet winner $x_1x_2x_3$ in $\mathcal{G}$: there is a directed path from $x_1x_2x_3$ to every other alternative and no incoming edges to $x_1x_2x_3$ (see Figure 3(d)). However, this global Condorcet winner $x_1x_2x_3$ is majority-dominated by $\bar{x}_1\bar{x}_2\bar{x}_3$ ($\bar{x}_1\bar{x}_2\bar{x}_3 \succ_{maj} x_1x_2x_3$), because two agents ($A_1$ and A_2) prefer $\bar{x}_1\bar{x}_2\bar{x}_3$ to $x_1x_2x_3$ (see Figure 3(a) and Figure 3(b)). □

Proposition 4 further shows that the strict preference relation $\succ_{\mathcal{G}}$ derived from the majority induced graph might be conflicting with the original majority preference relation $\succ_{maj}$. For instance, for the example in Figure 3, $x_1x_2x_3 \succ_{\mathcal{G}} \bar{x}_1\bar{x}_2\bar{x}_3$, however, $x_1x_2x_3 \prec_{maj} \bar{x}_1\bar{x}_2\bar{x}_3$.

From the above, it become clear that the majority induced graph may not always represent the majority preferences properly. In particular, a winners in the majority induced graph, i.e., a GCW, LCW or wLCW winner is not necessarily a (weak) Condorcet winner. However, we observe that a (weak) Condorcet winner must be a wLCW.

THEOREM 1. *Let $\mathcal{G} = (O, E)$ be the majority induced graph for a collection of CP-nets $\boldsymbol{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$. Then a (weak) Condorcet winner is also a weak local Condorcet winner in $\mathcal{G}$.*

PROOF. Suppose a (weak) Condorcet winner o is not a wLCW, then there exist at least one neighbour o' in $\mathcal{G}$ such that $o' \rightarrow o \in E$. That means, there is a majority of agents prefers o' to o ($o' \succ_{maj} o$), contradicting the fact that o is a (weak) Condorcet winner. Thus, a (weak) Condorcet winner must also be a wLCW. □

A (weak) Condorcet winner must be a wLCW, while the reverse does not hold: a wLCW is not necessarily a (weak) Condorcet winner. Nonetheless, this relation provides us a more efficient way to compute the majority winners among a large alternative space: first compute the set of wLCWs, and then compute the set of majority winners by filtering out those that are majority-dominated by some alternative.

REMARK. Here the notions of majority induced graph coincides with the definitions in [11] when the number of agents is odd, and differ only in the presence of incomparability between neighbours when the number of agents is even. According to [11], when a pair of neighbours o and o' are majority-incomparable, $\mathcal{G}$ contains directed edges both from o to o' and o' to o. However, their definition may exclude the weak Condorcet winners when the number of agents is even. For instance, if a weak Condorcet winner is majority-incomparable to one of its neighbours, then it is not considered to be a wLCW (nor a GCW or LCW) according to their definition.

4. COMPUTE THE (WEAK) CONDORCET WINNER

In this section, we present our proposed approach, `MajCP` (majority-rule-based collective decision-making with CP-nets), for computing the majority winning alternatives. The proposed approach includes the following two steps. First, we compute the set of wLCWs via a reduction to an extended SAT (Boolean satisfiability problem) for cardinality constraints (See Algorithm 1). Then, in the second step, the set of (weak) Condorcet winners can be obtained by filtering out those that are majority-dominated by some alternative.

Assume m agents $\mathbf{A} = \{A_1, \ldots, A_m\}$ are making decisions over a set of n variables $\mathbf{V} = \{X_1, \ldots, X_n\}$. The preference of each agent A_j is captured by a (possibly cyclic) binary-valued CP-net $\mathcal{N}_j$ and let $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$. We first reduce the problem of computing the set of wLCWs into a corresponding SAT problem. The variables in our reduction consist of the variables in the agents' CP-nets. Firstly, we generate a set of optimality constraints that a wLCW must satisfy according to majority rule. For each variable X_i, each agent A_j's has a conditional preference table $CPT_j(X_i)$ stating the conditional preference on the values of variable X_i with each instantiation of X_i's parents $Pa_j(X_i)$. We separate these condition entries in $CPT_j(X_i)$ into the following two categories.

- The set of parent context in which agent A_j prefers x_i to $\bar{x}_i$: $\mathbf{U}_{A_j}^{x_i \succ \bar{x}_i} = \left\{\mathbf{u} \in D(Pa_j(X_i)) \mid x_i \succ_{A_j}^{X_i|\mathbf{u}} \bar{x}_i\right\}$.
- The set of parent context in which agent A_j prefers $\bar{x}_i$ to x_i: $\mathbf{U}_{A_j}^{\bar{x}_i \succ x_i} = \left\{\mathbf{u} \in D(Pa_j(X_i)) \mid \bar{x}_i \succ_{A_j}^{X_i|\mathbf{u}} x_i\right\}$.

Let $P_j^i = \bigvee_{\mathbf{u} \in \mathbf{U}_{A_j}^{x_i \succ \bar{x}_i}} \mathbf{u}$ (resp. $\bar{P}_j^i = \bigvee_{\mathbf{u} \in \mathbf{U}_{A_j}^{\bar{x}_i \succ x_i}} \mathbf{u}$), i.e., the disjunction of the condition part of the entry whose conclusion is $x_i \succ \bar{x}_i$ (resp. $\bar{x}_i \succ x_i$) in the $CPT_j(X_i)$ of agent A_j (line 14–20). Note that if agent A_j has unconditional preference over a variable X_i, $Pa_j(X_i) = \emptyset$ and $x_i \succ_{A_j}^{X_i} \bar{x}_i$ (resp. $\bar{x}_i \succ_{A_j}^{X_i} x_i$), that means the condition P_j^i (resp. $\bar{P}_j^i$) is always $True$ and $\bar{P}_j^i$ (resp. P_j^i) is always $False$ (line 7–11). Thus, $x_i \succ_{A_j}^{X_i|P_j^i} \bar{x}_i$ (resp. $\bar{x}_i \succ_{A_j}^{X_i|\bar{P}_j^i} x_i$).

For each individual agent A_j, $\mathbf{U}_{A_j}^{x_i \succ \bar{x}_i}$ and $\mathbf{U}_{A_j}^{\bar{x}_i \succ x_i}$ are complementary, and thus $P_j^i = \neg\bar{P}_j^i$ (resp. $\bar{P}_j^i = \neg P_j^i$). For any setting $\mathbf{w} = D(\mathbf{W})$ ($\mathbf{W} = \mathbf{V} - \{X_i\}$) that satisfies P_j^i (resp. $\bar{P}_j^i$), then $x_i\mathbf{w} \succ_{A_j} \bar{x}_i\mathbf{w}$ (resp. $\bar{x}_i\mathbf{w} \succ_{A_j} x_i\mathbf{w}$).

Given a directed graph $\mathcal{G} = (O, E)$, for any two alternatives $o, o' \in O$ that differ only on the value of X_i: $o[X_i] = x_i$ and $o'[X_i] = \bar{x}_i$. Let $q = (m+1)/2$ (m is the total number of agents) (line 1). There is an directed edge $o \rightarrow o'$ (resp. $o' \rightarrow o$) in $\mathcal{G}$ if and only if, for the setting $\mathbf{w} = o[\mathbf{W}] (= o'[\mathbf{W}])$ and $\mathbf{W} = \mathbf{V} - \{X_i\}$, there exist a set of at least q agents, denoted by $\mathbf{S}$ ($\mathbf{S} \subseteq \mathbf{N}$), each agent $A_j \in \mathbf{S}$ has the following conditional (unconditional) preference $x_i \succ_{A_j}^{X_i|\mathbf{w}} \bar{x}_i$ (resp. $\bar{x}_i \succ_{A_j}^{X_i|\mathbf{w}} x_i$), i.e., $\mathbf{w}$ satisfies $\bigwedge_{A_j \in \mathbf{S}} P_j^i$ (resp. $\bigwedge_{A_j \in \mathbf{S}} \bar{P}_j^i$). Furthermore, there will be a set of $\binom{m}{q}$ distinct q-subsets of agents that satisfies this majority requirement, denoted by $\mathbf{C}$. Consequently, if the setting $\mathbf{w}$ satisfies $\bigvee_{\mathbf{S} \in \mathbf{C}} \left(\bigwedge_{A_j \in \mathbf{S}} P_j^i \right)$ (resp. $\bigvee_{\mathbf{S} \in \mathbf{C}} \left(\bigwedge_{A_j \in \mathbf{S}} \bar{P}_j^i \right)$), then there is an directed edge $o \rightarrow o'$ (resp. $o' \rightarrow o$), and thus $o \succ_{\mathcal{G}} o'$ (resp. $o' \succ_{\mathcal{G}} o$). For the purpose of explanation, we reason directly with cardinality formulas, which has been widely explored in CSPs and SAT (cardinality constraints), see e.g., [2] and [7]. For each variable X_i, let F_i and F_i' be the following cardinality formula respectively (line 24):

$$F_i = [\geq q]\left(P_1^i, \ldots, P_m^i\right) \tag{1}$$

$$F_i' = [\geq q]\left(\bar{P}_1^i, \ldots, \bar{P}_m^i\right) \tag{2}$$

Such that F_i (resp. F_i') is $True$ when at least q formulas among $P_1^i, \ldots, P_m^i$ (resp. $\bar{P}_1^i, \ldots, \bar{P}_m^i$) are $True$. Note that the cardinality formula F_i (resp. F_i') is logically equivalent to the classical propositional formula $\bigvee_{\mathbf{S} \in \mathbf{C}} \left(\bigwedge_{A_j \in \mathbf{S}} P_j^i \right)$ (resp. $\bigvee_{\mathbf{S} \in \mathbf{C}} \left(\bigwedge_{A_j \in \mathbf{S}} \bar{P}_j^i \right)$).

Given an directed graph $\mathcal{G} = (O, E)$, let o, $o' \in O$ be two alternatives that differ only on the value of a variable X_i, $o[X_i] = x_i$ and $o'[X_i] = \bar{x}_i$. Let $\mathbf{w} = o[\mathbf{W}] (= o'[\mathbf{W}])$ and $\mathbf{W} = \mathbf{V} - \{X_i\}$. If the setting $\mathbf{w}$ satisfies F_i (resp. F_i'), then there is an directed edge $o \rightarrow o'$ (resp. $o' \rightarrow o$). Consequently, the wLCWs must satisfy the following optimality constraints for each variable X_i (line 25).

DEFINITION 7 (OPTIMALITY CONSTRAINTS). *Given a collection of CP-nets $\boldsymbol{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, for each variable X_i, the majority-optimality constraint φ_i to the value of X_i is:*

$$\varphi_i = (F_i \Rightarrow x_i) \wedge (F_i' \Rightarrow \bar{x}_i) \tag{3}$$

Algorithm 1: MajCP

Input: **N**, a set of CP-nets of the agents;
Output: CW, a set of (weak) Condorcet winners

1 $q \leftarrow (m+1)/2$ where m is the total number of agents;
2 $\varphi \leftarrow True$;
3 **foreach** $X_i \in \boldsymbol{V}$ **do**
4 $list, list' \leftarrow \emptyset$;
5 **foreach** $\mathcal{N}_j \in \boldsymbol{N}$ **do**
6 **if** $Pa_j(X_i) = \emptyset$ **then**
7 **if** $x_i \succ_{A_j}^{X_i} \bar{x}_i$ **then**
8 $P_j^i \leftarrow True$; $\bar{P}_j^i \leftarrow False$;
9 **else**
10 $\bar{P}_j^i \leftarrow True$; $P_j^i \leftarrow False$;
11 **end**
12 **else**
13 $P_j^i \leftarrow False$; $\bar{P}_j^i \leftarrow False$;
14 **foreach** *cp-statement* $\in CPT_j(X_i)$ **do**
15 **if** $\mathbf{u} \in \mathbf{U}_{A_j}^{x_i \succ \bar{x}_i}$ **then**
16 $P_j^i \leftarrow P_j^i \vee \mathbf{u}$
17 **else**
18 $\bar{P}_j^i \leftarrow \bar{P}_j^i \vee \mathbf{u}$
19 **end**
20 **end**
21 **end**
22 **add** P_j^i to $list$; **add** $\bar{P}_j^i$ to $list'$;
23 **end**
24 $F_i \leftarrow [\geq q]\, list$; $F_i' \leftarrow [\geq q]\, list'$;
25 $\varphi_i \leftarrow (F_i \Rightarrow x_i) \wedge (F_i' \Rightarrow \bar{x}_i)$;
26 $\varphi \leftarrow \varphi \wedge \varphi_i$
27 **end**
28 $graphWinners \leftarrow$ the models of φ;
29 $CW \leftarrow$ `optimalityCheck(`$graphWinners$`)`;
30 **return** CW;

Note that if there is an odd number of agents, $F_i' = \neg F_i$ and the above constraint φ_i can be simplified to:

$$\varphi_i = (F_i \Leftrightarrow x_i)$$

Finally, let φ be the conjunction of all φ_i (one for each variable) (line 26):

$$\varphi = \bigwedge_{X_i \in \mathbf{V}} \varphi_i \tag{4}$$

THEOREM 2. *Let $\mathcal{G} = (O, E)$ be the majority induced graph for a collection of CP-nets $\boldsymbol{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$. An alternative o is a weak local Condorcet winner if and only if it satisfies the above SAT φ.*

PROOF. (Soundness) Let o be an alternative that satisfies φ. For every neighbour o' of o that differs on the value of a single variable $X_i \in \mathbf{V}$, as o satisfies $\varphi_i = (F_i \Rightarrow x_i) \wedge (F_i' \Rightarrow \bar{x}_i)$, then either there is an directed edge $o \rightarrow o'$ or there is no edge between o and o'. According to Definition 6, o is a wLCW.
(Completeness) Assume first that there is at least one wLCW o, and suppose that o does not satisfy φ. Then there exists at least one optimality constraint $\varphi_i = (F_i \Rightarrow x_i) \wedge (F_i' \Rightarrow \bar{x}_i)$ that o does not satisfy. As F_i and F_i' are mutually exclusive, and for the sake of simplicity we assume that o does not satisfy $F_i \Rightarrow x_i$. An implication is unsatisfied only when the hypothesis is $True$ and the conclusion is $False$. That is, o satisfies F_i yet $o[X_i] = \bar{x}_i$. Let o' be a neighbour of o, $o[\mathbf{W}] = o'[\mathbf{W}]$ ($\mathbf{W} = \mathbf{V} - \{X_i\}$) and $o'[X_i] = x_i$. Then, o' satisfy $F_i \Rightarrow x_i$. There must be an edge $o' \rightarrow o$ in $\mathcal{G}$ and thus $o' \succ_{\mathcal{G}} o$, contradicting the fact that o is a wLCW. Hence, the above SAT φ must be satisfied by all the alternatives that are wLCWs. □

As such, we reduce the problem of computing wLCWs into a SAT problem and the set of wLCWs can be obtained by computing the models of the corresponding SAT (line 28). Recall that a wLCW is not necessarily a weak Condorcet winner. In the second step, we need to test the majority optimality of each wLCW (i.e. a model of the corresponding SAT) by comparing it to all other alternatives and filtering out those that are majority-dominated by some alternative (line 29).

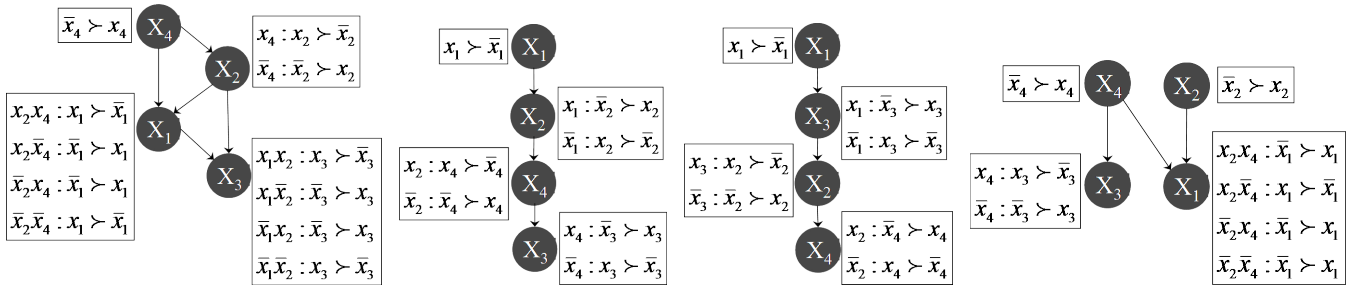
THEOREM 3 (COMPLEXITY). *Given a collection of m CP-nets $\boldsymbol{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, if $\forall \mathcal{N}_j \in \boldsymbol{N}$, the node in-degree is bounded by a constant, then translating the problem of computing weak local Condorcet winners into a corresponding extended SAT problem for cardinality constraints is polynomial.*

PROOF. Assume there are n variables and the number of parents of a node in the dependency graph of each agent is bounded by a constant d. In order to translate the problem of computing wLCWs into the corresponding SAT problem φ, we need to generate a majority-optimality constraint φ_i for each variable X_i. For each variable X_i, we need to check each $\mathcal{N}_j$'s conditional preference table $CPT_j(X_i)$. The number of cp-statements in $CPT_j(X_i)$ is exponential in the number of parents of X_i in the dependency graph of a $\mathcal{N}_j$. Since we assume that node in-degree is bounded by a constant d, the exponential is still a constant (i.e. 2^d) and the number of variables included in the condition entry of every cp-statement is also bounded by d. Thus, the running time of translation is $O(n \cdot m \cdot 2^d \cdot d)$. □

THEOREM 4 (COMPLEXITY). *Given a collection of m CP-nets $\boldsymbol{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, if $\forall \mathcal{N}_j \in \boldsymbol{N}$, the node in-degree is bounded by a constant, then i) checking whether an alternative is a weak local Condorcet winner is polynomial; and, ii) finding the set of weak local Condorcet winners is NP-complete.*

PROOF. Based on Theorem 2, to check whether an alternative o is a wLCW we just need to check whether o is a model of the corresponding extended SAT problem φ, that is, whether o satisfies the optimality constraint $\varphi_i = (F_i \Rightarrow x_i) \wedge (F_i' \Rightarrow \bar{x}_i)$ of each variable X_i. The constraint $F_i \Rightarrow x_i$ (resp. $F_i' \Rightarrow \bar{x}_i$) is satisfied if and only if the condition F_i (resp. F_i') is $False$ or the conclusion x_i (resp. $\bar{x}_i$) is $True$. For instance, if o assigns $\bar{x}_i$ to X_i, o satisfies $F_i' \Rightarrow \bar{x}_i$. Thus, o satisfies φ_i if and only if o also satisfies $F_i \Rightarrow x_i$. Also, as $o[X_i] = \bar{x}_i$, o satisfies $F_i \Rightarrow x_i$ if and only if F_i is evaluated to $False$. Checking the truth value of F_i can be done by counting the elements in the list of F_i that is evaluated to $True$: if there are fewer than $(m+1)/2$ formulas are evaluated to $True$ then F_i is evaluated to $False$. Suppose there are n variables and node in-degree is bounded by a constant d. Then there are m formulas listed in F_i and each formula is a disjunction of at most 2^d conjunctions of at most d literals. Consequently, the running time of checking whether an alternative is a model of φ is thus $O(n \cdot m \cdot 2^d \cdot d)$.

Regarding the problem of finding the set of wLCWs. As we already show that testing whether an alternative is a wLCW (i.e. is a model of φ) is polynomial, the problem of finding the set of



(a) CP-net of A_1 (b) CP-net of A_2 (c) CP-net of A_3 (d) CP-net of A_4

Figure 4: CP-nets of the agents

wLCWs (i.e. the models of φ) is in NP. To show hardness, we reduce 3-SAT to our problem: given a 3-CNF formula F, for each clause $(x_1 \vee x_2 \vee x_3) \in F$, we construct the optimality constraint: $\geq 2\,[\bar{x}_1, \bar{x}_2] \Rightarrow x_3$. Any satisfying assignment of the original 3-CNF formula, at lease one of x_1, x_2 and x_3 is true. If x_1 or x_2 are $True$, then the condition of the optimality constraint is not satisfied and thus the optimality constraint is satisfied. If x_1 and x_2 are both $False$, then x_3 is $True$, which satisfies the optimality constraint as this is the preferred value of a majority of agents. Hence, any model of the original 3-CNF formula is an optimal assignment of the set of optimality constraints. The argument reverses: any wLCW is also a model. □

We emphise here that the above complexity is for testing or finding the wLCWs rather than the (weak) Condorcet winners in Definition 2 and 3. As we show in Section 3 (Corollary 1), a wLCW is not necessarily a weak Condorcet winner. To find out the set of weak Condorcet winners, we still need to filter out from the set of wLCWs those candidates that are majority-dominated by some alternative. This checking is required even when there exists only one wLCW. Consequently, the complexity for finding the set of weak Condorcet winner remains PSPACE complete.

EXAMPLE. Now, we demonstrate the execution of the proposed approach with an example. Assume four agents $\mathbf{A} = \{A_1, A_2, A_3, A_4\}$ making decision over a set of four Boolean variables X_1, X_2, X_3 and X_4.Consider the agents' CP-nets depicted in Figure 4. We first generate a set of majority-optimality constraints that a wLCW must satisfy. For variable X_1, we refer to each agent A_j's conditional preference table $CPT_j(X_1)$:
A_1: $\mathbf{U}_{A_1}^{x_1 \succ \bar{x}_1} = \{x_2x_4, \bar{x}_2\bar{x}_4\}$ and $\mathbf{U}_{A_1}^{\bar{x}_1 \succ x_1} = \{x_2\bar{x}_4, \bar{x}_2x_4\}$, thus $P_1^1 = x_2x_4 \vee \bar{x}_2\bar{x}_4$ and $\bar{P}_1^1 = x_2\bar{x}_4 \vee \bar{x}_2x_4$;
A_2: the preference over variable X_1 is unconditional, $x_1 \succ_{A_2}^{X_1} \bar{x}_1$, thus $P_2^1 = True$ and $\bar{P}_2^1 = False$;
A_3: the preference over variable X_1 is unconditional, $x_1 \succ_{A_3}^{X_1} \bar{x}_1$, thus $P_3^1 = True$ and $\bar{P}_3^1 = False$;
A_4: $\mathbf{U}_{A_4}^{x_1 \succ \bar{x}_1} = \{x_2\bar{x}_4\}$ and $\mathbf{U}_{A_4}^{\bar{x}_1 \succ x_1} = \{x_2x_4, \bar{x}_2x_4, \bar{x}_2\bar{x}_4\}$, thus $P_4^1 = x_2\bar{x}_4$ and $\bar{P}_4^1 = x_2x_4 \vee \bar{x}_2x_4 \vee \bar{x}_2\bar{x}_4$.

Consequently, $F_1 = [\geq 3]\left(P_1^1, P_2^1, P_3^1, P_4^1\right)$ and $F_1' = [\geq 3]\left(\bar{P}_1^1, \bar{P}_2^1, \bar{P}_3^1, \bar{P}_4^1\right)$. F_1 can be simplified to $(x_2 \vee \bar{x}_4)$. F_1' is unsatisfiable and evaluated to $False$, because two formulas $\bar{P}_2^1$ and $\bar{P}_3^1$ out of four in the formula list of F_i' are $False$. Hence, the winning alternative must satisfy the following optimality constraint for variable X_1: $\varphi_1 = (x_2 \vee \bar{x}_4 \Rightarrow x_1) \wedge (False \Rightarrow \bar{x}_1)$. An implication is unsatisfied only when the hypothesis is $True$ and the conclusion is $False$, thus $False \Rightarrow \bar{x}_1$ is always $True$ and φ_1 can be simplified to $\varphi_1 = x_2 \vee \bar{x}_4 \Rightarrow x_1$.

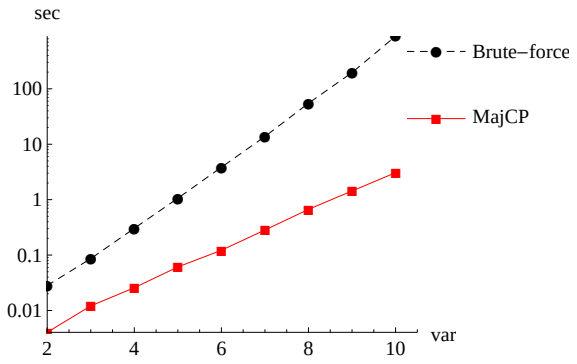
Similarly, we obtained the following optimality constraints (simplified form of the cardinality constraints) for variable X_2, X_3 and X_4:
X_2: $\varphi_2 = (\bar{x}_1x_3x_4 \Rightarrow x_2) \wedge (x_1\bar{x}_3 \vee x_1\bar{x}_4 \vee \bar{x}_3\bar{x}_4 \Rightarrow \bar{x}_2)$;
X_3: $\varphi_3 = (\bar{x}_1\bar{x}_2 \Rightarrow x_3) \wedge (x_1\bar{x}_2 \Rightarrow \bar{x}_3)$;
X_3: $\varphi_4 = \bar{x}_4$;
Consequently, we obtain the following SAT:
$\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 = (x_2 \vee \bar{x}_4 \Rightarrow x_1) \wedge (\bar{x}_1x_3x_4 \Rightarrow x_2) \wedge (x_1\bar{x}_3 \vee x_1\bar{x}_4 \vee \bar{x}_3\bar{x}_4 \Rightarrow \bar{x}_2) \wedge (\bar{x}_1\bar{x}_2 \Rightarrow x_3) \wedge (x_1\bar{x}_2 \Rightarrow \bar{x}_3) \wedge \bar{x}_4$

The above SAT has only one satisfied assignment $x_1\bar{x}_2\bar{x}_3\bar{x}_4$. After checking the majority optimality of $x_1\bar{x}_2\bar{x}_3\bar{x}_4$, it is also a weak Condorcet winner in this example.

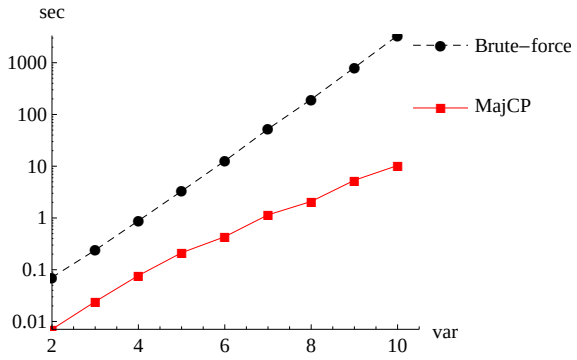
5. EXPERIMENT

In this section, we present the experimental results regarding the execution time of the proposed approach. We compare the performance of the proposed `MajCP` approach to a `Brute-force` algorithm, which runs a direct election over the alternative space. In these experiments, the numbers of agents are 5 and 15 respectively, and we vary the numbers of variables from 2 to 10. The number of parents of a variable in the agents' CP-nets is bounded by 6. For each number of agents and each number of variables, we generate 5,000 random examples of the agents' CP-nets.

The log-scale plots in Figure 5 show the average execution times of the `Brute-force` algorithm and the proposed `MajCP` approach in the case of 5 agents and 15 agents, respectively. It demonstrates that the proposed `MajCP` approach is much more efficient than the `Brute-force` algorithm. In general, for large numbers of variables, it offers several orders of magnitude improvement in performance over the `Brute-force` algorithm both for 5 agents and 15 agents. For instance, when there are 10 variables, the execution time of `MajCP` is reduced by more than three orders of magnitude as compared to `Brute-force` algorithm. We further test 100 cases for 11 variables and 5 agents (resp. 11 variables and 15 agents), which shows that the execution time of the `Brute-force` algorithm is on average more than 5000 seconds (resp. 9000 seconds). On the other hand, the proposed `MajCP` approach can produce the majority winners in about 10 seconds (resp. 15 seconds). Note that when there exist wLCWs (1 or more), the proposed `MajCP` approach still need to test the majority-optimality of the wLCWs by comparing each wLCW to all other alternatives. However, when there are no wLCWs, the proposed approach can return the result quickly by only solving the corresponding SAT problem. For instance, given 15 agents and 10 variables, when there does not exists any wLCWs, the proposed approach returns the results within 0.04 seconds. Table 1 provides the probability that there exists no wLCWs for the given agents' preferences in



(a) 5 agents



(b) 15 agents

Figure 5: Average execution time comparison (Log scale plot)

those experiments.

Table 1: The percentage of cases when there are no wLCWs

Agents	Variables				
	2	4	6	8	10
5	4.71%	12.73%	19.67%	22.11%	24.51%
15	5.44 %	15.99%	22.22%	25.42%	27.94%

6. CONCLUSION AND FUTURE WORK

In this paper, we have introduced an efficient approach to compute the set of winning alternatives from a collection of CP-nets based on majority rule. Unlike previous work where the agents' preferences are required to satisfy some restrictive conditions on the dependence graph (such as the existence of a common acyclic graph to all the agents), the proposed approach allows the agents to have different preferential independence structures and also works on cyclic CP-nets. It first computes a set of weak local Condorcet winners (wLCWs) by reduces the problem into an extended SAT (Boolean satisfiability problem) for cardinality constraints. Then the set of majority winning alternatives is a subset of wLCWs after filtering out those are majority-dominated by some alternative. The proposed approach reduces the size of search space and is computationally efficient.

Future research can extend the proposed approach to compute the winners of other aggregation rules. Another extension would be to investigate techniques to aggregate preferences that are represented by more powerful variants such as TCP-nets and UCP-nets.

7. ACKNOWLEDGEMENTS

We thank Jérôme Lang, Lirong Xia, Toby Walsh, Dongmo Zhang and anonymous reviewers for helpful discussions and comments. This work is partially supported by the ARC Discovery Grant DP0987380.

8. REFERENCES

[1] K. J. Arrow. *Social choice and individual values*. Wiley, New York :, 1951.

[2] B. Benhamou, L. Sais, and P. Siegel. Two proof procedures for a cardinality based language in propositional calculus. In *STACS '94: Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 71–82, 1994.

[3] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[4] C. Domshlak and R. I. Brafman. CP-nets - reasoning and consistency testing. In *KR*, pages 121–132, 2002.

[5] C. Domshlak, S. D. Prestwich, F. Rossi, K. B. Venable, and T. Walsh. Hard and soft constraints for reasoning about qualitative conditional preferences. *J. Heuristics*, 12(4-5):263–285, 2006.

[6] J. Goldsmith, J. Lang, M. Truszczynski, and N. Wilson. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research*, 33(1):403–432, 2008.

[7] P. V. Hentenryck and Y. Deville. The cardinality operator: A new logical connective for constraint logic programming. In *ICLP*, pages 745–759, 1991.

[8] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.

[9] P. K. Pattanaik. *Voting and collective choice; some aspects of the theory of group decision-making*. 1971.

[10] F. Rossi, K. B. Venable, and T. Walsh. mCP nets: Representing and reasoning with preferences of multiple agents. In *AAAI*, pages 729–734, 2004.

[11] L. Xia, V. Conitzer, and J. Lang. Voting on multiattribute domains with cyclic preferential dependencies. In *AAAI*, pages 202–207, 2008.

[12] L. Xia, J. Lang, and M. Ying. Strongly decomposable voting rules on multiattribute domains. In *AAAI*, pages 776–781, 2007.

Simulation and Emergence

Emerging Cooperation on Complex Networks

Norman Salazar, Juan A. Rodriguez-Aguilar and Josep Ll. Arcos
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
norman,jar,arcos@iiia.csic.es

Ana Peleteiro, Juan C. Burguillo-Rial
Dep. de Ingeniería Telemática
Universidad de Vigo
Vigo, Spain
apeleteiro,jrial@det.uvigo.es

ABSTRACT

The dynamic formation of coalitions is a well-known area of interest in multi-agent systems (MAS). Coalitions can help self-interested agents to successfully cooperate and coordinate in a mutually beneficial manner. Moreover, the organization provided by coalitions is particularly helpful for large-scale MAS. In this paper we present a distributed approach for coalition emergence in large-scale MAS. In particular, we focus on MAS with agents interacting over complex networks since they provide a realistic model of the nowadays interconnected world (e.g. social networks). Our experiments show the effectiveness of our coalition emergence approach in achieving full cooperation over different complex networks. Furthermore, they provide a clear picture of the strong influence the topology has on coalition emergence.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms,Experimentation

Keywords

cooperation, coalition emergence, consensus, MAS

1. INTRODUCTION

Achieving cooperation and coordination in multi-agent systems is a challenging issue [10]. These becomes even more difficult to accomplish when dealing with *self-interested* agents. Cooperation among self-interested agents is often hindered by *social dilemmas* [9]. In these dilemmas, agents must decide between a (short-term) individual benefit or a (long-term) group benefit. Individual decisions (self-interested), besides providing only momentary benefits, are detrimental if many agents take them (e.g. if many individuals try to download the same file at the same time, their download speed suffers greatly). Instead, group decisions (social) can result in a mutually beneficial cooperation that holds over time [17]. In MAS, examples of social dilemmas can be often observed in frequency spectrum assignation, load balancing, packet/message congestion, bandwidth allocation, etc. Therefore, mechanisms that promote the emergence and maintenance of cooperation for self-interested agents is an area of interest [7].

The emergence of cooperation is often studied in the context of the Prisoner's Dilemma (PD) theoretical framework [2]. This has been specially useful for understanding the role of local interactions and the maintenance of cooperation [16, 13, 11]. Moreover, these studies have been successfully applied to existing applications (e.g. Peer-to-Peer (P2P) systems [8]). Nonetheless, in P2P and many other complex systems, the problems relating to social dilemmas still exist.

To prevent social dilemmas and promote cooperation, Axelrod proposed a tribute/tax model [3]. According to this model, cooperation is achieved when agents form *coalitions* around some emerging *leaders*. To maintain their coalitions, leaders charge their agents some tribute/tax. In other words, leaders extort other agents with some pay in favor of a benefit (e.g. guaranteed cooperation, protection against cheaters). This is a clear example of the known tradeoff between the benefits *vs.* the costs of collaboration (e.g. taxes) [18].

Axelrod's model has been successfully adopted to help agents, on grid topologies, cooperate when using a spatial version on the PD [5]. However, whether cooperation is still possible on actual-world topologies via a tribute/tax model, such as the one described by Axelrod [3], remains *unexplored*. Complex networks provide a more realistic model of the topological features found in many nature, social and technological networks (e.g. social networks, the Internet, ecological populations) [1, 19]. Furthermore, it is known that they can influence emergence [15].

The main contribution of this paper is the design of a mechanism to emerge and sustain *full* and *profitable* cooperation, via a *single super-coalition*, but with a low collaboration cost (tax). Specially, since we found that: a) the coalition strategies employed by [5] cannot accomplish full cooperation on complex network topologies; and b) that the notion of tribute (having leader agents setting taxes) is unfair for the population as a whole. Therefore, our proposed approach contributes with: i) a set of coalition strategies that promote a profitable cooperation on complex networks; and ii) a consensus mechanism that allows coalition members themselves (instead of leaders) to reach a convention over the *fair* price to pay to be part of a coalition. Thus, unlike Axelrod's model, agents in our approach are no longer subject to leader extortion. Overall, this results in an ap-

Cite as: Emerging Cooperation on Complex Networks, Salazar, Rodriguez-Aguilar, Arcos, Peleteiro,Burguillo-Rial, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 669-676.

proach fair and profitable for all agents.

Moreover, we show that our approach has a high degree of *resilience* against the leader's failure. This is important, because if a leader fails, its whole coalition collapses, halting the cooperative behavior (i.e. leaders induce a single-point of failure). However, in our approach after the leader fails, agents promptly emerge a new coalition.

The paper is organized as follows. Section 2 briefly describes the base model and presents its evaluation on different complex networks. Next, in section 3 we propose and evaluate both a new set of coalition strategies and our consensus mechanism. Finally, in section 4 we draw some conclusions.

2. A BASE COALITION FRAMEWORK

The purpose of this section is twofold. Firstly, section 2.1 introduces the base mechanism for coalition emergence that we subsequently extend (in section 3) to support cooperation over complex networks. Secondly, in section 2.2 we empirically analyze the performance of the base mechanism over complex networks.

2.1 The Base Approach

In this section we summarize the model for coalition formation that we extend in this paper. The model is thoroughly described in [5], and it is based on Axelrod's model for the emergence of political actors described in [3]. The main motivation of the Axelrod's model in [3] is to promote cooperation by increasing the organization level of a multi-agent system. This is accomplished through the emergence of some leading agents that command coalitions of previously independent agents. Each agent within a coalition cooperates with its leader agent. Moreover, the leader also imposes the strategic behavior to follow against members and non-members of the coalition. Consequently, notice that the emergence of a single coalition guarantees full cooperation between all agents.

The model in [5] considers an agent population using a grid as its interaction topology. The interaction between agents is modeled as an n-person game, i.e. n agents interacting simultaneously, where each game is a spatial version of the Iterated Prisoner's Dilemma (IPD) [13] that takes into account each agent's number of neighbors. Every agent must decide whether to behave as a defector or cooperator during each round of the game, and they are payed according to the payoff matrix depicted in table 1. Therefore, in an attempt to maximize their individual payoffs, agents must also decide whether to join or leave a coalition, or switch to another one. To summarize, the model is composed of: (1) a role model describing the roles each agent may take on (independent, coalition member, and leader); (2) a game-based interaction model describing how agents interact (spatial IPD); (3) a collection of interaction strategies for the roles that agents play; and (4) a collection of *coalition strategies* for the roles that agents play.

First, the role model considers that each agent can play one out of three mutually exclusive roles:

- An *independent* agent decides its own interaction strategy (whether to cooperate or defect) during each game. It decides its next action using a probabilistic Tit-for-Tat (pTFT) strategy [5]. Unlike classical TFT [4], a pTFT strategy stochastically imitates the action played by the majority of an agent's neighbors in a previous round. Additionally, it has coalition strategies to decide whether to join or not a coalition.

		Agent j C	Agent j D
Agent i	C	(3,3)	(0,5)
	D	(5,0)	(1,1)

Table 1: Prisoner's Dilemma Payoff Matrix

- A *coalition member* agent leaves the decisions regarding its interaction strategy to its coalition leader. However, it still has coalition strategies to decide whether to leave the coalition (to either switch to a better one or in favor of independence) or stay in it. Moreover, a coalition member must pay some tax to its leader for the right to remain in the coalition. This tax serves as a guarantee for cooperation within the coalition.

- A *coalition leader* agent decides the interaction strategy for the whole coalition. Leaders impose that all the agents within a coalition cooperate between them, but defect when interacting with agents outside the coalition. A leader cannot disband its coalition. However, it must decide the taxes that its coalition members must pay to remain in the coalition. Notice that by applying a tax percentage to its coalition members, a leader increases its own income. A leader's income depends on: the amount of tax, the number of agents in the coalition, and the income of coalition members. Therefore, although choosing high taxes may lead to more short-term revenues, it may also lead to bankruptcy of coalition members, and hence to the collapse of the coalition (as observed in [3]).

Now we turn our attention to the actual coalition strategies employed by agents to decide whether to join, leave, or switch coalitions. These decisions mainly depend on the agents' payoffs when compared with their neighbors, and on their commitments. The notion of *commitment*, introduced in [3], reinforces cooperation between agents with previous cooperative interactions. In what follows, we abstract the coalition strategies presented in [5] as a collection of qualitative, role-based strategies:

Independent agent decision-making

1. *Join coalition (worst agents).* **If** my payoff is the worst in my neighborhood **then** join my best (payoff-wise) neighbor's coalition (request to form one if needed).

2. *Join coalition (moderate agents).* **If** my payoff is average in my neighborhood **and** I am committed to my best neighbor **then** join its coalition (request to form one if needed).

Coalition member decision-making

3. *Leave coalition (isolated agents).* **If** I am isolated (connection wise) from my coalition **then** leave it.

4. *Strengthen coalition (satisfied agents).* **If** my payoff is good **then** increase my commitment with my leader.

5. *Coalition switch (worst agents).* **If** my payoff is the worst in my neighborhood **and** the agent with the best payoff in my neighborhood is not my leader **then** switch to the best agent coalition.

6. *Coalition switch (unsatisfied agents).* **If** the agent with the best payoff in my neighborhood is not my leader **and** I have some commitment with this best agent **then** switch to its coalition.

7. *Leave coalition (unsatisfied agents).* **If** my commitment to the leader is low **and** the agent with the best payoff in my neighborhood is not my leader **and** this best agent is independent **then** leave my coalition.

The strategies above allow agents to decide how to behave with respect to coalitions. Firstly, only independent agents that are not obtaining good payoffs consider joining a coalition (strategies 1 and 2). Secondly, an agent obtaining good payoffs in its coalition, strengthens its commitment to the leader (strategy 4). Otherwise, an agent that performs poorly switches from its current coalition (strategy 5), whereas an agent that does not perform poorly but is unhappy with its leader may also either switch coalition (strategy 6) or simply leave the coalition (strategy 7) looking for potentially better coalitions.

Moreover, the model allows some exploration regarding interaction and coalition strategies by the introduction of a *mutation* probability. Mutation may randomly change either the action that independent agents choose to play during interactions, the decisions of agents regarding whether to leave a coalition or not, and the taxes charged by leaders. Therefore, mutation adds exploration to the strategic behavior of independent agents, coalition members, and leaders.

2.2 Coalition Formation over Complex Networks

As stated above, the approach proposed in [5] was successful in helping agents achieve full cooperation (or close to it) on grids. However, grid or grid-like topologies may not model the connectivity/topology that a MAS application may find in a more realistic environment (e.g. P2P, social networks). It has been argued that complex networks provide a more realistic model of the topological features found in many nature, social and technological networks [1, 14] (i.e. computer networks, social networks). Therefore, complex networks provide actual-world topologies where we can evaluate if the coalition formation results exhibited on the grid topology hold. Hence, in this section we aim at evaluating this coalition formation approach (hereafter referred to as the base approach) on actual-world topologies.

To that end, we ran a series of simulations of the base approach over different complex networks. The networks that we employed along with the results are described and discussed in the following subsections.

2.2.1 Network Topologies

This paper's experiments focus on small-world and scale-free networks since these type of networks are the ones that best model the most common networks appearing in societies and nature.

Small-world: These networks present the small-world phenomenon, in which nodes have small neighborhoods, and yet it is possible to reach any other node in a small number of hops. This type of networks are *highly-clustered* (i.e. have a high clustering coefficient). Formally, we note them as $W_V^{k,p}$, where V is the number of nodes, k the average connectivity, i.e., the average size of the node's neighborhood, and p the re-wiring probability. We used the Watts & Strogatz model [19] to generate these networks.

Scale-free: These networks are characterized by having a few nodes acting as highly-connected hubs, while the rest of them have a low connectivity degree. Scale-free networks are *low-clustered* networks. Formally we note them as $S_V^{k,-\gamma}$, where V is the number of nodes and its degree distribution is given by $P(k) \sim k^{-\gamma}$, i.e. the probability $P(k)$ that a node in the network connects with k other nodes is roughly proportional to $k^{-\gamma}$. We used the Barabasi-Albert algorithm [1] to generate these networks.

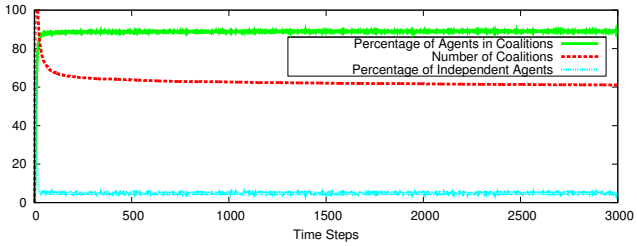


Figure 1: Coalitions in small-world topologies

2.2.2 Experimental Settings

The settings described in this section are also those that will be employed in the rest of this paper (unless otherwise indicated). Each *experiment* consisted of 50 discrete event simulations, each one running up to 20000 time steps (ticks). Each simulation ran with 1000 agents over either a small-world or scale-free underlying topology. Moreover, all the metrics of the simulations were aggregated using the inter-quartile mean (IQM). The experiments used a mutation probability of 0.05 (the same reported in [5]).

In all simulations, interaction topologies were generated by setting the following parameters: $W_{1000}^{10,0.1}$ in small-world networks and $S_{1000}^{10,-3}$ in scale-free networks. The clustering coefficients of the topologies are high (0.492) and low (0.056) respectively. Notice that a new interaction topology is generated per simulation.

2.2.3 Experimental Results

The purpose of first experiments was to determine whether or not the base approach is influenced by the underlying topology. To analyze the results we observed : i) the number of coalitions and independent agents (the closer to a single super-coalition, the higher the cooperation); ii) each agents' payoff with respect to its maximum payoff (the cooperation reward $\times$ the number of neighbors) and taxes; and iii) the topology of the leaders' neighborhoods. In general, the experiments showed that the behavior of the base coalition formation algorithm is strongly dependent on the network topology as we discuss next.

Small-World. Firstly, we observed that in MAS with a small-world connectivity (see figure 1), multiple coalitions emerged (~ 60). This fragmented population is quite a contrast with respect to the grid results, where a single coalition emerged given enough time. Moreover, figure 1 also shows that, at any given time step, around 5% of the population

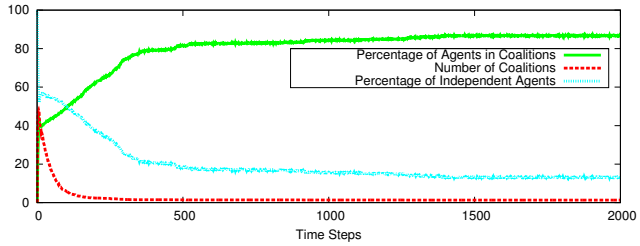


Figure 2: Coalitions in scale-free.

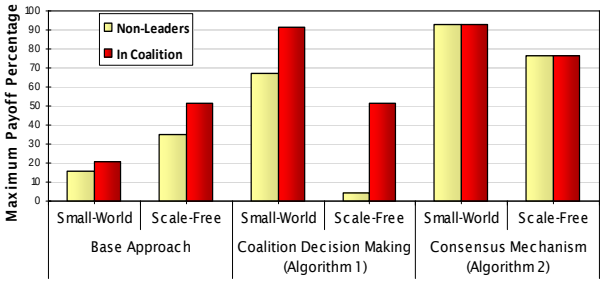


Figure 3: Non-leader (in coalition+independent) agents average payoff.

remains independent. However, the ceaseless spikes exhibited by the plots of both agents in coalitions and independent agents, indicate that agents are continuously leaving and joining coalitions. In other words, coalitions are rather *unstable* because their members continuously change.

With respect to the payoffs, figure 3 shows that the average payoff of an agent in a coalition is significantly low ($\sim$ 20 % of the maximum). Specially when compared with the $\sim$ 99% (of the maximum) obtained in the grid simulations (in [5]). The reasons behind this lower payoff are two-fold: 1) a *fragmented population*; and 2) *very high taxes* imposed by leaders. The former means that as a result of multiple coalitions and independent agents, it is very likely for agents in a coalition to interact (play) with agents outside their coalition (for which their strategy is an automatic defect). The latter occurs because leaders are not pushed to decrease their taxes. In particular, leaders charge their coalition members a $\sim$ 44% of their total payoffs. That fact that agents settle on paying such high taxes greatly differs from the results obtained on grids, where low tax values ($<$ 1% of the total payoff) were reached.

Scale-free. The results over scale-free topologies (depicted in figure 2) show that agents promptly gravitate towards a single leader, thus forming a single super-coalition. However, not all agents join the coalition ($\sim$ 18% of the population, namely $\sim$ 180 agents, remain independent). Moreover, figure 2 exhibits the same kind of instability exhibited by the small-world case (illustrated by the ceaseless spikes).

Interestingly, agents on this topology receive a higher payoff ($\sim$ 50% of the maximum payoff) than on small-world topologies, but still far from the 99% obtained in grids . This occurs because a highly populated single coalition amounts to a very high level of cooperation (i.e. $\sim$ 80% of the agents cooperate with each other). Nonetheless, once again, like in the small-world case, the agents in the coalition also pay very high taxes ($\sim$ 44% of their total payoff).

Moreover, an in-depth analysis of the simulations showed that the agents that became leaders had an interesting characteristic in common. They tend to be the agents with higher connectivity (i.e. they have more neighbors). Hence, the hubs (in particular the highly connected ones, although not necessarily the most connected ones) usually emerge as leaders. Consequently, this is also the reason why a single leader can emerge, since the considerable high number of neighbors that hub agents have with respect to the rest of agents ($\sim$20 vs. $\sim$150) puts them in an excellent influence position. Moreover, the relatively low number of hub agents means that only a few agents compete between themselves to become a leader, thus it is easier for one of them to dominate others.

In contrast, the neighborhoods under small-world topologies are very similar [1] (on average each agent has $\sim$ 10 neighbors) and thus all agents have more or less the same level of influence. Hence, this explains why multiple coalitions coexist (agents start with similar levels of influence).

Summary. Overall, the main drawbacks of the base model are: its sensitivity towards the topology and the coalitions' instability. The first one may be solved by analyzing and revising the base decision making logic (i.e. the coalition strategies), whereas the second issue is harder. The instability exhibited by coalitions mainly occurs because the high mutation (0.05) prompts the agents to *leave* their coalitions (as stated above). However, for large coalitions to appear, high mutation is necessary on both grid (as argued in [5]) and complex network topologies. In other words, mutation is both detrimental and crucial for the coalition formation process. Hence, adjusting mutation is challenging when we want to minimize the instability without affecting coalition emergence.

In the next section we focus on improving cooperation mainly by solving or minimizing the above-mentioned drawbacks.

3. IMPROVING COOPERATION

The aim of this section is to study how to maximize cooperation amongst agents (and consequently improving their payoffs). To that end, the base approach needs to be revised and extended to address the drawbacks identified in the previous section.

Specifically, along this section we focus on: a) achieving full cooperation by emerging a single super-coalition (avoiding a fragmented population); b) sustaining the single coalition through time by minimizing coalition instability; and c) lowering the taxes needed to maintain the coalition. Moreover, all of these needs to occur regardless of the underlying topology. However, notice that although a single coalition promotes cooperation and is beneficial for the agents' payoffs, a single leader becomes a potential single-point of failure, making the MAS vulnerable. Therefore, we also commit to an additional objective: d) the promptly re-emergence of a coalition if the leader fails.

3.1 Topology Influence

The experimental results in section 2.2 showed that the base coalition formation approach is considerably sensitive to the MAS underlying topology. In particular, we observed that the topology influences the structure of coalitions (frag-

[1] because of the small-world phenomenon, see [19]

mented population vs. single coalition). However, the topology also influences other aspects of emergence, i.e. the emergence time. Hence, the purpose of this subsection is to perform a sensitivity analysis of the decision making process (described in subsection 2.1) with respect to the topology.

3.1.1 Influence on Coalition Structures

The most noticeable topological effect observed during the previous experiments was the fragmented population. Specifically, in small-world topologies agents form multiple, different coalitions, which are detrimental to their total payoffs. Therefore, in what follows we aim to promote the emergence of a single coalition.

To understand why multiple coalitions emerged instead of a single one, we must first explain how we expected the base approach to behave. Initially, regardless of the topology, agents organize in small coalitions. Then, agents were expected to leave their coalitions in favor of independence or better coalitions if their payoffs were not sufficient. In other words, by continuously joining and leaving coalitions, agents were expected to incrementally move towards larger coalitions (under the principle that the larger the coalition the higher the payoff) until only a single one remained. However, as the experiments demonstrated in subsection 2.2.3, this behavior does not occur on small-world topologies. Hence, the join and/or leave coalition strategies do not behave as needed.

We determined that the shortcoming stems from *join coalition* strategies instead of leave coalition strategies. Our reasoning is that because of high mutation some agents will always leave their coalitions, thus the fault occurs when they (re-)join them. That is to say, in small-world topologies the join strategies are not moving the agents towards a larger coalition, and instead they keep the population fragmented. Specifically, this occurs because the combination of the small-world's inherent high clustering, the *commitment* notion, and *join coalition strategy 2*, prompt each agent to rejoin the coalition they just left (i.e. most agents never truly leave their coalitions).

We re-ran the experiments to verify if the *join coalition strategy* 2 truly halts the emergence of a single coalition. As expected, we confirmed that without this strategy, agents on small-world topologies are capable of emerging a single super-coalition. Moreover, interestingly enough we found that agents in the single coalition have the additional advantage of paying a significantly *low* tax ($\sim$5% of the agent's total payoff instead of $\sim$44%). The reason behind such low taxes is very reasonable. The fact that every agent can potentially become a leader (as discussed in section 2.2.3) drives a fierce competition between leaders to charge lower taxes (akin to a price war). Overall, low taxes translate onto higher payoffs for coalition agents ($\sim$90 % of the maximum), which is our main objective. Nonetheless, the instability of coalitions is still present and is accountable in lower average payoff obtained by the non-leader agents when compared to the coalition agents (see figure 3).

Nevertheless, the removal of join coalition strategy 2 is detrimental to scale-free topologies. Because of the highly connected hubs in scale-free networks, a single coalition promptly emerges. However, the low clustering of scale-free networks causes agents that recently became independent to remain independent for longer periods of time. This considerably increases the coalition's instability (around one third of the agents are independent at any given point in time). Basically, without a strategy to force agents into a coalition (such as join coalition strategy 2), the number of agents leaving a coalition is higher than the number of agents joining one. In other words, scale-free suffers the full-blown effect of mutation.

To summarize, we reaffirmed the fact that the effect of the coalition decision making process varies depending on the network topology. However, since agents are not capable of identifying the underlying topology where they interact, creating specific strategies for each topology is unrealistic. Nonetheless, when join strategy 2 is removed, coalition emergence is relatively similar in both small-world and scale-free, since only single coalition emerges. This is important because now only *one drawback* remains for both topologies: *instability* (although to a much higher degree in scale-free). Therefore, the remaining objective is to minimize instability, which is the focus of subsection 3.2.

3.1.2 Influence on Emergence Time

In the previous subsection we determined that a single coalition can emerge regardless of the topology. However, we did not mention that the time required for this single coalition emergence varies depending on the topology. In particular, we observed that agents in small-world require a longer time to group up unto a single coalition (4000 time steps) with respect to the agents on scale-free ($<$ 500 time steps). This time disparity is once again a product of the strong influence that hub agents have over the rest of agents. Thus, in this section we aim to speed-up the coalition emergence process on both topologies.

In the base approach, the switch and leave coalition strategies (3,5,6, and 7) are expected to improve coalition emergence time, since they prompt agents to leave their coalitions in search for better ones. However, the leave strategies targeting unsatisfied agents (6 and 7) are hardly ever employed. Therefore, we propose to replace them with the by far more aggressive *disband coalition* strategy. With this strategy, leaders of unprofitable coalitions may disband their coalitions and free multiple unsatisfied agents in just a single time step. This can be regarded as the dual of strategies 6 and 7, since instead of each agent leaving its leader, the leader leaves all its agents.

8. *Disband coalition (unsatisfied leader).* **If** I am a leader and I am not satisfied with my payoff **then** disband my coalition.

Algorithm 1, stands for the resulting coalition decision making process. Notice that after removing the join and leave strategies (strategies 2,6, and 7), none of the remaining strategies employ the notion of commitment employed in Axelrod's tribute model [3]. Thus, the strengthen coalition strategy (strategy 4) was also removed. That is to say, commitment between agents is not actually needed for coalition emergence. We re-ran the simulations to verify the speed-up provided by algorithm 1.

The results showed that by employing the disband strategy a single coalition emerges $\sim$ 12.5 % faster (than when employing strategies 6 and 7) in a small-world topology. Moreover, it speeds up the emergence on scale-free by $\sim$ 50%.

Overall, we have simplified the agents' coalition decision making algorithm. Therefore, we can now turn our attention to our remaining drawback: coalition instability.

Algorithm 1 CoalitionDecisionMaking

```
if (myRole = INDEPENDENT) then
    /* Strategy 1 */
    joinCoalitionWhenWorst(best_neighbor);
end if
if (myRole = COALITION_MEMBER) then
    /* Strategy 3 */
    leaveCoalitionWhenIsolated();
    /* Strategy 5 */
    switchCoalitionWhenWorst(best_neighbor);
end if
if (myRole = LEADER) then
    /* Strategy 8 */
    disbandCoalitionWhenBad();
end if
mutation(p_mutation);
```

3.2 A Consensus Mechanism for Stable Coalitions

After section 3.1 the only issue remaining that prevents full cooperation is coalition instability. Therefore, in what follows we propose to extend the coalition formation approach (in algorithm 1) to endow it with capabilities to minimize instability. However, to accomplish this we must first understand exactly what we are trying to minimize.

3.2.1 Rebellion vs. Mutation

Along this paper we have found that mutation is both a nuisance and a crucial factor for the coalition formation process. However, when analyzing its effects, we realized that the "mutation" employed by the base approach is actually a merge of two different concepts: classic mutation (a random change in the agents' properties) and *rebellion*. The former, has been well studied in the literature [12] and affects agents' actions to play and/or the taxes to charge, whereas the latter is the probability of an agent to become a rebel (leaving its coalition). Thus, in the base approach when mutation occurs in an agent, it randomly changes its actions and taxes, and it prompts the agent to leave its coalition (if applicable). That is to say, both random changes and rebellion occur concurrently. Nonetheless, rebellion (achieved by mutation in previous experiments) is the actual factor that is crucial for the coalition formation process. Hence, it must be treated as a separate entity if we want to minimize the instability resulting from it.

The importance of a rebellion capability is not hard to understand. We have discussed before that larger and stronger coalitions emerge when agents leave their current one to join others. However, the leave or switch coalition strategies do not activate that frequently, and it is actually the rebellion probability the factor that often drives agents to leave their coalitions. This is akin to the not always logical real-life rebellion, e.g. humans may rebel from a social group without actually knowing if there is something better somewhere else. However, as the instability in all previous experiments shows, continuous/constant rebellion is detrimental to agent coalitions. Thus, we propose that, to minimize instability, agents need to adjust their rebelliousness according to their needs (e.g. their payoffs).

3.2.2 The Consensus Mechanism

Rebellion is necessary during the coalition formation process. Nonetheless, it induces instability once a single coalition emerges. Therefore, agent rebelliousness needs to be controlled by the agents themselves accordingly (i.e. only rebel when necessary). Not only that, since agents are distributed entities, rebellion must be controlled distributedly.

Algorithm 2 The new coalition formation algorithm employed by each agent

```
1: interactWithNeighbors();
2: if (myRole ≠ LEADER) then
3:     spread(⟨[tax,p_rebellion],payoff⟩,p_spreading);
4:     [tax,p_rebellion] ←select(spreadings);
5:     innovate([tax,p_rebellion],p_innovation);
6: end if
7: coalitionDecisionMaking();
8: if (myRole = COALITION_MEMBER)
   & (tax<leader.getTax()) then
9:     leaveCoalition(p_rebellion);
10: end if
```

However, if we intend for rebellion to only occur when necessary, we firstly require to give rebellion a motive within the agent, i.e. why should an agent rebel? That is to say, rebellion needs to be dependent on some other property or characteristic of the agents. In the coalition formation process, dissatisfaction with respect to the taxes to pay provides a very logical and reasonable motive for rebellion. Therefore, we propose that an agent may only rebel once its coalition leader is charging more taxes than what the agent is willing to pay. Nevertheless, in both the base approach and in algorithm 1 the agents pay the taxes that the leader charges unconditionally. Hence, to relate taxes and rebellion the agents need to have the notion of how much they are willing to pay, i.e. a *tax threshold*. Moreover, like the rebellion probability, this tax threshold should also be decided by the agents themselves.

In human culture rebellion often occurs as a social movement. Individuals are more likely to rebel if their peers are rebelling, or are more likely to be satisfied with their taxes if their neighbors are satisfied. In other words, rebellion can be regarded as a collective decision. To that end we propose to employ a collective adaptive approach to reach a consensus about the rebellion probability and tax threshold. This proposed collective approach, inspired on the social contagion phenomenon [6], is designed to collectively emerge conventions/consensus about properties common to the agents of a MAS. Under this approach agents with *good properties* (ones that help them improve their payoffs) are more likely to spread them to other agents. For the coalition formation scenario, agents attempt to spread their rebellion probability and tax threshold. For instance, an agent spreading that its tax threshold and rebellion resulted in a high payoff, is likely to persuade other agents to adopt that threshold and rebellion.

Algorithm 2 outlines to the coalition formation algorithm designed to achieve full cooperation and closely maximize the individual agents' payoff on complex networks. The consensus mechanism is included in lines 2-6. Each non-leader agent firstly attempts to spread, with probability $p_{spreading}$, its rebellion and tax threshold using its payoff as an evaluation metric. This is followed by each agent having to decide which of all the incoming spreadings to take (line 4). In our case, an agents always takes the incoming spreading with

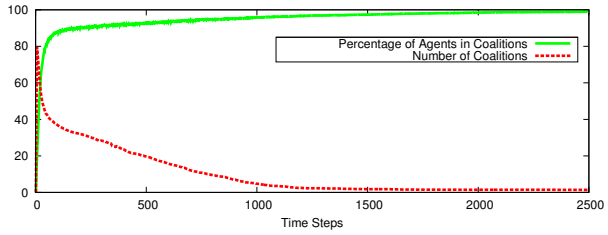


Figure 4: Coalition evolution with consensus on small-world topologies.

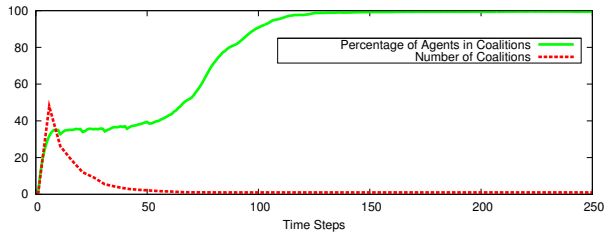


Figure 5: Coalition evolution with consensus on scale-free topologies.

highest payoff (elitist selection). Finally, the rebellion probability and threshold are randomly changed with probability $p_{innovation}$ (line 5).

3.2.3 Sustaining Cooperation

To evaluate the new capacity embedded into the agents, we ran experiments using a moderated spreading probability (0.2) and a low innovation rate (8×10^{-4}). Additionally, the rebellion probability and tax threshold take on values in the range (0,1).

In general, the experimental results showed that with algorithm 2 most agents in the MAS receive high payoffs. Specifically, for both topologies a *stable* single super coalition emerges with a leader that charges low taxes.

The experiments on small-world topologies (depicted in figure 4) show that initially (less than 50 time steps) agents arrange themselves in different coalitions (~ 80), which promptly start to disappear into a *single coalition*. Specifically, the single leader emerges in just ~ 1100 time steps, and around time step 2000 most agents ($\sim 99.5\%$) are already part of the single super-coalition. In other words, a single stable coalition arises such that, almost no agent leaves (very low number), and where agents have a high payoff ($\sim 93\%$ of the maximum, as shown in figure 3). Moreover, the time needed to emerge such coalition is faster than before ($\sim 60\%$ faster, see subsection 3.1.2). These results are achieved through the emergence of low tax values ($\sim 2.5\%$ of the total payoff) together with an extremely high rebellious capacity ($\sim 55\%$). This combination translates to the lemma: "low taxes or rebellion!", which the leaders are forced to comply.

Regarding scale-free topologies (see figure 5), a single coalition is achieved faster than before (in less than 200 time steps vs. ~ 300). What is more, the coalition now is completely stable (very unusual for an agent to leave it) and the taxes ($\sim$20% of the agent's total payoff) are lower than when employing the base approach or just algorithm 1 ($\sim 44\%$ in both cases). When comparing with small-world, observe that the process is similar (an initial peak in the number

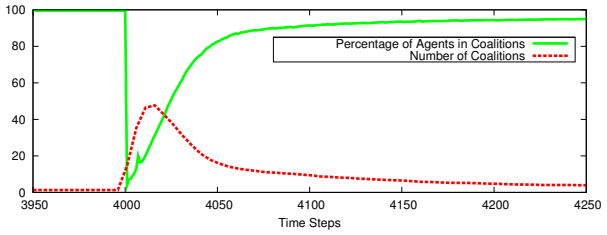


Figure 6: Fault resilience on small-world topologies.

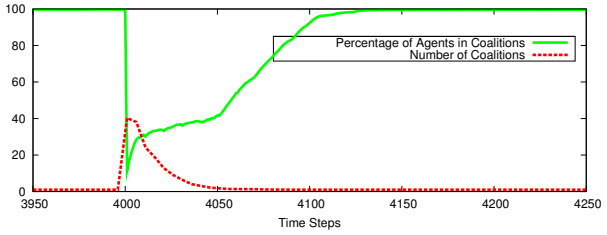


Figure 7: Fault resilience on scale-free topologies.

of coalitions that then decreases into a single coalition) but much faster (10 times faster).

Finally, although full cooperation is closely achieved, it comes with an associated cost: *extra communication*. The spreadings sent by agents represent additional messages. Nonetheless, to emerge a single coalition each agent in a scale-free topology needs to send only ~ 4 messages, while an agent in a small-world topology needs ~ 40 messages.

3.2.4 Fault Resilience

Notice that, in actual (real-world) environments, our cooperation scheme has an associated risk: the existence of a single leader. If the leader agent becomes a target of malicious attacks or fails by chance, all the agents in the coalition will immediately become independent. Therefore, the experiments in this section were designed to evaluate the resilience of our approach to such failures.

To that end we repeated the experiments in the previous section, but now attacking the leader once the single coalition is stable. Specifically, after 4000 time steps (once a single coalition has emerged and proven to be stable) we completely removed the leader agent from the MAS to simulate the leader's failure.

Figures 6 and 7 depict how agents react after the leader is taken down. In general, observe that the response is similar for both topologies. After the leader disappears and all agents become independent, multiple coalitions begin to emerge. However, these coalitions do not last very long (less than 50 time steps) and rapidly start to disband so their members can join a single super-coalition. The peaks in the small-world and scale-free *number of coalitions* plots depicts this transition. The single super-coalition emergence occurs faster because agents already have some good estimations of the tax threshold and rebellion probability (i.e. they are not searching for these values from scratch). Furthermore, once again agents on scale-free are quicker to emerge a single coalition than the small-world ones (< 100 against < 600 time steps). When compared with the previous experiments (figures 4 and 5), emergence is twice as fast on scale-free and four times faster on small-world.

Overall, the experiments show that coalition emergence with a consensus mechanism is resilient against leader failures, which was also one of our main objectives (mentioned at the beginning of this section).

4. CONCLUSIONS

In this paper we confirmed that coalitions indeed facilitate cooperation between self-interest agents. However, we found that the coalition formation process is considerably sensitive to the MAS topology. In particular, to the complex network topologies that model actual-world environments.

To that end we proposed a new distributed, lightweight and efficient coalition emergence approach. We showed that agents on complex network topologies employing this approach can achieve full cooperation by grouping into a single super-coalition. Moreover, agents in this super-coalition can maintain cooperation over time in exchange of some significantly low tax, which is agreed by the agents themselves (thus increasing their overall profits). Hence, closely maximizing their payoffs.

In our experiments, we determined that *rebellion* is a crucial factor for coalition emergence. Through rebellion, smaller and unprofitable coalitions disappear so that bigger ones can rise. Moreover, the agent population can use rebellion to pressure leaders to decrease their taxes. Consequently, increasing competitiveness among leading agents. This contrasts with Axelrod's model [3], where leaders were the ones who pressured the population to the point of extortion. Overall, our proposed approach results in a faster single-coalition emergence and in lower taxes for the population as a whole. Nonetheless, the emergence time and the taxes still vary depending on the topology.

On the one hand, the lowly-clustered, with highly-connected hubs, structure of scale-free topologies gives hub agents an inherent advantage over the rest of the population. Specifically, hub agents can promptly emerge as leaders, dominating the population and getting away with somewhat higher taxes. On the other hand, in the highly clustered small-world topologies, any agent has the potential to become a leader, thus sparking a fiercer and longer (time-wise) price war, which results in much lower taxes.

Furthermore, we determined that commitment to either other agents or leaders (and employed in [3] and [5]) is not essential for coalition formation and maintenance. Even without commitment, a single coalition can emerge and be sustained over time as long as the agents are satisfied with their leaders, which is likely to occur since a leader is always under the threat of rebellion when misbehaving.

Finally, even though it is known that employing a leader based super-coalition introduces a single point of failure into the MAS, our proposed approach is resilient against leader failures (e.g. DOS attacks, disappearance, removal). However, we plan to study how multiple coalition could emerge when the population is divided by goals.

Acknowledgment

The first author thanks the CONACyT scholarship. The work was funded by EVE (TIN2009-14702-C02-01), AT (CSD2007-0022), and the Generalitat of Catalunya grant 2009-SGR-1434.

5. REFERENCES

[1] R. Albert and A. L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47, 2002.

[2] R. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.

[3] R. Axelrod. *Building New Political Actors. The Complexity of Cooperation: Agent-based Models of Competition and Collaboration*. Princeton University Press, 1997.

[4] K. Binmore. *Game Theory*. Mc Graw Hill, 1994.

[5] J. C. Burguillo-Rial. A memetic framework for describing and simulating spatial prisoner's dilemma with coalition formation. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 441–448, 2009.

[6] R. Burt. Social contagion and innovation: Cohesion versus structural equivalence. *American J. of Sociology*, 92:1287–1335, 1987.

[7] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman. On cooperation in multi-agent systems. *Knowl. Eng. Rev.*, 12(3):309–314, 1997.

[8] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111. ACM, 2004.

[9] T. Hogg. Social dilemmas in computational ecosystems. In *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, pages 711–716, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[10] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[11] P. Langer, M. Nowak, and C. Hauert. Spatial invasion of cooperation. *Journal of Theoretical Biology*, 250:634–641, 2008.

[12] M. S. Miguel, V. M. Eguiluz, R. Toral, and K. Klemm. Binary and multivariate stochastic models of consensus formation. *Computing in Science and Eng.*, 7(6):67–73, 2005.

[13] M. Nowak and R. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, 1992.

[14] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics and endemic states in complex networks. *Physical Review E*, 63:066–117, 2001.

[15] J. Pujol, J. Delgado, R. Sangüesa, and A. Flache. The role of clustering on the emergence of efficient social conventions. In *IJCAI 2005*, pages 965–970, 2005.

[16] F. Schweitzer, L. Behera, and H. Muehlenbein. Evolution of cooperation in a spatial prisoner's dilemma. *Advances in Complex systems*, 5(2–3):269–299, 2002.

[17] O. Shehory and S. Kraus. Coalition formation among autonomous agents: Strategies and complexity. In C. Castelfranchi and J. P. Muller, editors, *From Reaction to Cognition*, number 1, pages 57–72, 1995.

[18] K. Tanimoto. Coalition formation interacted with transitional state of environment. In *Systems, Man and Cybernetics 2002*, volume 6, pages 6–9, 2002.

[19] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

An Investigation of the Vulnerabilities of Scale Invariant Dynamics in Large Teams

Robin Glinton, Paul Scerri, Katia Sycara
Robotics Institute, Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA
rglinton, pscerri, katia@cs.cmu.edu

ABSTRACT

Large heterogeneous teams in a variety of applications must make joint decisions using large volumes of noisy and uncertain data. Often not all team members have access to a sensor, relying instead on information shared by peers to make decisions. These sensors can become permanently corrupted through hardware failure or as a result of the actions of a malicious adversary. Previous work showed that when the trust between agents was tuned to a specific value the resulting dynamics of the system had a property called scale invariance which led to agents reaching highly accurate conclusion with little communication. In this paper we show that these dynamics also leave the system vulnerable to most agents coming to incorrect conclusions as a result of small amounts of anomalous information maliciously injected in the system. We conduct an analysis that shows that the efficiency of scale invariant dynamics is due to the fact that large number of agents can come to correct conclusions when the difference between the percentage of agents holding conflicting opinions is relatively small. Although this allows the system to come to correct conclusions quickly, it also means that it would be easy for an attacker with specific knowledge to tip the balance. We explore different methods for selecting which agents are Byzantine and when attacks are launched informed by the analysis. Our study reveals global system properties that can be used to predict when and where in the network the system is most vulnerable to attack. We use the results of this study to design an algorithm used by agents to effectively attack the network, informed by local estimates of the global properties revealed by our investigation.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Theory, Experimentation

Keywords

Emergent behavior, Self-organisation, Distributed problem solving

1. INTRODUCTION

Cite as: An Investigation of the Vulnerabilities of Scale Invariant Dynamics in Large Teams, R. Glinton, P. Scerri and K. Sycara, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 677-684.

[1] In the near future, large heterogeneous teams of robots, agents, and people will be utilized to solve problems in a variety of applications including search and rescue and the military. The sheer size of such teams will mean that the amount of data collected by the team will be overwhelming for its constituents. For this reason, team members will need to share concise information abstractions to maintain shared situational awareness.

The physics of communication, along with environmental constraints, will require team members to communicate via a point to point associates network. This will in turn lead to complex information dynamics and emergent phenomena, which in turn leads to unpredictability.

This paper shows that small amounts of anomolous information introduced to such a belief sharing system can cause errors on a system-wide scale due to the intrinsic dynamics of the system. This could potentially be exploited by a malicious agent attempting to disrupt such a system. Both analytical and empirical evidence is provided to support this assertion.

Previous attempts to describe the vulnerabilities of complex networked system primarily focus on finding vulnerabilities in the network topology without consideration of the dynamics of the process taking place on the network [1]. In this work, the dynamics on the network have a dramatic impact on the vulnerability of the system. Studies which have considered how to influence network dynamics of a complex system include [2]. These all focus on a single type of information spread whereas here we can have conflicting data that fundamentally changes the dynamics and introduces new vulnerabilities due to the way information is fused on the network.

It was recently shown that a team of agents could tune their local trust such that the frequency distribution of cascades of changes in belief followed a power law [3]. When the team was tuned like this, the team's ability to rapidly reach correct conclusions despite noisy data and limited communications was shown to be dramatically higher. However, in this paper we show that when a system is tuned like that, it also becomes extremely vulnerable to malicious attack.

We conduct an analysis to show that for a system exhibiting scale invariant dynamics, a single anomalous sensor reading could result in a number of agents on the order of the size of the system coming to the incorrect conclusion. The analysis compares the rate at which the probability that an agent is on the edge of coming to a correct conclusion, called the percolation probability, increases relative to the same probability for an incorrect conclusion. The anal-

[1]This research has been sponsored in part by AFOSR FA95500810356.

ysis reveals that these two numbers remain close until the agents in the system converge. Although this difference is biased towards correct conclusions, the analysis shows that this difference is small enough for a few anomalous sensor readings to push large numbers of agents towards incorrect conclusions.

To confirm the predictions of the analysis we empirically explore the effect of injecting a single incorrect sensor reading into the system on the correctness of conclusions reached by agents in the system. We show empirically by exhaustively searching trajectories of system execution that there is always a point in that trajectory where injecting a single sensor reading can lead to system wide incorrect conclusions. We further show that an adversary could mount an effective attack on the system if the adversary had global knowledge of the distance of the system from the percolation threshold for the incorrect conclusion.

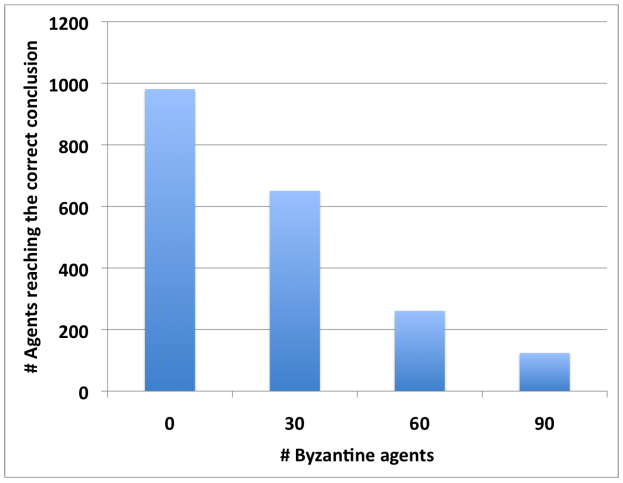


Figure 1: Belief sharing system exhibiting scale invariant dynamics is vulnerable to a small percentage of Byzantine agents.

Just as complex systems can be attacked from external sources, it is also possible for attacks to originate from within. Thus it is necessary to understand the potential vulnerabilities of such a system to threats from within. To this end we study the vulnerability of the agents within the system to reaching incorrect conclusions as a result of the action of Byzantine agents within the system. Specifically, we study mechanisms for picking the most vulnerable points in the network for attack by Byzantine agents. We explore Several different mechanisms for selecting which nodes are Byzantine, using methods typically employed in the study of the vulnerabilities in network topologies to network disintegration. The study reveals that the most effective method is that which selects the nodes with the maximum number of neighbors. Finally, our study shows that as the number of Byzantine agents in the network increases, the trust range between agents that results in a scale invariant distribution of cascades is no longer optimal. As the number of byzantine agents increases the optimal value of trust is lowered slightly with the agents becoming slightly more conservative to account for the misinformation circulating in the system.

In a large distributed system it is unlikely that an adversary would have access to the global network state or topology, thus it is desirable to study whether an effective attack on the system could be launched using only local knowledge of the network state and topology. To investigate the feasibility of a practical attack we developed a local algorithm, inspired by [4], where Byzantine agents use knowledge of the local connectivity and a local estimate of the percolation threshold to decide when and where to focus an attack. We found that such an attack is as effective, in reducing the number of agents that come to a correct conclusion, as an attack mounted with full knowledge of the system state and network topology.

The remainder of this paper is organized as follows: Section 2 gives an overview of the model of a belief sharing system used to study emergent vulnerabilities. Section 3 presents an analysis that reveals a vulnerability of such a system to small amounts of anomalous information introduced by an adversary. Section 4 empirically explores the vulnerability of the system to spoofed sensor readings introduced by an adversary with global system knowledge. Section 5 empirically explores the vulnerability of the system to Byzantine agents with detailed knowledge of the network topology and state. Section 6 explores the feasibility of effective attacks based on partial knowledge of the system. Section 7 presents the related work and Section 8 presents conclusions and future work.

2. MODEL

In this section, we formally describe the underlying model used in the remainder of the paper. A cooperative team of agents, $A = \{a_1, \ldots, a_{|A|}\}$ are connected by a network, $G = (A, E)$ where E is the set of links in G which connect the agents in A. An agent a_i may only communicate directly with another agent $a_j \in N_{a_i}$ if $\exists e_{i,j} \in E$ where we refer to the set N_{a_i} as its *neighbors*. The average number of neighbors that the agents in G have is defined as $< d >$ where $< d >= \frac{\sum_i |N_{a_i}|}{|A|}$.

Sensors, $S = \{s_1, \ldots, s_{|S|}\}$ provide noisy observations to the team. Only one agent can directly see the output of each sensor. The sensors return binary observations about some fact b from the set $\{true, false\}$. We refer to the probability that a sensor s will return a correct observation as its reliability r_s. The reliability of a sensor is known to the agent that receives observations from it.

In the remainder of this paper, unless otherwise specified, $|A| = 1000$, $|S| = |A|/20$ and $r_s = 0.55 \forall s$.

A key assumption of the model is that it is infeasible for agents to communicate actual sensor observations to one another and that they may only communicate whether they currently believe the fact to be *true*, *false* or if they are undecided, *unknown*.

Each agent a_i uses either an observation received from a sensor or conclusions about b communicated by neighbors to form a belief $P_{a_i}(b \rightarrow true)$ about b. A new observation is incorporated into the current belief to form a new belief $P'_{a_i}(b \rightarrow true)$ using an expression of Bayes' Rule with cp as the conditional probability that the neighbors conclusion is correct. In this model cp acts as a measure of the trust between agents.

An agent will come to a conclusion about the truth of the fact and communicate this conclusion to neighbors if its belief in that conclusion exceeds a fixed threshold. The details of the belief update calculation and thresholding were taken from [3]. When an agent comes to a conclusion and communicates with neighbors, the neighbors may then come to a conclusion and communicate. This chain of conclusion formation is called a *cascade*. Previous work showed that agent conclusions are most accurate when the probability $P(c)$ that c agents change their belief during a cascade is given by $P(c) \propto c^{-3/2}$. The most important metric used in this paper is T_a, the number of agents in the network coming to the correct conclusion. We define the system under study to be vulner-

able if there are small sets of Byzantine agents $\hat{a}$, subset $\hat{a} \subset |A|$ such that T_a and $|\hat{a}|$ is greatly reduced. Another objective is to find times at which the system is most vulnerable to the injection of anomalous sensor readings and agent conclusions.

3. THEORY OF SCALE INVARIANT VULNERABILITY

In this section we conduct an analysis that reveals a vulnerability to attack in systems which exhibit scale invariant dynamics. Such systems have been shown to enable very accurate and efficient belief fusion using very little communication. We would like to leverage this efficiency to design practical information fusion systems. However, first it is necessary to understand potential vulnerabilities of such a system to adversarial action. To this end we analyze the difference in the rate at which agents in the system reach correct conclusion (called the percolation probability for that conclusion) relative to the rate at which they approach incorrect conclusions. Our, analysis reveals that although agents overwhelmingly reach correct conclusions at a higher rate, the difference to the rate at which they near incorrect conclusions is small. This suggest that when the majority of agents are close to making a decision, a single anomalous sensor reading could offset this balance causing a large percentage of agents to reach the incorrect conclusion instead.

Previous work [5] showed that the probability of a large cascade disseminating a conclusion system wide is given by:

$$\sum_{\hat{k_t}} \sum_{\hat{s_t}} \beta(\hat{k_t}, \beta(\hat{g_t})) P(\hat{k_t} \mid \hat{s_t}) P(\hat{g_t} \mid \hat{s_t}) \beta(\hat{s_t}) \quad (1)$$

This occurs when the percolation probability for that conclusion exceeds a threshold called the percolation threshold.

The vector $\hat{k_t} = [k_0, k_1, \ldots, k_t]$ gives the sequence of the sizes of avalanches that occurred over time. Similarly the vector $\hat{g_t}$ gives the sequence of false avalanches that occurred. (Note for the model presented in this paper, only a single cascade per time step is possible). Finally the vector $\hat{s_t} = [s_0, s_1, \ldots, s_t]$ gives the sequence of sensor readings input to the system up until time t. The terms in Equation 1 are as follows: The term $\beta(\hat{s_t})$ gives the probability of a specific sequence of sensor readings input to the system, $P(\hat{k_t} \mid \hat{s_t})$ and $P(\hat{g_t} \mid \hat{s_t})$ give the probability of a resulting sequence of cascade sizes of correct and incorrect conclusions respectively. Finally $\beta(\hat{g_t}, \hat{k_t})$ gives the probability that a random agent in the network will be touched by a net number of correct cascades such that it is one correct communication from a neighbor away from reaching the correct conclusion.

Starting with Equation 1 we show that the difference between the probability of a large cascade of correct conclusions and a large cascade of incorrect conclusions is small just before a large cascade of correct conclusions occurs, revealing a vulnerability in the system. To facilitate ease of computation we simplify Equation 1 by observing that the scale invariant distribution is heavy tailed, meaning that the probability of a cascade of size 1 is close to 1. It is then reasonable to assume that before a large cascade occurs, all cascades are of size 1. Under this assumption, given a specific sequence of sensor readings $\hat{s_t}$, all of the probability mass of the cascade sequence distribution $P(\hat{k_t} \mid \hat{s_t})$ collapses to a single possible sequence of cascades. With this simplification Equation 1 is reduced to Equation 2:

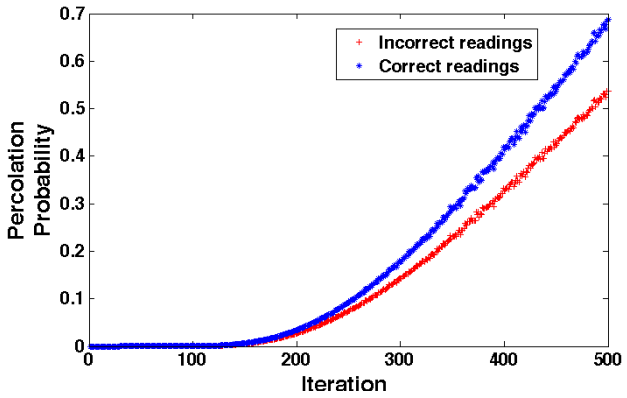


Figure 2: The percolation probability for incorrect information stays near that for correct information until just before the threshold is exceeded.

$$\sum_{\hat{k_t}} \sum_{\hat{s_t}} \beta(\hat{k_t}, \hat{g_t}) \beta(\hat{s_t}) \quad (2)$$

All three terms of this equation are binomially distributed. We can further simplify computation using this expression by recognizing that a binomial distribution can be approximated by a normal distribution. The first term in the equation which gives the probability of the difference between the number of competing cascades that reached the agent is then normally distributed with mean $\mu = \frac{n_T - n_F}{|A|}$ and standard deviation $\sigma = \frac{n_T}{|A|} \frac{1}{1-|A|} + \frac{n_T}{|A|} \frac{1}{1-|A|}$. Where n_T and n_F give the number of correct sensor readings and incorrect sensor readings in the the sequence $\hat{s_t}$. The probability of a net number of false cascades touching the agent is obtained by simply switching the n_T and the n_F in the normal distribution.

With this substitution it is easy to numerically integrate Equation 2, to give the percolation probability for correct and incorrect cascades. The result of this computation is shown in Figure 2. The x-axis of this figure gives the timestep and the y-axis gives the percolation probability. For the random network the calculation was conducted for, the percolation threshold that would result in a large cascade is .33. In the figure it is evident that the percolation probability for a correct cascade reaches this threshold first. However, at this point the percolation probability for the large incorrect cascade is 0.25. This difference corresponds to less than 5% of the agents in the system. For the system under study with $|A| = 1000$, this is less than 50 agents. This estimate is a maximum because the analysis was predicated on only avalanches of size 1 occuring and the assumption of a loop free network. In practice relaxing either of these conditions would reduce the number of agents necessary to upset the balance and cause a large cascade of incorrect information.

The conclusion is that relatively few sensor readings or a small number of Byzantine agents could potentially cause a system on the verge of large numbers of agents reaching the correct conclusion to have the exact opposite occur. Furthermore, this result suggests that the system is particularly vulnerable near the percolation threshold. Although the curves in Figure 2 are even closer together at lower percolation probabilities, additional Byzantine agents or anomalous sensor readings would be required to drive the system closer to the percolation threshold. For example at iteration 300

an additional 150 agents would need to be influenced to drive the system to the percolation probability for a cascade of incorrect conclusions. Figure 4 illustrates the vulnerability. As the agents in the system near a correct conclusion, there is a smaller group of agents on the edge of an incorrect conclusion. A single anomalous sensor correction can set off a large cascade among such agents leading to large numbers of agents coming to incorrect conclusions.

4. BYZANTINE SENSORS

In this section we investigate the vulnerability of a system exhibiting scale invariant dynamics of belief exchange to small amounts of anomalous sensor readings introduced to sensors by a malicious attacker. The results of this section show that for all of the network topologies with the exception of Small World, there was always a point in time at which injecting a spurious sensor reading would result in large numbers of agents reaching the incorrect conclusions. In addition, experiments reveal that an adversary with knowledge of the number of agents in the system 1 communication from a neighbor away from reaching a correct conclusion could use this information to decide the best times to introduce spurious sensor readings into the network for maximum impact on the conclusions reached by agents with minimal intervention.

We conducted experiments to explore this potential system vulnerability. In the first experiment we test if an adversary with total knowledge of the system, including all of the possible trajectories of the system dynamics could cause the agents in the system to adopt the wrong conclusions. In this experiment we exhaustively search trajectories of the system simulation for points where introducing a single incorrect sensor reading will result in a cascade for which greater than half of the agents in the system incorporate the incorrect sensor reading into their belief. The exact procedure for searching the system trajectories is as follows. First, a *snapshot* of the system is taken where the current belief state of all of the agents is recorded. Next, we exhaustively consider what would happen if incorrect sensor readings were introduced to every permutation of two sensors in the system. For each such permutation, the resulting cascades, if any, are allowed to propagate until the system quiesces. The agents are then restored to their states before the introduction of the incorrect readings before the next permutation is explored. If a large cascade does not result, the agents are returned to their previous belief state and the system is allowed to evolve as if the intervention did not occurr. The entire procedure is then repeated.

In this experiment we recorded the number of large cascades that occurred as a result of malicious intervention during 10 rounds of the above procedure, where each round consists of 100 steps, where each step consists of the permutation search discussed above. The parameter values used during this experiment were $|A| = 1000$, $|S| = 1/20|A|$, $s_r = 0.55$, and $< d >= 4$. The results of the experiment are given by Figure 3. The x-axis gives cp and the y-axis gives the number of rounds out of 10 in which greater than 50% of the agents incorporated the incorrect information artificially introduced to the sensors.

The plot shows that during almost every round, there is a point in the system trajectory where introducing incorrect information at the sensors would have resulted in a large cascade, propagating this incorrect information to more than half the agents in the system. This only occurs for rounds when the value of cp approaches the value which results in scale invariant dynamics. This suggests that an omniscient agent could almost always cause the agents in the system to come to the incorrect conclusion. This of course is not practical and in the next experiment we investigate what information could be used by a malicious actor to mount a practical attack on the system.

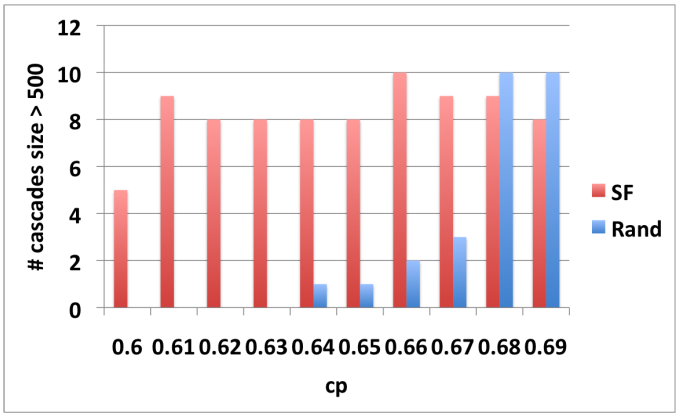


Figure 3: Cascades resulting from malicious intervention at sensors.

The preceding experiment showed us that there is almost always a point in the trajectory of the system where the system is extremely vulnerable to malicious intervention using a small amount of misinformation. However, the experiment did not tell us anything about *when* the system is most vulnerable. Specifically, the experiment did not reveal what properties of the system could be used by a malicious actor to decide when to inject misinformation at the sensors. We hypothesize, due to the results of Section 3 that the system would be most vulnerable to such an attack when the system is on the edge of making a decision. That is when the agents are approaching a percolation threshold for a large correct avalanche.

The percolation threshold in this case is a network specific probability that a randomly selected agent requires a single communication from a neighbor to come to a conclusion. We conducted an experiment to test this hypothesis. In this experiment we simulated 1000 runs of the system and injected a single incorrect reading at a randomly chosen sensor using two methods to decide when to inject the reading. In the first method we simply randomly selected the time-step at which to inject the incorrect sensor reading. In the second method the reading was injected when the percolation probability was at the percolation threshold. We repeated this for each network topology under study. The results are given in Table 5.

Network	Random success rate	Percolation success rate
SF	0%	95%
R	0%	63%
SW	0.03%	83%

Figure 5: The effect of using the percolation probability of the network to decide when to attack the network compared to random attack.

5. BYZANTINE AGENTS

In this section we investigate the vulnerability of a belief sharing system exhibiting scale invariant dynamics to attacks on, or malfunction of the agents that exchange fused information within the system. We experiment with three methods for selecting Byzantine nodes, all based on global knowledge of the network topology.

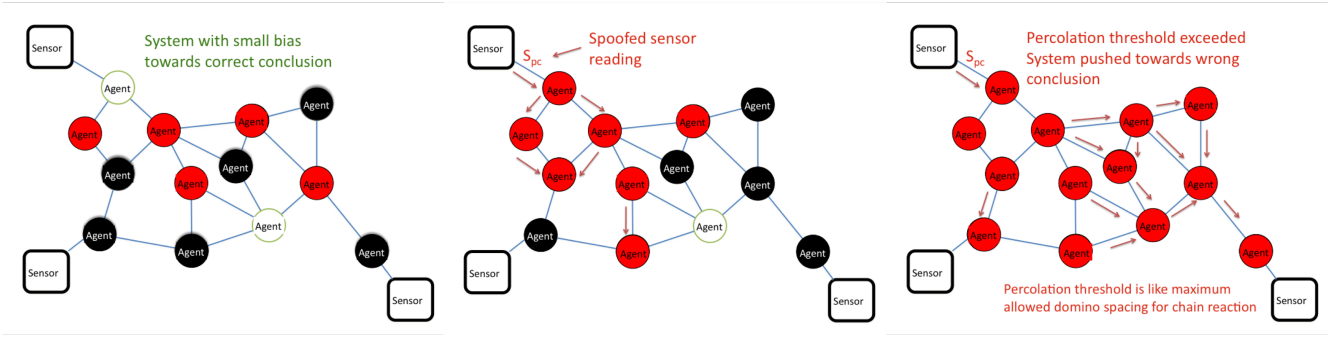


Figure 4: A single incorrect sensor reading introduced by an adversary can percolate through the network causing widespread incorrect conclusions.

The analysis of Section 3 revealed that changing the decisions of a relatively small number of agents in the system could dramatically reduce the number of agents reaching the correct conlusion. This suggests that a small number of Byzantine agents could influence the conclusions of the majority of the agents in the system by sharing incorrect information or noise.

In this section we analyze this assertion by empirically exploring the vulnerability of the system to the action of malicious or malfunctioning agents. Specifically we analyze the effect of malicious agents in the system on the performance of the system as measured by the the number of agents in the system that reach the correct conclusion. We investigated two types of Byzantine agents. The first type of Byzantine agents we investigated pathologically share incorrect information. The second type shares random information. Both types of agent simply ignore any information received from neighbors or sensors.

One of the key results of this section is that a relatively small number of Byzantine agents dramatically reduces the number of agents that reach a correct conclusion over all network types. In addition, we find that the trust value that results in scale invariant dynamics is no longer optimal when a small number of Byzantine agents are present.

5.1 Byzantine agent selection

Three different methods of selecting which nodes in the simulation would be Byzantine were used in experiments. In the first method, nodes are simply drawn at random from a uniform distribution over all of the nodes in the system. The second method which we call the maximum influence method is a modified version of the method due to Kleinberg [6]. Using this method, a node is selected by the number of nodes that would be *infected* by a cascade starting at that node. We call this a nodes influence number. The nodes with the highest influence numbers are selected. To calculate the influence number of node Q each node is initially marked uninfected. Next node Q communicates with its neighbors. When these neighbors receive this communication they draw a real number in the range $[0, 1]$ from a uniform distribution. If this number exceeds a threshold, the node marks itself and communicates with neighbors otherwise it does nothing. When all communication ceases, the influence number of Q is the number of nodes in the network marked infected.

The third method of Byzantine node selection picks the nodes with the largest number of neighbors, this is called the max node method. Figure 6 shows the node that would most likely be selected

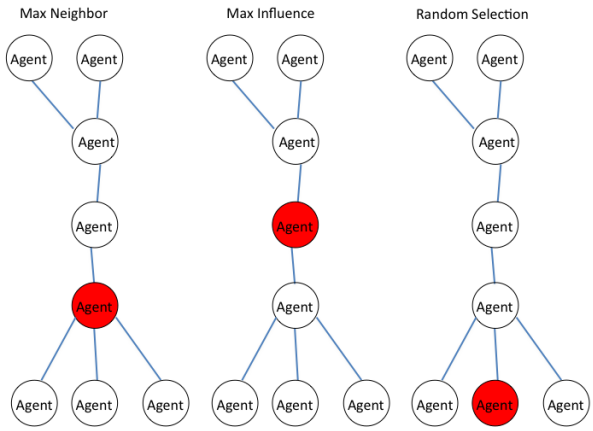


Figure 6: Three methods used for selecting Byzantine nodes.

first in a particular graph structure. The max node method picks the node that simply has the highest fanout while the max influence method is biased towards the node with the most pathways to the other nodes in the network.

5.2 Byzantine agent experiments

First we conducted an experiment to investigate the result on system performance of Byzantine nodes which pathologically share incorrect information with neighbors. In the experiment, we investigated how system performance as measured by the number of agents reaching the correct conclusion, changed as the number of Byzantine nodes in the system was varied. Experimental parameters are as follows: The number of Byzantine nodes in the system was varied from 0-10% of $|A|$ in increments of 1%. All remaining graphs in this section were produced using the parameter values $|A| = 1000$, $|S| = 50$, $r_c = 0.2$, and $r_s = 0.55$, and $< d >= 4$. We also varied the structure of the communication network used by the agents. The results are given by Figures 7 and 8. In Figure 7 the x-axis gives the number of Byzantine agents out of 1000. The y-axis gives x the number of agents out of 1000 that come to the correct conclusion. Each curve represents a different network topology including Random, Scale Free, and Small Worlds networks. The leftmost plot shows the results when Byzantine nodes are selected at random, the middle plot shows the results for nodes selected using the maximum influence method, and the leftmost plot shows

the results when nodes that have the largest number of neighbors are selected.

The first trend evident across all of the communication networks is that with a relatively small percentage of Byzantine nodes, the number of agents that comes to the correct conclusion drops dramatically. In fact with 10% of the nodes in the system, only the Scale-Free network has greater than half the agents in the network reaching the correct conclusion. The theoretical limit, due to Lamport [7], says that agents in a network can reach a correct consensus with a maximum of 33% of the nodes as Byzantine. The system under study requires less than 10% of the agents to be Byzantine, to prevent a correct consensus in the truth of the fact being monitored. Also all networks are most vulnerable when nodes with the maximum number of neighbors are chosen.

For the ScaleFree network, the trend of the vulnerability of this system with respect to the way nodes are selected for the injection of misinformation, reflects the known results for the vulnerability of the ScaleFree network to the removal of nodes. The ScaleFree network proves most robust of all of the networks when Byzantine nodes are selected at random, with 60% of the agents reaching the correct conclusion with 10% of the nodes Byzantine. Conversely, the ScaleFree network is most vulnerable when the nodes with the largest number of neighbors are selected. In this case, with only 1% of the nodes Byzantine, less than 10% of the agents in the network come to the correct conclusion. This can be explained by the extremely skewed distribution of the number of neighbors that each nodes has in a scale free network. A Scale Free network has a long tailed distribution, with a few nodes, called hubs, having a large number of neighbors and the remainder of the nodes having relatively few neighbors. When nodes are selected at random, their is a low likelihood that the hubs will be selected and the fused information originating at the hubs overwhelms that spread by the Byzantine nodes. Conversely, the hubs have a disproportionately large effect on the network spreading misinformation widely when they are Byzantine.

The second trend evident across all of the networks is that for the Random network topology, there is a distinct threshold in the number of Byzantine nodes beyond which the number of agents that reach the correct conclusion drops suddenly and dramatically. This threshold is 6% of the agents for both the random agent selection and selection for agents with the maximum number of neighbors. This threshold drops to 4% when the maximum influence method is used for node selection.

All network topologies perform about the same for the maximum influence method of selecting nodes to be Byzantine. This suggests that the specific dynamics of this system have a large effect on the which nodes are vulnerable within the system. Otherwise the generic influence spreading, which is dependent on the static topology of the network itself, would be much more effective at means of picking Byzantine nodes to cause the maximum number of nodes to come to the incorrect conclusion.

The network with the Small Worlds topology shows a linear drop in the number of agents reaching the correct conclusion with increasing numbers of Byzantine agents.

Over all for a system with these dynamics, the Scale Free network topology would be the best choice. It is least vulnerable to all attacks except attacks on the hubs. Since the hubs in the network are relatively few, they would take a relatively small amount of computational resources within a system to monitor for intrusion. Furthermore, an attacker would need a large amount of information about the topology of the network to select nodes for attack effectively. Below its vulnerability threshold, the random network is the least vulnerable, and would be the best choice of network topology in a secure environment where an attacker could only select relatively few nodes to attack.

Figure 8 shows how the value of cp which results in the largest number of agents reaching the correct conclusion, and hence which associated system dynamics as discussed in Section 2 are least vulnerable, as the number of Byzantine nodes in the system changes. The x-axis of the figure gives the number of Byzantine agents out of 1000 in the system. The y-axis gives the center of mass of cp. The center of mass is the mean value of cp over simulation runs weighted by the number of agents that reach the correct conclusion for that value of cp. The center of mass is defined mathematically as $\sum_i \frac{cp_i * nT_i}{nT_i}$, where nT_i is the number of agents that reached the correct conclusion for simulation run i. The most notable trend in Figure 8 is that for the network with the Random topology there is a distinct shift of the trust value cp that gives the best performance, away from the value that results in scale invariant dynamics.

The high level conclusion of this Section is that the *scale invariant* dynamics that were previously showed to lead to high accuracy in the conclusions of agents, leaves the system vulnerable to intervention by a small number of Byzantine nodes. This means that a system utilizing scale invariant dynamics, or that intrinsically had such dynamics would either need to operate in a very secure environment, or explicitly have a mechanism to detect the presence of Byzantine nodes.

6. ATTACKS WITH LIMITED INFORMATION

In previous sections experiments have shown that an adversary could dramatically reduce the accuracy of agent's conclusions using global system knowledge. However, in practice, it is more likely that an adversary would have only partial knowledge of the system. To investigate the vulnerability of the system to attacks based on partial system knowledge, we developed an algorithm used by Byzantine agents to attack the system using only local information about the system. In sections4 and 5 we found that the system was most vulnerable at times when close to the percolation threshold in agent decisions. We also found that most networks exhibiting scale invariant dynamics were most vulnerable at nodes with many neighbors. For this reason, the Byzantine agents executing our attack strategy use local estimates of the percolation threshold in the network to decide when to attack and knowledge of the local network topology to decide where to attack.

The details of the algorithm are as follows. The Byzantine agent draws a random number in the range $[1, |A|]$ from a uniform probability distribution. If this number falls below a threshold, which we call the activity threshold, the agent continues to operate normally, fusing conclusions of neighbors and communicating the resulting conclusion. This threshold is intended to ensure that only a preselected percentage of the Byzantine agents in the system are active at any time. If the random number drawn by the agent exceeds the activity threshold, the agent estimates the distance of the agent and it's neighbors from the percolation threshold. The knowledge of the percolation threshold suggests that the attacker would have knowledge of the high level topology of the network (e.g. Random vs. Scale Free) but not specific details of the connectivity in the network. If this estimate is within a given distance from the

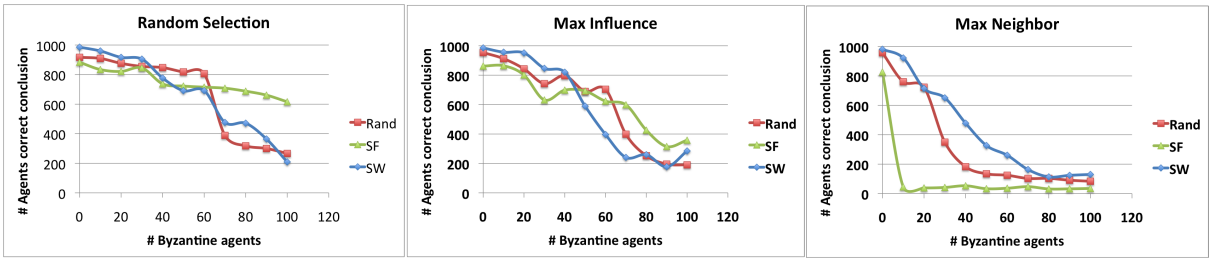


Figure 7: The effect of Byzantine nodes on the correctness of the conclusions of agents across the three methods for selecting Byzantine nodes.

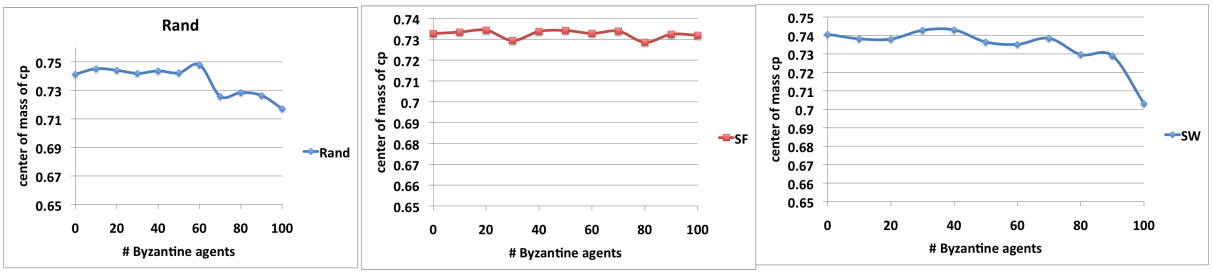


Figure 8: The effect of Byzantine nodes on the cp that gives best performance.

percolation threshold for the network, the agent then sends several incorrect conclusions to its neighbor that has the highest number of network links.

We conducted an experiment to test the efficacy of this algorithm over a range of networks. The parameters used are the same as those for previous sections. The activity threshold is varied between 0.05 and 0.30 (effectively varying the number of active Byzantine agents between 5% and 30%. This is plotted against the average number of agents that reach the correct conclusion. This plot is shown in Figure 9. The plot shows that relatively few agents

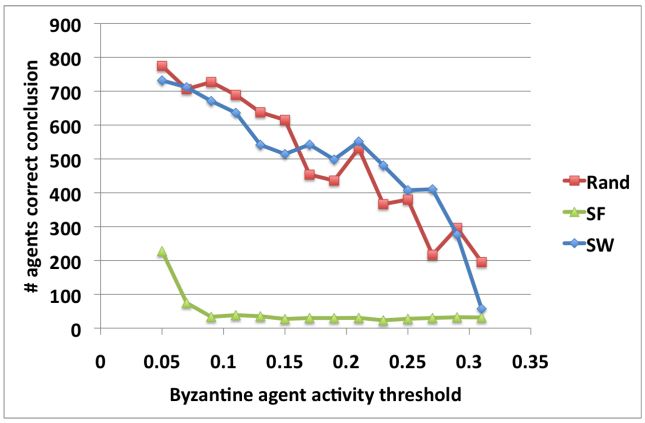


Figure 9: The effect of Byzantine nodes using only local knowledge of the system on the accuracy of the conclusions reached by agents in the network.

using the algorithm, dramatically reduce the number of agents in the system reaching correct conclusions over a range of network topologies.

7. RELATED WORK

There have been several studies conducted to investigate models whose dynamics are governed by cascades on complex networks. These include models of fads[8, 9], rumors [10], gossip[11], forest fires [12], and diseases[13, 14]. Common to all of these models is that the dynamics are governed by the spreading of a single influence. In contrast, our model investigates competing influences which significantly alters the dynamics of a system.

In [15], Parunak presents a model of the collective convergence of agents to a cognitive state. This model is similar to ours in that it does include multiple states that agents can converge to and hence competition between states. Parunak focuses on studying the macroscopic performance of the system. We build upon Parunak's investigation by analyzing the dynamics of the system directly and investigating the relationship between the dynamics and the performance of the system.

A number of studies have investigated the impact of Byzantine nodes on the performance of a distributed system and mechanisms for coping with their presence [16],[17],[18]. We extend these studies by investigating how the efficacy of Byzantine agents are impacted by the dynamics of a system exhibiting scale invariance in belief exchange.

Previous work has extensively explored methods for picking network nodes that are most vulnerable to fracturing the structure of the network [1], [19]. This paper considers the impact of many of the metrics discussed in this body of work on information dynamics on a network by using them for the placement of agents that spread misinformation on a belief sharing network.

Recently there has been significant interest in social networks [20], [21] and the impact of those networks on performance of a group. For example, Xu looked at the impact of networks on routing information to a specific agent [22]. Kleinberg, looked at the impact of the network on the performance of decentralized search algorithms [6], when a single agent has information valuable to the system. We build on both of these contributions by investigating

the case when a large percentage of the agents in the team are both sources and sinks for information, which fundamentally changes the dynamics of information exchange in the system.

8. CONCLUSIONS AND FUTURE WORK

When information exchange between agents exhibits scale invariant dynamics, the speed and reliability with which the team can converge to correct conclusions, despite noisy data and highly limited communication is dramatically increased. Before, this property can be leveraged to design efficient information fusion, we need to understand the vulnerability of the system to malicious intervention. In this work we found that scale invariant dynamics make a system susceptible to the presence of Byzantine agents and sensors. We showed analytically that when the agents in the system are near to a correct conclusion, they are simultaneously near to coming to an incorrect conclusion. This leaves the system vulnerable to small amounts of anomalous information and small number of Byzantine agents. We found that Byzantine agents were most effective at reducing the accuracy of the conclusions of other agents when placed at high degree nodes in the network. We further found that attacks were most effective when launched when the network is close to a percolation threshold in the decisions of agents. In future work, we propose to extend the model to capture additional features of information sharing, including beliefs of several variables and a richer communication model, while maintaining the mathematical simplicity that allows the types of detailed analysis above. We also intend to simulate features that are harder to model mathematically, such as the ways mobile sensors might be redeployed based on initial conclusions and how other coordination activities can influence belief convergence. Finally we intend to develop mechanisms for detecting Byzantine or malfunctioning agents and mitigating their impact on system performance informed by the algorithm described in this work.

9. REFERENCES

[1] Y. Chen, G. Paul, R. Cohen, S. Havlin, S. P. Borgatti, F. Liljeros, and H. E. Stanley, "Percolation theory applied to measures of fragmentation in social networks," *Phys. Rev. E*, vol. 75, no. 4, p. 046107, Apr 2007.

[2] M. Lelarge, "Efficient control of epidemics over random networks," in *SIGMETRICS/Performance*, 2009.

[3] R. Glinton, P. Scerri, and K. Sycara, "Exploiting scale invariant dynamics for efficient information propagation in large teams," in *Proc. of AAMAS'10*, 2010.

[4] R. Cohen, S. Havlin, and ben Avraham D., "Efficient immunization strategies for computer networks and populations," *Physical Review Letters*, 2003.

[5] P. Glinton, R.and Scerri and K. Sycara, "An explanation for the efÞciency of scale invariant dynamics of information fusion in large teams," in *Proceedings of Fusion*, 2010.

[6] J. Kleinberg, "Complex networks and decentralized search algorithms," in *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.

[7] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[8] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades," *Journal of Political Economy*, vol. 100, no. 5, p. 992, 1992. [Online]. Available: http://www.journals.uchicago.edu/doi/abs/10.1086/261849

[9] W. DJ, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Science*, vol. 99, pp. 5766–5771, 2002.

[10] M. Nekovee, Y. Moreno, G. Bianconi, and M. Marsili, "Theory of rumour spreading in complex social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 1, pp. 457–470, 2007.

[11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2508–2530, 2006.

[12] S. Clar, B. Drossel, and F. Schwabl, "Scaling laws and simulation results for the self-organized critical forest-fire model," *Phys Rev E*, vol. 50, p. 1009Ð1018, 1994.

[13] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Phys. Rev. Lett.*, vol. 86, no. 14, pp. 3200–3203, Apr 2001.

[14] V. M. Eguíluz and K. Klemm, "Epidemic threshold in structured scale-free networks," *Phys. Rev. Lett.*, vol. 89, no. 10, p. 108701, Aug 2002.

[15] V. Parunak, "A mathematical analysis of collective cognitive convergence," in *AAMAS '09*, 2009, pp. 473–480.

[16] D. Dolev and H. R. Strong, "Authenticated algorithms for byzantine agreement," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 656–666, 1983.

[17] J.-P. Martin, "Fast byzantine consensus," *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 3, pp. 202–215, 2006.

[18] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," *Applications and the Internet Workshops, IEEE/IPSJ International Symposium on*, vol. 0, p. 368, 2003.

[19] H. J. . A.-L. B. RŐka Albert, "Error and attack tolerance of complex networks," *Nature*, vol. 406, pp. 378–382, July 2000.

[20] D. Watts and S. Strogatz, "Collective dynamics of small world networks," *Nature*, vol. 393, pp. 440–442, 1998.

[21] A.-L. Barabasi and E. Bonabeau, "Scale free networks," *Scientific American*, pp. 60–69, May 2003.

[22] Y. Xu, P. Scerri, B. Yu, S. Okamoto, M. Lewis, and K. Sycara, "An integrated token-based algorithm for scalable coordination," in *AAMAS'05*, 2005.

The Evolution of Cooperation in Self-Interested Agent Societies: A Critical Study

Lisa-Maria Hofmann
Karlsruhe Institute of Technology
Karlsruhe, Germany
lisamariahofmann@gmail.com

Nilanjan Chakraborty
School of Computer Science
Carnegie Mellon University
Pittsburgh, USA
nilanjan@cs.cmu.edu

Katia Sycara
School of Computer Science
Carnegie Mellon University
Pittsburgh, USA
katia@cs.cmu.edu

ABSTRACT

We study the phenomenon of evolution of cooperation in a society of self-interested agents using repeated games in graphs. A repeated game in a graph is a multiple round game, where, in each round, an agent gains payoff by playing a game with its neighbors and updates its action (state) by using the actions and/or payoffs of its neighbors. The interaction model between the agents is a two-player, two-action (cooperate and defect) Prisoner's Dilemma (PD) game (a prototypical model for interaction between self-interested agents). The conventional wisdom is that the presence of network structure enhances cooperation and current models use multiagent simulation to show evolution of cooperation. However, these results are based on particular combination of interaction game, network model and state update rules (e.g., PD game on a grid with imitate your best neighbor rule leads to evolution of cooperation). The state-of-the-art lacks a comprehensive picture of the dependence of the emergence of cooperation on model parameters like network topology, interaction game, state update rules and initial fraction of cooperators. We perform a thorough study of the phenomenon of evolution of cooperation using (a) a set of popular categories of networks, namely, grid, random networks, scale-free networks, and small-world networks and (b) a set of cognitively motivated update rules. Our simulation results show that the evolution of cooperation in networked systems is quite nuanced and depends on the combination of network type, update rules and the initial fraction of cooperating agents. We also provide an analysis to support our simulation results.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Multiagent systems

General Terms

Experimentation

Keywords

Emergent behavior, Evolution of Cooperation, Repeated Games on Graphs

Cite as: The Evolution of Cooperation in Self-Interested Agent Societies: A Critical Study, Lisa-M. Hofmann, N. Chakraborty, and K. Sycara, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 685-692.

1. INTRODUCTION

The emergence of cooperation in a system of interacting self-interested agents has been studied in social science [3], evolutionary biology [13] and physics [18]. Examples of evolution of cooperation can be seen in natural systems including cellular structures like RNA [20], microbial organisms [6], animals [17], and humans [2]. The interaction model is a Prisoners' Dilemma (PD) game, which is a well-known game-theoretical model to study social dilemma situations among rational, self-interested, utility maximizing agents. Each player has two actions (or strategies): cooperate and defect. Defect is a dominant action, i.e., the payoff for playing defect is higher irrespective of the opponent's action. Thus, in the one-shot version, both players should always choose to defect, which is the only Nash equilibrium of the game. However, the equilibrium is not Pareto-efficient, i.e., both players would be better off by choosing to cooperate. Hence, a *social dilemma* arises. This contradicts the observed phenomenon of cooperation in human experiments. Repeated interaction was shown to be one of the factors for evolution of cooperation in two-agent PD games [3]. However, in multiagent interaction, evolutionary game theory has shown that in a big (or infinite) population, if players have repeated random encounters, a population of cooperators cannot resist invasion by defectors, and thus cooperation cannot survive. Defection is the only evolutionary stable strategy. Subsequently, it was shown that if the interaction between the players has a network structure, cooperation emerges and can be sustained. This phenomenon was initially shown (via multiagent simulations) for grids [13] and later for scale-free networks [16] or graphs with adaptive topology [22]. In this paper, we perform simulation studies to critically examine the following question: *Under what conditions does cooperation emerge in a network of interacting agents?*

A repeated PD game proceeds in multiple rounds. In each round, an agent plays the game with all its neighbors and earns the aggregate payoff of all the games. The agent uses the payoff of its neighbors (including self) to decide the action for the next round. Nowak and May [13] used this model to show evolution of cooperation in a system of agents organized in a grid and used *imitate-best-neighbor* as a deterministic update rule. Subsequent work showed emergence of cooperation with the agents organized according to different network structure and using different update rules (see [16], and [18] for a review). However, these results are based on particular combination of interaction game, network model, and state update rules. The state-of-the art lacks a compre-

hensive picture of the dependence of the emergence of cooperation on the model parameters like the network topology, the update rules and the initial fraction of cooperators.

Our motivation for studying evolution of cooperation is two-fold. First, we want to understand the reasons behind evolution of cooperation in self-interested agents in natural systems. The complementary sociological question of emergence of conflict in a society of humans can also be studied in the same framework [11]. The second motivation comes from the design of autonomous artificial multiagent societies (e.g., an autonomous robot colony operating on extra-planetary surfaces). Social dilemma situations where an individual robot objective is in conflict with the social objective may arise and it is impractical for a designer to foresee every possible situation. An alternate way is to design protocols that ensure cooperation among the agents in social dilemma situations. In this paper, we will not concern ourselves with the applications aspect. We will perform simulation studies to characterize the parameters and provide basic understanding of situations under which cooperation emerges in a multiagent society.

For the multiagent society, we assume simple agents that are myopic, of bounded rationality, organized according to a graph with fixed topology and repeatedly play a PD game with each other. We study evolution of cooperation using (a) a set of popular categories of networks, namely, grids, random, scale-free, and small-world networks and (b) a set of cognitively motivated state (or action) update rules. The rules we use are both deterministic and stochastic in nature. Cooperation is said to evolve in a society if the initial fraction of cooperators is lower than the final fraction of cooperators. We show by simulation that the phenomenon of evolution of cooperation is quite nuanced and depends on the graph topology, the initial fraction of cooperators, and the state update rule. In particular, we show that using the imitate-best-neighbor rule (as used in [13]), cooperation evolves in grids or scale-free networks for $d > 0.3$ but not in random or small-world networks (where d is the initial fraction of cooperators). We also show that the stochastic update rule used in [16] works only for scale-free networks and not for other types. This is significant because it shows that using the same update rule may not work across all network topologies. The update rules that show uniform performance irrespective of the network topology are (a) imitate the best action in your neighborhood (BS) and (b) win stay, lose shift (WSLS). BS ensures emergence of cooperation for $d \geq 0.6$, whereas WSLS ensures evolution of cooperation for $d \leq 0.5$. Moreover, for a given network type WSLS leads to the same final fraction of cooperators irrespective of the initial fraction. Although WSLS was shown to be a winning strategy update rule in two-player games, to the best of our knowledge, this rule has not been used in multiplayer repeated games. We believe that our characterization of the conditions under which cooperation evolves gives a more complete picture about emergence of cooperation for repeatedly interacting networked agents. This is the primary contribution of our work.

This paper is organized as follows: In Section 2, we discuss the relevant literature and in Section 3, we define our mathematical model including the network structures and state update rules used in the paper. In Section 4, we describe our simulation setup and in Section 5 we present our findings. In Section 6 we present our conclusions and outline future research directions.

2. RELATED WORK

The literature on using repeated games for studying evolution of cooperation among self-interested agents, can be classified according to the the number of players, interaction game model, and the interaction structure of the players. Game play can be between two players or between multiple players. In the multiagent setting, the agents may form an unstructured population where players randomly interact with each other or there may be structured interaction between them. For structured interaction, the interaction network may be of fixed or variable topology [22]. Both PD and the snowdrift game [7] has been used as the interaction model between agents, although (arguably) the PD game is more popular. For two-player games Axelrod first showed in a computer tournament that state update rules that rely on reciprocal altruism, such as *tit-for-tat*, where a player starts with cooperation and then imitates its opponent, can lead to the evolution of cooperation [2]. Similar results have been obtained for *win-stay, lose-shift* [12]. In this work, we concentrate on repeated PD games in population of agents whose interaction network has a fixed topology. Therefore we will restrict our review to repeated PD games in graphs.

Nowak and May [13] first demonstrated that cooperation evolves for memoryless agents playing repeated PD game with their 8 neighbors in a two-dimensional grid. The update rule used was deterministic imitate-best-neighbor. They show that cooperation evolves over a wide range of payoff parameters and the final fraction of cooperators is independent of the initial fraction. They also note that cooperators and defectors exist in clusters (or patterns) and the patterns are unstable against small random perturbations [10]. Subsequent research has tried to replicate the evolution of cooperation in different networks and using different update rules [16, 19, 5, 1]. A comprehensive review on evolutionary games in graphs including repeated games in graphs is given in [18].

Santos et al. [16] investigate the influence of Barábasi-Albert scale-free networks on cooperative behavior in comparison to complete, single-scale and random scale-free networks and show a clear rise in the final fraction of cooperators with the heterogeneity of the degrees. The update rule is a stochastic imitation rule (rule SA in Section 3). Tang et al. [5] demonstrates that there exist optimal values of the average degree for each kind of network leading to the best cooperation level. They test random, Barábasi-Albert scale-free, and Newman-Watts small-world graphs under a stochastic update rule that depends on the normalized payoff difference to a randomly chosen neighbor. They show via simulation that there is an optimal degree for cooperation in each network which is quite constant over a certain range of T (payoff for defecting when the opponent cooperates). Cooperation is highest for small average degrees ranging from 3 to 8. However, this is only done for an initial fraction of 0.5, a stochastic update rule, and 10 different realizations of the particular graph. The results on evolution of cooperation have been usually obtained on different networks using a particular state update rule. The concern that changing the state update rule may affect the evolution of cooperation has not been addressed in the literature. Therefore, we study the evolution of cooperation across a variety of networks with different update rules.

There has also been work on repeated PD games in graphs with variable topology [22, 8]. In [22], the initial graph is assumed to be a random network and the agents are allowed to (stochastically) break links with their neighbors playing defect and form a new link with their neighbor's neighbor. The authors show that this boosts cooperation in the society. In this paper, we do not consider variable graph topology. A study similar to ours can be done for networks with variable topology and we keep this as a future work.

3. PROBLEM MODEL

3.1 Network Models

The agent interactions can be encoded as an undirected graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ are a set of n nodes (or agents) and $E \subseteq V \times V$ is a set of edges. The graph topology is fixed throughout the game. Two agents v_i and v_j are neighbors if $(v_i, v_j) \in E$. $\mathcal{N}(i) = \{v_j | (v_i, v_j) \in E\} \subset V$ is the set of v_i's neighbors and $|\mathcal{N}(i)|$ is the *degree* of node v_i. $\mathcal{N}^+(i) = \mathcal{N}(i) \cup \{v_i\}$.
We use four different graph types for the simulations:
Scale-free network: In a scale-free graph, the distribution of node degree follow a power law, $N_d \propto d^{-\gamma}$, where N_d is the number of nodes of degree d and $\gamma > 0$ is a constant (typically $\gamma \in [2, 3]$). We use the Barabási-Albert model with average degree 4 [4].
Small-world network: A small-world graph shows a high clustering coefficient (as defined in [21]) and a short average path length. We use the Watts-Strogatz model with average degree 4 [21]. First, a ring is built and each node is connected to the 2 neighboring sites on each side. Then, links are randomly released and reconnected to other nodes. We set the rewiring probability to 0.2, which leads to an average degree of roughly 4.
Random network: A network where a link between nodes is set with a predefined probability p. The probability that a vertex v_i has k_i neighbors follows a binomial distribution $B(n-1, p)$. For large n and $p \leq 0.05$ the degree distribution can be approximated by a Poisson distribution $Prob(k_i = k) = \exp(\lambda) \cdot \frac{\lambda^k}{k!}$ with $\lambda = n \cdot p$. We set $p = 0.05$ to ensure connectedness. The clustering coefficient is usually low.
Grid: A grid is a two-dimensional lattice where each inner player has 4 neighbors, each boundary player 3 and each corner player 2. The clustering coefficient is 0.

3.2 Repeated PD Games in Networks

A PD game is a two-player game where each agent has two actions, $\mathcal{A} \widehat{=} \{\text{cooperate}(1), \text{defect}(0)\}$. The payoffs for two players are symmetric with the payoff matrix entries

	1	0
1	R	S
0	T	P

For a PD, $T > P > R > S$ holds and for repeated PD games $T + S < 2R$. We assume $R = 1, P = 0.1, S = 0$ with the incentive to defect T being the only parameter.

In a repeated PD game in a network, there are n-players that form the nodes of the graph and the game proceeds in rounds. Each round has two phases: (a) In the game playing phase the players play the PD game with all their neighbors with a fixed strategy and compute their total payoff. (b) In the strategy update phase, each player updates its strategies according to the same *action update rule*. Such a rule might be a function of the neighbors' states, payoffs and/or the agent's own state and payoff. In our model the action update rule is synchronous.

Let $s_i(t)$ denote the state of player i at round t. The total payoff, $p_i(t)$, is the sum of the payoffs of the separate games in player i's neighborhood $\mathcal{N}(i)$:

$$p_i(t) = \sum_{j \in \mathcal{N}_i} [Rs_i(t)s_j(t) + T(1 - s_i(t))s_j(t) + S(1 - s_j(t))s_i(t) + P(1 - s_i(t))(1 - s_j(t))]$$

3.3 State Update Rules and Convergence

In each round, the agents update their states according to a common state update rule. The rules that we use can be classified along two axes: innovative or non-innovative and deterministic or stochastic. Rules that use states already existing in the neighborhood are non-innovative (e.g., imitate-best-neighbor or imitate-best-strategy) whereas rules that can switch to a strategy not in their neighborhood are called innovative rules (e.g., win-stay, lose-shift). We use the following rules:

Imitate-best-neighbor (IB): Each agent imitates the action of the wealthiest agent (including itself) in the next round. If two or more players have the same payoff, the agent chooses randomly between them. The state update for agent i can be formalized as

$$s_i(t) = s_j(t-1) \text{ where } j = \arg \max_{k \in \mathcal{N}^+(i)} (p_k(t-1))$$

Imitate-best-strategy (BS): An agent copies the strategy that accumulates the highest payoff in its neighborhood. Each agent sums up the payoff of all cooperating as well as the payoff of all defecting neighbors including itself.
Let agent i play strategy s_1 in round $t-1$ and have q_i neighbors. We denote its neighbors playing strategy s_1 and i itself by G_1, where $|G_1| = n_1$. The neighbors playing s_2 are denoted by G_2, where $|G_2| = n_2$. It holds that $G_1 \cup G_2 = \mathcal{N}_i^+$ and $n_1 + n_2 = q_i + 1$. Let w be the probability of switching. The update rule in any round t is as follows:

$$w = \begin{cases} 1 & \text{if } \sum_{i \in G_1} p_{t-1}(i) < \sum_{k \in G_2} p_{t-1}(k) \\ 0 & \text{otherwise} \end{cases}$$

Win-stay, lose-shift (WSLS): In a multiplayer setting, a strategy is maintained only if the current payoff p is at least as high as in the former round. We need to introduce a short-term (one-round) memory in order to calculate the payoff difference. In our case, there are only two possible strategies s_1 and s_2. In any round t the update rule is

$$w = \begin{cases} 1 & \text{if } p_{t-1} < p_{t-2} \\ 0 & \text{if } p_{t-1} \geq p_{t-2} \end{cases}$$

Stochastic imitate-best-neighbor (stIB): This rule represents a stochastic version of the IB rule. Each agent i picks the best neighbor j in $\mathcal{N}^+(i)$ and imitates its strategy with a probability w depending on the payoff difference $\Delta p_{i,j}$: $w = 1/(1 + exp(-\beta \Delta p_{i,j}))$. In test runs, $\beta = 0.75$ gives a

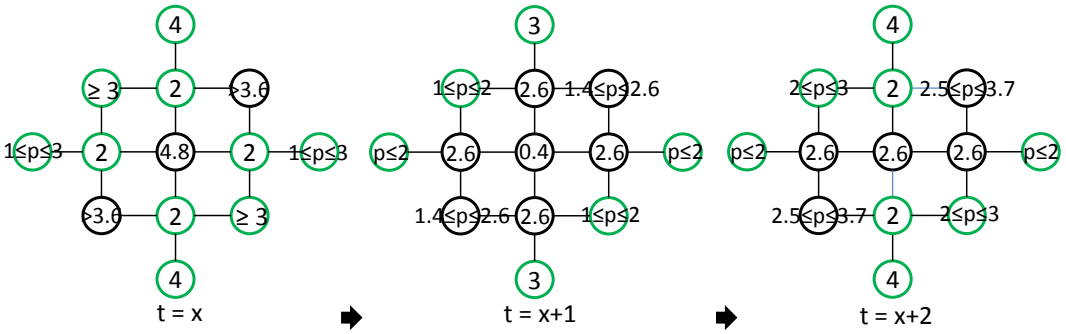


Figure 1: A typical oscillatory state with the IB rule. Green nodes represent cooperators, black defectors, the number in a node corresponds to its payoff p. We only display relevant part of the grid. The state of period x will be repeated in period $x + 3$.

reasonable trade-off between the payoff difference and the probability to switch.

Stochastic imitate-best-strategy (stBS): The strategy that yields a higher payoff in a neighborhood is imitated with probability of its total payoff divided by the total payoff in the neighborhood. Otherwise, the player keeps its current strategy.

Stochastic win-stay, lose-shift (stWSLS): If a player's payoff deteriorates in round t, the player will switch strategies with a probability w depending on the difference of its current and last payoff $\Delta p_{t-1,t}$: $w = 1/(1 + exp(-\beta \Delta p_{t-1,t}))$ with $\beta = 0.75$.

Stochastic imitate-random-neighbor (SA): The rule is described and used in [16]: for each i, one neighbor j among all k_i neighbors is picked at random. Only if $p_j > p_i$, i imitates j's strategy with probability $(p_j - p_i)/k_> D_>$, where $k_> = \max(k_i, k_j)$ and $D_> = \min(T, 1) - \max(S, 0)$.

The stochastic rules that we use are counterparts of our deterministic rules (except for the SA rule, taken from [16]). The deterministic IB rule is taken from [13] and the WSLS is taken from [12]. These rules are simple heuristics that have been shown to be used by humans for decision making under certain circumstances. Note that we have not used the best response strategy because for our model it always leads to evolution of defection among all agents. The imitating rules also have an evolutionary biology interpretation. Instead of a player updating its state, we can say that in each round, neighbors are competing against each other for occupying the empty node in their middle. The player with highest payoff, i.e., *fittest* player wins and its strategy gets replicated (with some probability in stochastic updates).

Steady States: Since our repeated game model is a dynamical system and we will use simulations to study the evolution of cooperation it is important to understand the convergence properties of the system to design appropriate stopping criterion for simulations. Note that the all-cooperate and all-defect solutions are trivial steady states for all the state update rules. For deterministic rules, a steady state is reached if the concatenated strategy vector (state vector) of all agents repeats itself $\mathbf{s}_t = \mathbf{s}_{t-1}$. For our system, we can show that we may not reach a steady state. Figure 1 shows a simple example demonstrating that oscillations can occur in a repeated PD game in grids. Figure 1, shows the part of a grid network where players will keep changing strategies. For some boundary nodes we give ranges for their payoff. As long as these payoff requirements are fulfilled, we do not have to consider any further players. Simple calculation shows that the system will oscillate.

For stochastic rules, the notion of convergence is different as the state vector represents the realization of the current *probability to cooperate* of each player. Thus, in the strict sense, this probability has to stay within a certain range for each player over time to ensure convergence. From our results we see that this does not happen as the current probability in a particular round t does not depend on the one of the round $t-1$, but on the realization of this probability. A more simple criterion could be that running averages of f_c for each player do not change much.

4. SETUP OF THE SIMULATIONS

We test three deterministic and four stochastic update rules on four different networks: scale-free, small-world, grid and random networks. Three stochastic rules are counterparts of the deterministic versions, the fourth comes from the literature. We call the combination of a particular update rule, graph and initial fraction of cooperators a *setting*. For each setting, we perform 100 runs, each with a different realization of the graph if there is a stochastic component in the setup (except for grids, where there is no stochastic component). We compute the average final fraction of cooperators, f_c, over all the runs and also compute the standard deviation over the final fraction of cooperators, σ. For all the results that we present σ is quite low except where we explicitly mention. For a given initial fraction of cooperators, each player is randomly assigned the action cooperate or defect such that the ratio of total number of cooperators to defectors is equal to the given fraction.

Stopping criterion for simulations: For deterministic rules, we simulate for a maximum of $t = 500$ rounds. If the simulation converges, we take the last state as final result. If not, we average over the last five rounds. For stochastic update rules, we simulate over $t = 5000$ rounds and average over the last 100 rounds. The number of rounds to average over was heuristically determined after finding that the deviation over the last few runs usually is very low.

We use values of ($T \in \{1.1, 1.2\}$) for most of our simulations, except for grids. Higher values of T will give rise to more defection. We test several initial fractions of cooperators d ($d \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$). In some cases we test additional values in order to determine more exact thresh-

olds or to point out differences between certain settings.

In forming our graphs we ensure that all of them are connected. We simulate on graphs with 750 and 1000 nodes. The IB rule in random networks is the only setting where we find the scaling of a network to change results for f_c, because the average degree changes with the number of nodes and the number of neighbors matters in imitating rules. Therefore, simulations of this setting have be to made with caution and be tested for different levels of n and p. In all other settings, differences in f_c between $n = 750$ and $n = 1000$ are smaller than 5%. Research about the influence of the average degree states that an increasing average degree usually leads to less cooperation [14].

5. SIMULATION RESULTS

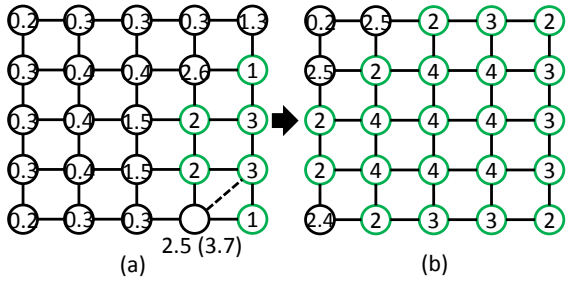


Figure 2: In a grid where the dotted link does not exist, the defector only yields a payoff of 2.5 and cooperation will spread until it reaches state (b) with $f_c = 84\%$ after 5 rounds. Adding only one link to (a) and therefore increasing the clustering coefficient increases the defector's payoff to 3.7 and leads to 100% defection after 3 rounds ($T = 1.2$).

We want to have a general insight about conditions under which there is *evolution of cooperation*, i.e., the final fraction of cooperators f_c is higher than the initial fraction d. Table 1 gives the general findings. Most of the literature focuses on whether cooperation emerges at all (by looking at the final fraction of cooperators), whereas we want to focus on the relation of the final fraction of cooperators to the initial fraction, which has not received much attention so far. Comparing the obtained results of $T = 1.1$ and $T = 1.2$, there is hardly any difference except for random networks. We will come back to this issue in 5.4. Thus, we only display results for $T = 1.2$ in Figure 3. In grids, there is no difference for the IB rule, because the interaction structure is very simple. Possible constellations of payoffs and strategies are very limited and stay the same for these two levels of T. If we set $T = 1.3$ in grids, one new constellation that helps defectors at the first glance actually leads to the collapse of bigger clusters of defectors and therefore to higher final fractions of cooperators. From simulations we see that this development abruptly ends starting from $T = 1.4$, where defectors are better off in most constellations and f_c drops drastically.

5.1 Scale-Free Networks

In scale-free networks, all the state-update rules that we study show emergence of cooperation for different ranges of the initial fraction of cooperators. From Figure 3, we see that the IB rule leads to evolution of cooperation for both deterministic and stochastic versions. For the deterministic IB rule, we further observe, that even though most of the runs converge, there is a large standard deviation $\sigma = 0.15$ to 0.46. This is because the distribution of f_c is bimodal: either f_c drops to zero or reaches very high values. Averaging over 100 runs gives the obtained high levels. It has already been pointed out that the important factor for cooperation or defection in scale-free networks is the behavior of high-degree nodes [16]. If a high degree node defects it can exploit all linked cooperators and gain a high payoff. Imitating the best, all its neighbors will switch to defection. From this state onwards it is not likely that defectors find a wealthier cooperator as cooperators surrounded by defectors do not obtain payoffs. Additionally, a high-degree node still accumulates relatively high payoffs even for defect-defect links because of sheer number. In simulations where we apply a normalization of payoffs by the number of neighbors a drastic drop of f_c is observed and there is no evolution of cooperation. Another indicator revealing the importance of the heterogeneity in degrees is that we find the highest degree node cooperating in 100% of the runs with high final levels and defecting in all runs with a very low final percentage of cooperators. If there are cooperators in the cases with low f_c, they usually occur in a cluster around a wealthy cooperator.

For the SA rule used by Santos et al [15], we find a higher final fraction of cooperators than with any IB rule in scale-free networks. Like the deterministic IB rule, in this case, the standard deviation σ is very high, but unlike the IB rule the distribution is not bimodal. Although the SA rule performs well for scale-free networks, it does not lead to any cooperation in grids and random networks and only small to levels in small-world graphs.

5.2 Grids

In grids we find high cooperation rates with the IB rule. A closer look at the dynamics shows that clustering is the crucial factor for the success of cooperators in grids as already pointed out in [13]. A cluster of cooperators is a set of cooperating nodes that are connected to each other. Boundary players of clusters of cooperators are the nodes that are also linked to defectors. In settings with low values of T, a 2×2 cluster of cooperators already leads to propagation of cooperation through the whole network. There are several possibilities for defectors to survive, e.g., in corners, in a line of 4 players or in several spatial structures. However, too big clusters of defectors become unstable at low levels of T. For clusters of cooperators, the well-defined interaction with defecting neighbors originating from the typical grid structure, is helpful as defectors cannot exploit cooperators in the middle of clusters. Cooperators on the boundary will not turn into defectors as long as the neighboring defector has less than four cooperating neighbors. The cooperator in the middle of the cluster is the wealthiest player and backs up the boundary cooperators.

However, note that the stochastic IB rule does not lead to cooperation in grids. The success of cooperators depends on the formation of clusters. However, cooperators in the middle of clusters may randomly turn into defectors. A single defector surrounded by only cooperators can turn all its neighbors into defectors. The above intuition is also true for the SA rule and hence it does not lead to cooperation.

Table 1: Summary of the evolution of cooperation with different update rules and networks. *Yes* denotes that there is evolution of cooperation with the range of the initial fraction of cooperators d for cooperation to emerge given in parentheses. Thresholds for d are indicative and not exact.

Rule/Graph	Scale-free	Small-world	Grid	Random
IB	yes ($d > 0.3$)	no	yes ($0.3 \leq d < 0.9$)	no
BS	yes ($d \geq 0.6$)			
WSLS	yes ($d \leq 0.5$)			yes ($d \leq 0.7$)
stIB	yes ($d > 0.5$)	yes ($d > 0.5$)	no	no
stBS	yes ($d > 0.5$)	yes ($d > 0.2$)	yes ($d > 0.2$)	yes ($d > 0.7$)
stWSLS	yes ($d < 0.45$)	yes ($d < 0.5$)	yes ($d < 0.45$)	yes ($d < 0.65$)
SA	yes	no	no	no

However, for the BS rule, a single defector surrounded by cooperators cannot destroy the cluster and so the stochastic BS rule leads to evolution of cooperation for $d \geq 0.2$. In fact, the stochastic version outperforms the deterministic BS rule. The propagation of defection by single defectors can also account for the fact that we do not find evolution of cooperation in grids with IB for a very high initial fraction of cooperators ($d = 0.9$). Defectors most likely do not appear in clusters and can therefore exploit all cooperating neighbors at once, which gives them a high payoff and leads to defection in their neighborhoods.

5.3 Small-World Networks

In Table 1 we see that there hardly is emergence of cooperation in small-world networks with the IB rule. This is especially interesting because the main graph features as the average degree and its standard deviation are almost the same as in grids, where the IB rule leads to cooperation. To discover reasons for the differences we have to look at the clustering coefficient c. c is very high in small-worlds ($c \approx 27\%$) in comparison to grids, where $c = 0$. We have seen before that well-defined boundaries between groups of cooperators and defectors in grids help to propagate cooperation. The Watts-Strogatz model constructs a small-world graph starting from a ring. The rewiring process creates shortcuts between different neighborhoods and can turn inner players into boundary players if the shortcut links them to defectors. Thus, some small clusters of cooperators that would have grown in grids cannot grow in small-worlds. Figure 2 gives an example how a slightly higher clustering coefficient leads to $f_c = 0$ instead of $f_c = 84\%$ for $c = 0$.

The final fraction of cooperators usually slightly drops from the initial fraction in small-worlds. Even if f_c increases slightly over the starting point for medium levels of d, we do not consider this as evolution of cooperation because the standard deviation ranges from 0.04 to 0.16. The stochastic BS rule yields 100% cooperation with small initial fractions and turns out to be the most successful rule in small-worlds. Starting from a medium level of d, the deterministic version leads to evolution of cooperation, too. The deterministic and the stochastic WSLS yield a medium level of cooperation.

5.4 Random Networks

Random networks are the only network where simulations with $T = 1.1$ and $T = 1.2$ yield different results. However, we do not consider results for $T = 1.1$ to be reliable because of high standard deviations and low convergence rates. Therefore, we discuss the results for $T = 1.2$ which show 100% convergence and a lower σ.

The WSLS rule yields the highest levels of cooperation in random graphs compared to other networks. The IB rule hardly leads to cooperation in random networks. The results for the BS rule are drastic, as f_c turns out to be either 0 or 1. We see from the simulation results that the jump occurs between $d = 0.55$ and $d = 0.6$. We will give the explanation in the discussion of the BS rule. The stochastic version yields the best result for random graphs.

5.5 The Imitate-Best-Strategy Rule

In all settings, we find that the BS rule does not lead to evolution of cooperation for any initial fraction, $d \leq 0.5$, whereas it takes place for all $d \geq 0.6$. This phenomenon is extraordinarily strong in random networks, where the final fraction of cooperators, f_c jumps from 0 to 1. Further simulations indicate that there is a threshold for the evolution of cooperation that occurs between $d = 0.55$ and $d = 0.65$.

We now estimate analytically the threshold value of the parameter d for cooperation to emerge in scale-free graphs, small-worlds, and grids, under some simplifying assumptions. In our model, each player i has 4 neighbors on average. Let x be the fraction of cooperating neighbors of a node i and let each neighbor in $\mathcal{N}(i)$ have 4 neighbors with y the fraction of cooperators in their neighborhoods (i's two-step neighborhood) a constant (but need not be the same as x). We assume that x is representative of the whole network, i.e., the fraction of cooperators in the network at that round is x. Since the average degree is 4, we assume that x can take the values $0, 0.25, 0.5, 0.75, 1$. Recall that in this rule, a player decides which strategy to take according to the wealthiest strategy in its neighborhood. In general, i will cooperate if

$$x(yR + (1-y)S) > (1-x)(yT + (1-y)P) \qquad (1)$$

which is 0.56(0.55) for $x = y$ and $T = 1.2(1.1)$. From Equation 1, we also find that i will always prefer to play defect for $x \leq 0.5$ (irrespective of the value of y). However, for $x = 0.75$, i will play cooperate in every case where $y \geq 0.25$ (which is very likely, as x is representative of the whole network). Thus, from this simple analysis, we predict a threshold value of $d = 0.55$ for cooperation to emerge. Note that the argument above accounts for a simple average-case because it assumes the degree of each node to be 4 and an even distribution of cooperators in all two-step neighborhoods. As the actual x and y for a given neighborhood can differ from the average we find our simulation threshold to be slightly different from the predicted value of $d = 0.55$.

For random networks also, any agent i will cooperate if Equation 1 holds, i.e., an agent i will cooperate for $x \geq 0.56$

when $x = y$ and $T = 1.2$. Since the average degree is not 4 the calculation of the values of x for which cooperation emerges is more complicated for $x \neq y$. Note, that the number of cooperators in i's neighborhood follows a binomial distribution #coop $\sim$ bin$(n-1, d)$. Using Equation 1, for a given x, we can calculate the value of y required for i to be cooperating. Thereafter using the binomial distribution, we can compute the probability that such a fraction y will exist in i's two-neighborhood. For example, if $x = 0.47$, y should be ≥ 0.58. However the probability that $y \geq 0.58$ is equal to 0.046. Thus it is unlikely for agent i to play cooperate. However, for any $x \geq 0.56$ cooperation is very likely. For example, for $x = 0.61$, y should be more than 0.47 and the probability for $y \geq 0.47$ is 0.93. Thus, in this case also a threshold value of $d = 0.56$ would ensure cooperation, which is in good agreement with our simulations.

5.6 The Win-Stay, Lose-Shift Rule

Although the WSLS rule was shown to perform very well in two-agent settings it has not been investigated in multiplayer settings. An interesting aspect of WSLS is that for every network, it leads to the same f_c irrespective of the initial d. However, the actual value of f_c reached depends on the type of network. Another surprising aspect is that WSLS always leads to evolution of cooperation, if $d \leq 0.5$ and is the only strategy to do so across all the types of networks studied. We note that the update for an agent depends on the own payoff over time and therefore indirectly on strategy distributions of the neighbors. Furthermore, the rule is innovative, such that defectors surrounded by defectors are still able to change to cooperation, which is never possible in imitating rules. We found examples of how parts of a network can easily turn from all-defection to all-cooperation and vice versa in several time steps only.

We note that for the WSLS strategy most of the simulations do not converge. For the deterministic WSLS, we do not find convergence except in random networks for $T = 1.2$, where all runs converge (usually within 20 rounds). However, even though the runs do not converge, the standard deviation usually is lower than 0.016. Thus, we can have reasonable confidence about the correctness of our findings. We note that this update rule is especially helpful in systems where one does not have an influence on the initial fraction of cooperators but wants to ensure a medium level of cooperation. The stochastic WSLS yields slightly higher levels of cooperation in grids, scale-free, and small-world networks and here also the final fraction of cooperators is constant (independent of the initial fraction). Here, the standard deviation is lower than 0.01.

6. CONCLUSIONS

In this paper, we performed a comprehensive simulation study of the phenomenon of evolution of cooperation in self-interested multiagent societies. Our research shows that general statements on evolution of cooperation in networked multiagent systems cannot be made. The emergence of cooperation depends on the type of network, the state update rule, and the initial fraction of cooperators. We find a high dependency of final results on the initial fraction especially in imitating, non-innovative rules. We observe that the evolution phenomenon do not depend on the size of the network as long as the network is large enough to show its typical properties and crucial network parameters do not change with the number of nodes. Our main findings are as follows:

- In scale-free networks, almost all the state update rules lead to evolution of cooperation. However, the deterministic imitation rule and stochastic imitation rule of [16] perform better.
- For small-world networks stochastic BS performs best.
- For grids the deterministic IB performs the best and most stochastic rules (except stochastic BS) do not perform well.
- For random networks WSLS performs the best.
- WSLS gives the interesting result that for every type of network we studied, the final fraction of cooperators reaches a constant value. Further, this is the only rule that ensures evolution of cooperation for low initial fraction of cooperators. This result holds across all types of networks.
- The BS rule also has the interesting property of supporting evolution of cooperation above a threshold value of initial fraction of cooperators across all networks.

We also find that stochastic versions of deterministic rules usually perform slightly better. The final results still highly depend on the network: e.g., rules that work very well in scale-free graphs do not have to be successful in grids. Furthermore, results for different stochastic rules can vary greatly in the same setting. In most cases we find them to yield similar results as the deterministic versions.

Future Work: In this paper we have considered the PD game as an interaction model with a fixed topology of interaction. An important future direction of research is to relax the assumption of fixed topology. Although versions of this problem has been studied [22], there is no restriction placed on the topology of the graph, except that it remains connected. An interesting extension would be to study the evolution of cooperation in variable topology graph where the statistical properties of the graph is maintained (i.e., a scale-free graph remains scale-free). Another future research agenda is to give a broad understanding of rules and networks for emergence of cooperation in the Snowdrift game. A first glance at pilot simulations also shows different behavior for different settings [9].

From the theoretical perspective there are a few interesting directions that can be pursued. The results obtained from WSLS seem to indicate some universal underlying phenomenon for the rule. Theoretical understanding of why there is a uniform final fraction of cooperators for WSLS in a given type of network is an important research direction. Moreover, here we have prescribed rules and tried to analyze whether the rules lead to evolution of cooperation. Designing a rule that guarantees a certain level of cooperation irrespective of the network topology is an important problem that we wish to pursue.

Acknowledgements

This research was funded by ONR MURI grant N000140811186 and ARO MURI grant W911NF-08-1-0301.

7. REFERENCES

[1] G. Abramson and M. Kuperman. Social games in a social network. *Phys. Rev. E*, 63(3):030901, Feb 2001.

[2] R. Axelrod. *The evolution of cooperation*. Basic Books, New York, NY, 1984.

[3] R. Axelrod and W. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[4] A.-L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[5] C.-L. Tang, W.-X. Wang, and X. Wu. Effects of average degree on cooperation in networked evolutionary game. *Eur. Phys. J.*, 53(3):411–415, 2006.

[6] B. J. Crespi. The evolution of social behavior in microorganisms. *Trends Ecol. Evol.*, 16:178–183, 2001.

[7] M. Doebeli. Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature*, 428:643–646, 2004.

[8] H. Ebel and S. Bornholdt. Coevolutionary games on networks. *Phys. Rev. E*, 66, 2002.

[9] L.-M. Hofmann. The evolution of cooperation in self-interested agent systems. Master's thesis, Karlsruhe Institute of Technology, 2011.

[10] Y. F. Lim, K. Chen, and C. Jayaprakash. Scale-invariant behavior in a spatial game of prisoners' dilemma. *Phys. Rev. E*, 65(2), 2002.

[11] L. Luo, N. Chakraborty, and K. Sycara. Modeling ethno-religious conflicts as prisoner's dilemma game in graphs. *Computational Science and Engineering, IEEE International Conference on*, 4:442–449, 2009.

[12] M. Nowak and K. Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364, 1993.

[13] M. A. Nowak and R. M. May. Evolutionary games and spatial chaos. *Nature*, 359:826, 1992.

[14] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak. A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505, 2006.

[15] F. C. Santos and J. M. Pacheco. Scale-free networks provide a unifying framework for the emergence of cooperation. *Physical Review Letters*, 95(9), 2005.

[16] F. C. Santos, J. M. Pacheco, and T. Lenaerts. Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. USA*, 103:3490–3494, 2006.

[17] J. M. Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(2):15–18, 1973.

[18] G. Szabo and G. Fath. Evolutionary games on graphs. *Physics Reports*, 446(4-6):97–216, 2007.

[19] M. Tomassini, E. Pestelacci, and L. Luthi. Social dilemmas and cooperation in complex networks. *International Journal of Modern Physics C*, 18:1173–1185, 2007.

[20] G. J. Velicer. Social strife in the microbial world. *Trends in Microbiology*, 11(7):330–337, 2003.

[21] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[22] M. G. Zimmermann and V. M. Eguíluz. Cooperation, social networks, and the emergence of leadership in a prisoners dilemma with adaptive local interactions. *Phys. Rev. E*, 72(5):056118, 2005.

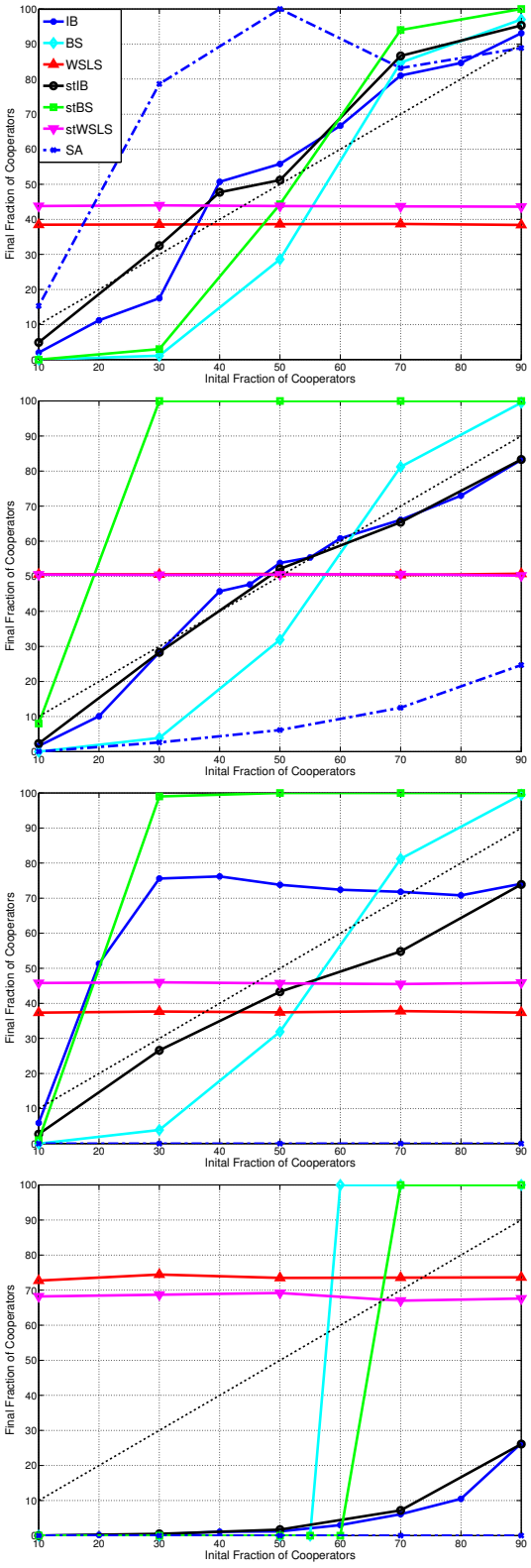


Figure 3: Final fraction of cooperators for $T = 1.2$, $n = 750$ and four different networks: scale-free, small-world, grid and random networks (in this order). Evolution of cooperation occurs where the final fraction is above the dotted black line.

A Model of Norm Emergence and Innovation in Language Change

Samarth Swarup
Network Dynamics and Simulation Science Lab, Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA, USA
swarup@vbi.vt.edu

Andrea Apolloni
Institut des Systèmes Complexes Rhône-Alpes, and Laboratoire de Physique, École Normale Supérieure de Lyon
69007 Lyon, France
andrea.apolloni@ens-lyon.fr

Zsuzsanna Fagyal
School of Literatures, Cultures, and Linguistics, Department of French, University of Illinois at Urbana-Champaign
Urbana, IL, USA
zsfagyal@illinois.edu

ABSTRACT

We analyze and extend a recently proposed model of linguistic diffusion in social networks, to analytically derive time to convergence, and to account for the innovation phase of lexical dynamics in networks. Our new model, the degree-biased voter model with innovation, shows that the probability of existence of a norm is inversely related to innovation probability. When the innovation rate in the population is low, variants that become norms are due to a peripheral member with high probability. As the innovation rate increases, the fraction of time that the norm is a peripheral-introduced variant and the total time for which a norm exists at all in the population decrease. These results align with historical observations of rapid increase and generalization of slang words, technical terms, and new common expressions at times of cultural change in some languages.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Algorithms, Experimentation, Theory

Keywords

Social Simulation, Lexical Innovation, Norms, Degree-biased Voter Model

1. INTRODUCTION

Multiagent modeling and analysis is being increasingly applied to the study of language change [1; 3, e.g.]. In this view, a language is seen as an emergent phenomenon from the interactions between a population of communicating agents, and change in language is driven by linguistic factors, such as frequency of use, and social factors like social network structure and popularity. Computational modeling is especially relevant with respect to language change; it provides tools to explore the large-scale consequences of small incremental changes that are typically studied empirically at the individual level or the level of small communities.

One of the foundational questions in this respect is, how do linguistic norms emerge, and how do they change? Recently, Fagyal et al. [9] proposed a model, known as the degree-biased voter model (DBVM), to study the role of network structure and popularity in the spread of linguistic variants. They showed that the DBVM brings together in one model two separate factors in the emergence of linguistic norms: the role of network positions, in particular the contribution of central and peripheral agents referred to as *hubs* and *loners*, and the role of popularity in determining which linguistic variants are preferentially copied and propagated. These factors had been separately attested to in the empirical sociolinguistic literature [5; 16; 18; 19, e.g.], but never combined into a model of norm emergence before.

However, their model left some important questions unaddressed:

- From an analytical perspective, how long does it take for a norm to emerge, i.e., what is the *time to convergence*?
- From a sociolinguistic perspective, their model does not address the *innovation phase* of the dynamics. Who creates the new variants that go on to become norms?

The first question is relevant in that the time to convergence is directly related to the time it takes to switch between norms, which would allow to investigate cycles of fashion and fad quantitatively. The second question is important for understanding diffusion dynamics at times of increased cultural contact when the innovation rate, for instance in the lexicon via borrowing or other means, is particularly high.

In the present work, we analyze the DBVM to derive time to convergence in terms of the size of the network. We also introduce an extension to the model to include innovation, and we numerically address the question of which network positions have an advantage in terms of generating new norms. Languages tend to be stable for long periods, and then change in bursts (typically triggered by large-scale social change). Our analysis and extension here, therefore,

Cite as: A Model of Norm Emergence and Innovation in Language Change, Samarth Swarup, Andrea Apolloni, and Zsuzsanna Fagyal, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 693-700.

combine to present a more complete model of linguistic dynamics. The extended model shows that in situations where the innovation rate is high, there tend to be multiple variants in competition in the network, and the period of time for which norms exist is lower. Further, many peripheral variants become the norm in the network. This behavior is congruent with qualitative observations of certain types of lexical change in French and possibly in other languages.

The rest of this paper is organized as follows: first we provide some linguistic context and a description of the DBVM, then we analyze the model to derive expressions for time to convergence. This is followed by a discussion of the DBVM dynamics, which we extend to include a parameter for innovation. We present some simulations to analyze this extended model, and show that at low innovation rates loners, i.e., peripheral agents who might influence others but do not listen to anyone else in the network, are more likely to produce the variants that later become norms in the network. As the innovation rate increases, however, both the fraction of time that the norm is a peripheral-introduced variant and the total time for which a norm exists at all in the population decrease. We discuss the relevance of this model to changes in French in the 19th Century.

2. THE DEGREE-BIASED VOTER MODEL

The importance of the social network in language change has been recognized for a long time. Bloomfield first suggested a thought experiment, where "every time any speaker uttered a sentence, an arrow were drawn into the chart pointing from his dot to the dot representing each one of his hearers. At the end of a given period of time, say seventy years, this chart would show us the density of communication within the community" [4, p. 46]. He hypothesized that these "lines of communication" and "the relative prestige of social groups" were the two main conditioning factors of "the spread of linguistic features" [4, p. 345].

Since then several researchers have studied the role of social networks in language change, by mapping out specific networks and recording the spread of linguistic variants and emergence of norms over these networks [7, 8, 15, 18, 19, 27]. Theorizing has focused on the roles of central "hubs" or "leaders" and peripheral "loners" or "lames" in the diffusion process.

These and other studies have resulted in two competing models of language change: Labov's and Eckert's work has supported the so-called *two-step flow of influence* model [14]. This model says that the centrally-connected leaders are responsible for introducing new variants into the local network, and that they themselves are primarily influenced by other leaders. On the other hand, work by the Milroys supports the *weak-tie model of influence* [11, 12]. In this model, it is the loosely-connected peripherals who introduce new variants into the local network, which they are able to do because they are relatively free from the regulatory influence of the local leaders, and more in touch with outsiders.

These models are at odds with each other because they posit different roles for the central and peripheral members of the network: hubs are considered agents of innovation in one, and conservative regulators in the other; peripherals are considered barely involved in the linguistic life of the network in one, and sources of novel variants in the other. The question then arises, how can these seemingly mutually contradictory explanations be reconciled?

Fagyal et al. proposed the degree-biased voter model to answer this question [9]. In this model, each node in the network (corresponding to an agent) is initialized with one variant of a linguistic variable. A variant can be phonetic, such as a flapped or fully released /t/ in the word "mittens" (number of variants, $v = 2$), or it can be a stylistic or contextual variant of a lexical item such as the French *voiture* as 'véhicule', 'char', 'tacot', or 'bagnole' ($v = 4$), etc. Further, edges in the network are directed, and an edge from node A to node B is interpreted to mean that A can copy B.

Once the simulation starts, at each time step, an agent copies a neighbor's variant with probability proportional to the neighbor's in-degree (the number of edges pointing to the neighbor). Thus, the probability that neighbor i will be chosen to copy from is,

$$P(i) = \frac{k_i^{in}}{\sum_j k_j^{in}}, \quad \forall i, j \in \mathcal{N} \tag{1}$$

where k_i^{in} is the in-degree of neighbor i, and $\mathcal{N}$ is a set consisting of all the neighbors of the current node. Note that the sum in the denominator is taken over all the neighbors of the node.

They showed that on a scale-free network with a small number of loners, this model results in the rapid emergence of norms, where nearly all the agents are in the same state (except the loners initialized in a different state). Loners remain fixed in their initial states because they have no links pointing to another agent (meaning they do not copy anyone else), but can still influence the dynamics within the network because they have (a very small number of) links pointing to them (meaning others can copy them). The presence of these loners makes the system a *driven*, or out-of-equilibrium, system. Thus the norms, while stable for long periods, will eventually be replaced by other norms, as some of the agents stochastically copy one of the loners in a different state, and this new variant gets propagated through the network. Interestingly, they showed that norms do not appear if degree-biasing is not present, which implies that norms emerge only when the system is *close* to equilibrium.

Their model points to a resolution of the debate over the two competing models of language change formulated by linguists by suggesting that both interpretations can be seen as valid at different instants of observation of the stochastic process of linguistic diffusion. Hubs essentially fulfill the roles of enforcing norms, but they also rapidly spread new variants when they themselves change their state. Loners tend to hold on to their variants, which then sometimes stochastically work their way up to the hubs because of short path lengths in a scale-free network, and thereby trigger changes in norms.

In this paper, we make this analysis more quantitative by analytically deriving the time-scale of norm emergence, as we now do.

3. ANALYZING THE DBVM

For analysis, we simplify the model slightly, by considering a system of N nodes connected through *un*directed links. We indicate with k the degree of each node and with n_k the fraction of nodes with degree k. We suppose that the degree distribution is a power law with exponent ν.

We also assume that the network is perfectly uncorrelated (a Molloy-Reed network [20]), which means that the proba-

bility of an edge between any two nodes is given by,

$$P(edge\ xy) = \frac{k_x k_y}{N^2}.$$

Thus, the probability that node x copies node y in the DBVM is given by,

$$\begin{aligned} P(x\ copies\ y) &= P(edge\ xy)\frac{k_y^\beta}{\sum_j P(edge\ xj)k_j^\beta}, \\ &= \frac{\frac{k_x k_y}{N^2} k_y^\beta}{\frac{k_x}{N^2}\sum_j k_j k_j^\beta}, \\ &= \frac{k_y^{\beta+1}}{\sum_j k_j^{\beta+1}}, \end{aligned}$$

where the summation is over all the nodes in the network. The coefficient β is the weight of the node. When $\beta = 0$ we obtain the standard voter model [23, 24], and when $\beta = 1$, we obtain the canonical DBVM. Now, $\sum_j k_j^{\beta+1} = N\sum_k k^{\beta+1} n_k$, where n_k is the fraction of nodes of degree k. We define $\mu_{\beta+1} = \sum_k k^{\beta+1} n_k$ as momentum of order $\beta+1$. Therefore,

$$P(x\ copies\ y) = \frac{k_y^{\beta+1}}{N\mu_{\beta+1}}. \tag{2}$$

We further assume that a node can have one of only two variants or states (i.e., $v = 2$), which we denote with +1 (state up) or -1 (state down). We indicate with ρ_k (correspondingly, 1-ρ_k) the fraction of nodes with degree k in state up (state down). At each iteration a node is chosen and one of its neighbours is picked up: if the states of the two nodes are different the first node copies the state of the second one with a probability based on the degree of the second one. The probability for a node with degree k and state down to switch state can be shown to be given by:

$$R_k(\rho_k) = n_k(1-\rho_k)\sum_j \frac{j^{\beta+1} n_j \rho_j}{\mu_{\beta+1}} = n_k(1-\rho_k)\omega_{\beta+1} \tag{3}$$

where $\omega_{\beta+1}$ is called the weighted magnetization. Correspondingly, the probability for a node with degree k and state up to switch is given by:

$$L_k(\rho_k) = n_k\rho_k \sum_j \frac{j^{\beta+1} n_j(1-\rho_j)}{\mu_{\beta+1}} = n_k\rho_k(1-\omega_{\beta+1}). \tag{4}$$

From now on we concentrate on the DBVM, which mean we assume $\beta = 1$ in what follows. The state of the system is defined at every time by the vector $\boldsymbol{\rho} = (\rho_1, \rho_2, \cdots, \rho_k)$ representing the fraction of nodes with degree k and state +1. We indicate with $P(\boldsymbol{\rho}, t)$ the probability that the system at time t is in the configuration $\boldsymbol{\rho}$. At each time step, the fraction ρ_k can change by a quantity $\delta_k = \frac{1}{Nn_k}$ representing the fact that one of the nodes has switched state. Indicating with $\delta t = 1/N$, the time evolution of the system is ruled by:

$$\begin{aligned} P(\boldsymbol{\rho}, t+\delta t) = P(\boldsymbol{\rho}, t) &+ \sum_k L_k(\rho_k+\delta_k)P(\rho_k+\delta_k, t) \\ &+ \sum_k R_k(\rho_k-\delta_k)P(\rho_k-\delta_k, t) \\ &- \sum_k \left(R_k(\rho_k)+L(\rho_k)\right)P(\rho_k, t) \end{aligned} \tag{5}$$

where $P(\rho_k \pm \delta_k, t)$ indicates the configuration differing for the state of one node with degree k, the first two sums in the right hand side indicate the system is reaching the configuration $\boldsymbol{\rho}$, while the last one indicates the departure from the configuration. Making a Taylor expansion with respect δ_k of equation (5) till the second order, we obtain the Fokker-Planck equation for the system:

$$\begin{aligned} \delta t\frac{\partial P(\boldsymbol{\rho}, t)}{\partial t} &= \sum_k \frac{1}{Nn_k}\frac{\partial}{\partial \rho_k}\left((L(\rho_k) - R(\rho_k))P(\rho_k, t)\right) \\ &+ \sum_k \frac{1}{2(Nn_k)^2}\frac{\partial^2}{\partial \rho_k^2}\left((L(\rho_k) + R(\rho_k))P(\rho_k, t)\right) \end{aligned} \tag{6}$$

The coefficients in the sums of (6) can be expressed in terms of the quantities ρ_k, n_k and ω_2 as:

$$\begin{aligned} (R_k(\rho_k) - L_k(\rho_k)) &= n_k(\omega_2 - \rho_k) \\ (R_k(\rho_k) + L_k(\rho_k)) &= n_k(\rho_k + \omega_2 - 2\rho_k\omega_2) \end{aligned} \tag{7}$$

Moreover we notice that since $\delta_k^2/\delta t = 1/(Nn_k^2)$, the second term in (6) is sub-leading and can be ignored, giving:

$$\frac{\partial P(\boldsymbol{\rho}, t)}{\partial t} = \sum_k (\omega_2 - \rho_k)P(\rho_k, t) \tag{8}$$

We use equation (8) to evaluate the time-evolution of the average value (on the ensemble of all the possible configurations $\boldsymbol{\rho}$) of ω_2 (indicated as $\langle\omega_2\rangle$):

$$\begin{aligned} \langle\omega_2\rangle &= \int \sum_k \frac{k^2 n_k \rho_k}{\mu_2} P(\boldsymbol{\rho}, t) d\boldsymbol{\rho} \\ \frac{d\langle\omega_2\rangle}{dt} &= \int \sum_k \frac{k^2 n_k \rho_k}{\mu_2}\frac{dP(\boldsymbol{\rho}, t)}{dt} d\boldsymbol{\rho} \\ &= \sum_{k,k'} \int \frac{k^2 n_k \rho_k}{\mu_2}\frac{\partial((\rho'_k - \omega_2)P(\boldsymbol{\rho}, t))}{\partial \rho'_k} d\boldsymbol{\rho} \\ &= -\int \sum_{k,k'} \frac{k^2 n_k((\rho_k - \omega_2)P(\boldsymbol{\rho}, t))}{\mu_2}\frac{\partial \rho_k}{\partial \rho'_k} d\boldsymbol{\rho} \\ &= \langle\omega_2\rangle - \langle\omega_2\rangle = 0 \end{aligned} \tag{9}$$

where we have integrated by parts and exploited the fact that the derivative term $\frac{\partial \rho_k}{\partial \rho'_k} = \delta(k, k')$, i.e., it is null when $k \neq k'$ and equal to 1 otherwise. The result implies that the average weighted magnetization is conserved for a fixed initial condition on the distribution $\boldsymbol{\rho}$. The existence of a conserved quantity, in our case ω_2, determines the exit probability, the probability of reaching a consensus state [24]. Moreover we notice that the conservation of the average weighted magnetization determines the evolution of the density ρ_k:

$$\begin{aligned} \langle\rho_k\rangle &= \int \rho_k P(\boldsymbol{\rho_k}, t) d\boldsymbol{\rho} \\ \frac{d\langle\rho_k\rangle}{dt} &= \int \rho_k \frac{dP(\boldsymbol{\rho}, t)}{dt} d\boldsymbol{\rho} \\ &= \int \rho_k \frac{\partial((\rho_k - \omega_2)P(\boldsymbol{\rho}, t))}{\partial \rho_k} d\boldsymbol{\rho} \\ &= \langle\omega_2\rangle - \langle\rho_k\rangle \end{aligned} \tag{10}$$

that has as solution:

$$\langle\rho_k(t)\rangle = \langle\omega_2\rangle - (\langle\omega_2\rangle - \rho_k(0))e^{-t} \tag{11}$$

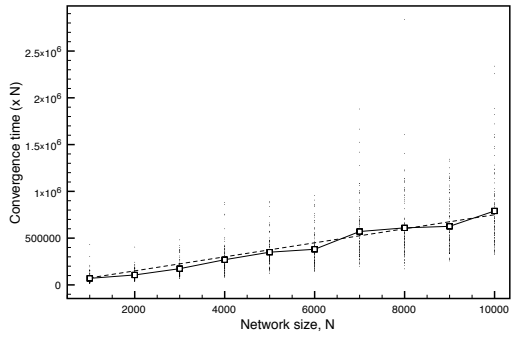


Figure 1: A simulation to show time to convergence for varying network sizes. We ran 100 independent trials for each network size, and the dots show the convergence time on each trial. The solid line shows the average convergence time. The dashed line plots $y = 75x$, which shows a very good fit with the numerically determined average convergence time.

meaning that immediately all the ρ_k reach the common value $\langle\omega_2\rangle$. These last two results (9,11) (the weighted magnetization conservation and the time behaviour of $\langle\rho_k\rangle$) are valid for a generic value of β. The consensus time $T(\rho)$, the time at which all the nodes in the system have the same state, can be easily evaluated using the adjoint of the Fokker Planck equation (6):

$$\sum_k(\omega_2-\rho_k)\frac{\partial T(\rho)}{\partial\rho_k}+\frac{1}{N}\sum_k(\omega_2+\rho_k-2\omega_2\rho_k)\frac{\partial^2T(\rho)}{\partial\rho_k^2}=-1 \tag{12}$$

Since $\rho_k \simeq \omega_2$ the first sum in (12) is null and can be eliminated. Moreover we can apply a change of variable:

$$\frac{\partial}{\partial\rho_k}=\frac{\partial\omega_2}{\partial\rho_k}\frac{\partial}{\partial\omega_2}=\frac{k^2n_k}{\mu_2}\frac{\partial}{\partial\omega_2} \tag{13}$$

and then equation (12) can be rewritten as:

$$-1=\frac{\partial^2T}{\partial\omega_2^2}(\sum_k\frac{k^4n_k}{N\mu_2^2}\omega_2(\omega_2-1)) \tag{14}$$

The equation can be easily solved in terms of ω_2:

$$T(\omega_2)=N\frac{\mu_2^2}{\mu_4}\left[(1-\omega_2)\ln(\frac{1}{1-\omega_2})+\omega_2\ln(\frac{1}{\omega_2})\right] \tag{15}$$

.

The time of consensus depends on the initial randomness in the distribution of state through ω_2, a finite term, and the size of network N explicitly and through the momenta. The size dependence is a function of both the exponent of the degree distribution and the momenta considered. The consensus time is a function of the momenta of the degree distribution and depends both on the exponent of the degree distribution ν and on the weight β. We consider $2 \leq \nu < 3$, the maximum degree being $k_{max} = N^{\frac{1}{\nu-1}}$, and for a generic m-momentum it follows that :

$$\mu_m \sim \begin{cases} N^{\frac{m}{\nu-1}-1} & m > \nu-1, \\ \log N & m = \nu-1, \\ 0(1) & m < \nu-1. \end{cases} \tag{16}$$

The exponents of the momenta appearing in (15) are 2 and 4, which are larger than $\nu - 1$. Using (16), thus,

$$T(\omega_2) \simeq \text{constant} \tag{17}$$

meaning that the time to convergence is constant in the size of the network. Note that in this analysis, one time step is taken to involve N node updates. If we count each node update as a time step, then we expect a linear relationship between the size of the network and the time to convergence.

We verify the result numerically by generating random scale-free networks (without loners) with varying N and $\nu = 2.5$, and measuring the time to convergence. This is shown in fig. 1 where we plot the time to convergence for 100 independent trials for each network size. The network size was varied from 1000 to 10000 in steps of 1000. The convergence time for each run is plotted with a dot, and the average for each network size is shown with squares joined by a solid line. We see that a linear function, as expected, provides a good fit to the data.

4. MODELING INNOVATION

The DBVM assumes that the population begins with a set of variants, and no new variants are introduced after that. This raises the question, where do the original variants come from? One possibility is that for a given linguistic feature, only a few variants are possible, and they are found very quickly, leaving no room for further innovation of that feature. In this case, it is safe to say that all variants exist "from the beginning" in the population. This is not the case in certain instances of lexical change, where new words and near-synonyms for the same concept are not limited in numbers.

The other approach, then, is to say that some form of innovation is always occurring. In this case, there would be no reason to believe that only peripherals (or some other subgroup) innovate. We assume, instead, that anyone can innovate, at any time (though the innovation rate might be low). This view is close to the position adopted by Baxter et al., with their Utterance Selection Model [2], where nobody produces exactly the same utterance every time. In their model innovation can be understood as being due to random variation in speech production, or due to noise in the communication channel. In our model, however, we are interested in discrete innovation, i.e., we are modeling change in the lexicon, which may be triggered by external circumstances, such as the need for new words with the spread of new technologies, or increased contact between different speech communities.

It has been suggested, for instance, that in times of accelerated cultural change quite a few new items with new meaning as well as new items with near-synonymous meanings to existing words can enter the lexicon. Such lexical innovations can come from two sources. The first means of lexical enrichment that can lead to lexical inflation over time is borrowing, which can arise even in situations of relatively superficial cultural contact (see Thomason and Kauffman's borrowing scale [26]). The second way is a specific type of language-internal innovation and borrowing process, called argots, jargons, and taboo [13, p. 420]. This second type seems to be the most appropriate analogy to consider with our innovation and diffusion model.

So the question we now ask is, during periods of high innovation rate, what sorts of norms will emerge in a population?

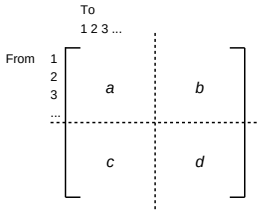


Figure 2: In the R-MAT algorithm, the adjacency matrix is recursively divided into quarters, and each quarter has a probability (a, b, c, d) associated with it. Starting with an empty matrix, we choose quarters recursively according to these probabilities until we get to a single cell, whereupon we set that cell to 1 to indicate a link.

We extend the DBVM to try to answer this question, by introducing a parameter p, the probability for innovation. In this model, a node can copy a neighbor chosen with probability proportional to the neighbors degree, as before, or, with probability p, introduce a new variant into the language.

We study the above question by keeping track of the source of each new variant, so that when we see a norm emerge in the population, we can tell which agent introduced it in the population. More precisely, we evaluate the probability that a variant that becomes a norm was introduced by a peripheral agent (or equivalently, by a non-peripheral agent).

5. THE DBVM WITH INNOVATION

In this extended model, we assume that there are v possible *initial* variants of a certain linguistic feature. To initialize the model, we assign a uniformly randomly chosen variant to each agent in the network at time $t = 0$. At each time step after that, we choose one of the agents uniformly randomly. This agent updates its variant by copying one of its neighbors with probability $(1 - p)$, where p is the innovation rate. With probability p, therefore, the agent introduces a new variant into the population. Variants are numbered starting with 1.

We keep a running count of the number of agents with each variant in the network. If one of the variants is in use by more than 90% of the population, we say that that variant has become the *norm*. Note that this means there can be periods when there is no norm in the population.

We also keep track of which agent introduced a particular variant, which will allow us to estimate the probability of variants generated by a particular class of nodes (e.g., loners) to become the norm in the network.

5.1 Generating the interaction network

Following Fagyal et al. [9], we generate the interaction network using the R-MAT algorithm [6]. R-MAT, which stands for Recursive MATrix, works by creating a set of nested communities in the network. The algorithm operates on the adjacency matrix of the network. An adjacency matrix describes a network as follows: if agent x is influenced by agent y in the social network (i.e. there is a link from x to y), then we place a 1 at row x and column y of the adjacency matrix, otherwise we place a 0 at that location.

The R-MAT algorithm uses four parameters, (a, b, c, d), which correspond to four quarters of the adjacency matrix,

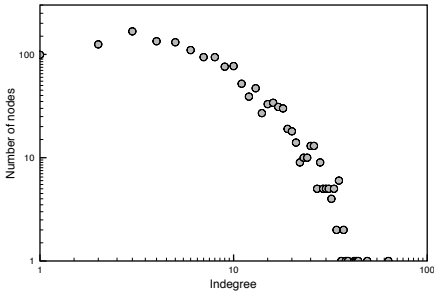


Figure 3: Indegree distribution of a network with 900 nodes and 7561 edges, generated by R-MAT.

as shown in fig. 2. We start with an adjacency matrix filled with zeroes. We then choose a quarter of the matrix with probability corresponding to its parameter. We chose the parameters $a = 0.5$, $b = 0.1$, $c = 0.1$, and $d = 0.3$. These parameters mean, for example, that half the time we choose the upper left quarter of the matrix. We then treat the chosen quarter as a new matrix, divide it into quarters, and again choose one quarter with the same set of probability parameters. This process is repeated recursively until we end up with a single cell, whereupon we set the value at that cell to 1. Again, following Fagyal et al. [9], we created a network with 900 nodes and added links to the adjacency matrix 9000 times, which resulted in 7561 unique links.

Another advantage of using the R-MAT algorithm is that it automatically results in a small number of loners ($\sim$5% of the nodes), which avoids having to artificially choose a small number of peripheral nodes to designate as loners. The generated network has a heavy-tailed power-law-like degree distribution, as shown in fig. 3, and the behavior of the DBVM on these networks is similar to its behavior on scale-free networks.

6. SIMULATIONS

A single time step of the model corresponds to a single agent updating its variant, either by copying a neighbor or by innovating. Note that if an agent chooses to copy a neighbor, its variant may not actually change, because the chosen neighbor's variant might be the same as the agent's own.

Each simulation is run for 40 million time steps. We always start with $v = 8$ initial variants. The choice of number of initial variants is arbitrary; the qualitative dynamics are the same for other (small) values of v. Once the simulation starts, agents introduce new variants in the population with innovation rate p. We count the number of individuals for each variant in the population every ten thousand time steps. If a particular variant is being used by more than 90% of the population, we say that it is the norm. We mark this on the graph by a single point for that variant number at that timestep.

Figure 4 shows norms when the innovation rate, $p = 0.0001$. We see that nearly all the time, the norm is one of the original eight variants (which are numbered from 1 to 8). Very rarely, a new variant (with number greater than 8) becomes the norm.

Figure 4 suggests that if we observe a variant as the norm in a population, it is due to a peripheral member, with high probability. The next simulation increases the innovation

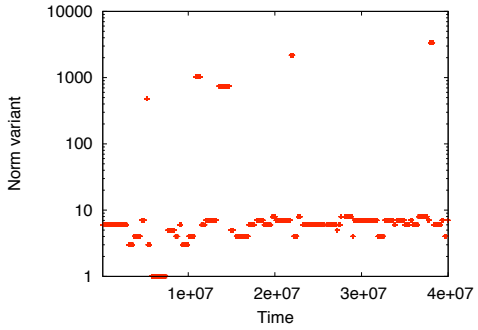


Figure 4: Norms are primarily the variants held by peripheral members, when the innovation rate is 0.0001.

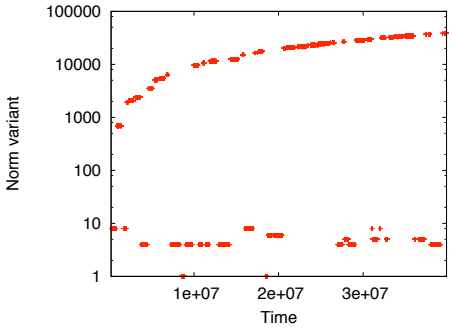


Figure 5: Norms are relatively equally divided between peripheral and non-peripheral variants when the innovation rate is 0.001.

rate by an order of magnitude, i.e., $p = 0.001$ now. The result is shown in figure 5. We see that, in this case, norms are much more evenly split between the original variants and innovative variants. This shows that probability of a non-peripheral-introduced variant becoming the norm depends on the innovation rate. In other words, if we observe a norm in a network, the statistical answer to "who introduced this into the population?" depends on the rate at which innovations are being introduced into the population as a whole. To get a more precise picture, we did a number of runs for various values of p, varying it from 0.0001 to 0.01. The results are shown in figure 6.

We did a ten runs for each value of p. Figure 6 shows two curves. The dashed line is the average fraction of the total simulation time for which a norm exists in the population. We call this the *norm time*. The norm time varies from one run to another because, even though the network is the same every time[1], the initial state of all the nodes is set randomly. The solid line shows the fraction of the norm time for which the norm was a variant introduced into the population by a loner. We call this the *loner fraction*. The error bars show one standard deviation.

Note that while the total number of variants generated over the span of the simulation is quite large, there are relatively few variants circulating in the network at any given time. The lifetime of an innovation is quite short because new variants are lost with high probability as nodes re-copy an existing variant from another node after they generate an innovation.

There are a few interesting things to note in figure 6. One is that as the innovation rate increases, the fraction of time that the norm is a peripheral-introduced variant decreases and correspondingly the fraction of time that the norm is a non-peripheral-introduced variant (1 - loner fraction, not shown in figure 6) increases. Second, at the same time, the fraction of the total time for which a norm exists at all in the population decreases with increasing innovation rate. When the innovation rate is 0.01, i.e. when an agent innovates only with a one in *hundred* probability, no norms appear in the population. This means that agents have to be *rather* conservative if norms are to exist at all. Third, we can use the fraction of time that a norm exists at all in the population as an index to determine innovation rate, and thus the probability that the norm is due to a peripheral member. This means that even if we do not know the rate at which innovations are being introduced into the population (and empirically, we can't), we can still estimate the probability that a norm is due to an innovation introduced by a peripheral.

As the innovation rate increases, the loner fraction decreases, which means that it becomes more and more likely that an innovation introduced by someone other than a loner can become the norm. The loner fraction drops below 0.5 when the innovation rate is approximately 0.002 in this simulation. At this point, it becomes more likely that an innovation introduced by a non-loner will become the norm, than that an innovation introduced by a loner will become the norm. Note that for this value of the innovation rate (and above this value), the norm time has dropped to about 25% or less. Thus, for variants introduced by non-loners to be more likely to become the norm, the innovation rate must be so high that norms only exist in the population for brief intervals.

7. LEXICAL INFLATION IN FRENCH

These findings seem to align with certain types of lexico-semantic change, such as lexical inflation, in natural languages. The following examples will focus on lexical change in French, which corresponds to one of the best known and described examples of this type of change in modern European languages. Lexical inflation is a process by which lexical items with the same meaning and similar stylistic use tend to accumulate and persist in the lexicon over time [21, p. 155], [10, p. 118]. There is general consensus among linguists that the lexicon resists the inclusion of too many perfect synonyms, i.e. lexical items duplicating the same meaning, but partial or near-synonyms can be quite numerous. While theoretical models of near-synonymy are still debated (see [25]), their practical implications have been observed for many years.

Parallel to new near-synonyms entering the language, old lexical items also need to persist for lexical inflation to occur. As Posner [21, p. 155] notes with respect to lexical inflation in French: "Most words that have outlived their time are not consigned to the dustbin, but to the attic, whence they can be taken out, dusted down, and brought back into use for special occasion." In other words, words do not neces-

[1]Since we use only one network, the values we have computed are network-specific, but the qualitative results are the same across different network instantiations.

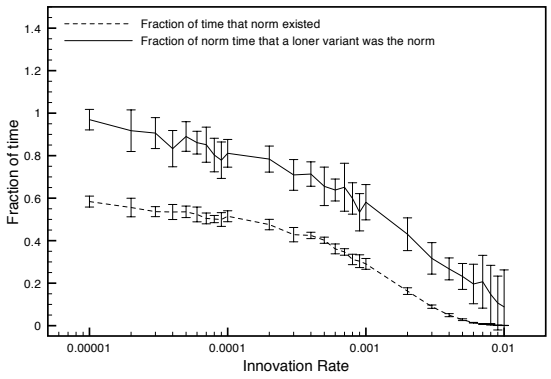


Figure 6: The dashed line shows the fraction of the total time for which a norm existed in the population, which we call "norm time", against innovation rate. We see that the norm time becomes essentially zero when the innovation rate is close to 0.01. The solid line shows the fraction of *norm* time for which the norm was a variant introduced by a loner (or one of the original variants, which are held on to only by loners after the first norm emerges). We call this the "loner fraction".

sarily disappear; just become more peripheral in the lexicon. Word-loss has indisputably characterized French, and other languages, historically. An especially large number of words were lost from French during the early modern period. Verb forms, such as *gésir* (to lie) and *quérir* (to seek) that, although not particularly frequent, had particularly irregular conjugations led to the replacement of these verbs by regular forms (*coucher* (to lie), *chercher* (to seek)). However, it has also been argued that since at least the late 18th-century the overall picture seems to be that of lexical inflation: "the number of different words (types) occurring in texts has not ceased to grow, even though for some individual words the number of instances (tokens) has regressed." [21, p. 155]. The question is: what possible mechanisms, if any, might have motivated this type of inverse relationship between type-frequency and token-frequency?

Furthermore, as near-synonyms tended to accumulate and persist in the lexicon starting from the late 18th-century, historians of French also noticed that the use of many previously peripheral and/or specialized lexical items became generalized. Casting his net of spoken lexical forms much wider than dictionaries of his time, Lazare Sainéan [22], among others, left literary usage behind to analyze an impressive array of fringe vocabulary spoken by the early 20th-century Parisian society's have-nots. He studied the jargon of solders, butchers, sailors, shoemakers, printers, and other corporations, as well as the secret terms, or argots, of thieves, beggars, prostitutes, pimps, and professional gamblers. Together with the terms of his times' entertainment industry, the Parisian cabarets, Sainéan also listed the meaning and stylistic connotations of terms handed down from child language and dialectal borrowings, or provincialisms. This wide variety of lexical items studied in their historical context lead him to one general conclusion: "Having followed the evolution of the language of criminals until the 19th-century, I came to the conclusion that the last traces of this idiom (whose sole reason to exist was its secrecy) have blended into modern-day working-class Parisian French. [...] This modern argot led to a unified idiom spoken by millions of Parisians and French people." [22, VII-VIII]. The sole reason for this "penetration of jargon into 'the vulgar' (i.e., working-class spoken French)", according to Sainéan, was the result of more frequent and "infinitely more easy" contact between different segments of French society.

Analogies between the dynamics of our computational model and the above story of lexical innovations in industrial-age Paris are suggestive. Increased innovation and large-scale spread of slang words and group-specific technical terms are first noticed in French in the modern era, i.e. starting from the late 18th-century when task-oriented labor divisions and technological advances in the manufacturing sector bring in close and regular contact members of traditionally tight-knit communities in close-reach from each other (i.e., small in diameter and showing high clustering). Marked by flagrant social inequalities (i.e., possibly of scale-free degree distribution), these networks could have been prominent sites of the type of innovation and distribution dynamics exhibited in our general model. The question whether the (inverse) relationship between increased type-frequency and decreased token-frequency is indeed governed by the same statistical dynamics as the innovation rate increase vs. norm-time decrease in our model remains to be investigated empirically in very large written and spoken language corpora. What we hope to have accomplished in this paper is a more precise formulation of the next series of hypotheses to be tested on lexical inflation in French and other languages.

8. CONCLUSION

In this paper, we have analyzed and extended a model of linguistic innovation and diffusion in social networks. We have shown how to derive the time to convergence in the degree-biased voter model. Our analysis follows the technique of Sood and Redner [24] of grouping nodes by degree to derive the Fokker-Planck equation for the system. From this we derive the adjoint equation, and the expression for convergence time follows. It turns out that time to convergence in the DBVM is simply linear in the size of the network, when time is measured as the number of updates, which we confirmed with a simulation.

The previous model is analogous to stable sedentary societies where there are a small number of variants for any linguistic variable. However, as is well-attested in historical linguistics, during periods of accelerated cultural change, languages must adapt to a greater number of innovations, especially in the lexicon. We model this situation by including a probability of innovation into the DBVM. We did simulations to qualitatively understand the nature of this extended model, and saw that as innovation rate increases, the duration of norms decreases, as is indeed the case historically. We also discovered that the probability of loner or peripheral variants becoming the norm tends to be substantially higher than non-loner variants. This has also been empirically noted, in 19^{th}-century French for example, which saw a large number of terms from argots and jargons being incorporated into the mainstream. Our approach suggests that a simple stochastic model might account for a great deal of this change.

We do not, however, claim that the above are the only reasons for linguistic change, or that simple stochastic models

can account for all the variation observed empirically. There are a number of essential sociolinguistic factors left out by our model, including effects of gender, age, and social identity. Our goal is to model these factors incrementally, in order to make sure that the effects of each new factor are fully examined before including them in the model.

We end this paper by underscoring the importance of computational modeling in sociolinguistics. Language is a very complex adaptive system. The dynamics of large-scale interactions and long-terms change are, we believe, impossible to understand fully without a rigorous mathematical theory and computational tools [17] that allow linguists to experiment with factors identified in small-scale empirical studies in sociolinguistics.

9. ACKNOWLEDGMENTS

S.S. is supported in part by NSF Nets Grant CNS- 0626964, NSF HSD Grant SES-0729441, NSF PetaApps Grant OCI-0904844, NIH MIDAS project 2U01GM070694-7, DTRA R&D Grant HDTRA1-0901-0017, DTRA CNIMS Grant HDTRA1-07-C-0113, NSF NETS CNS-0831633, DHS 4112-31805, DOE DE-SC0003957, NSF REU Suppl. CNS-0845700, US Naval Surface Warfare Center N00178-09-D-3017 DEL ORDER 13, NSF Netse CNS-1011769 and NSF SDCI OCI-1032677.

A.A. is supported by DynaNets. DynaNets acknowledges the financial support of the Future and Emerging Technologies (FET) program within the Seventh Framework Program for Research of the European Commission, under FET-Open grant number: 233847.

Zs.F.'s research on this topic is supported by the University of Illinois's Research Board for the project entitled "Modeling lexical change the role of political centers and marginal groups in the selection and spread of lexical innovations in 19th- and 20th-century France."

10. REFERENCES

[1] G. Baxter, R. Blythe, W. Croft, and A. McKane. Modeling language change: An evaluation of Trudgill's theory of the emergence of New Zealand English. *Language Variation and Change*, 21(2):257–293, 2009.

[2] G. J. Baxter, R. A. Blythe, W. Croft, and A. J. McKane. Utterance selection model of language change. *Physical Review E*, 73:046118, 2006.

[3] C. Beckner, R. Blythe, J. Bybee, M. H. Christiansen, W. Croft, N. C. Ellis, J. Holland, J. Ke, D. Larsen-Freeman, and T. Schoenemann. Language is a complex adaptive system: Position paper. *Language Learning*, 59:1–26, 2009.

[4] L. Bloomfield. *Language*. U. Chicago Press, Chicago, London, 1933.

[5] D. Britain. Exploring the importance of the outlier in sociolinguistic dialectology. In D. Britain and J. Cheshire, editors, *Social Dialectology: In Honour of Peter Trudgill*, pages 191–208. Amsterdam/Philadelphia: John Benjamins, 2003.

[6] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *SIAM conference on data mining*, 2004.

[7] P. Eckert. Adolescent social structure and the spread of linguistic change. *Lang. Soc.*, 17(2):183–207, June 1988.

[8] P. Eckert. *Linguistic Variation as Social Practice*. Blackwell, Malden, MA, 2000.

[9] Z. Fagyal, S. Swarup, A. M. Escobar, L. Gasser, and K. Lakkaraju. Centers and peripheries: Network roles in language change. *Lingua*, 120(8):2061–2079, 2010.

[10] R. Gouws. Aspects of lexical semantics. In *Solving Language Problems: From General to Applied Linguistics*, pages 98–132. Exeter University Press, Exeter, 1996.

[11] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[12] M. Granovetter. The strength of weak ties: A network theory revisited. *Sociological Theory*, 1:201–233, 1983.

[13] H. Hock. *Principles of Historical Linguistics*. Mouton de Gruyter, Berlin, New York, 2nd edition, 1999.

[14] E. Katz and P. Lazarsfeld. *Personal Influence*. Free Press, Glencoe, IL, 1955.

[15] W. Labov. The social origins of sound change. In W. Labov, editor, *Locating Language in Time and Space*, pages 251–266. Academic Press, NY, 1980.

[16] D. Lønsmann. From subculture to mainstream: The spread of English in Denmark. *Journal of Pragmatics*, 41:1139–1151, 2009.

[17] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, The Google Books Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, January 2011.

[18] J. Milroy and L. Milroy. Linguistic change, social network and speaker innovation. *Journal of Linguistics*, 21:339–384, 1985.

[19] L. Milroy. *Language Change and Social Networks*. Blackwell Publishers, Oxford, 1987.

[20] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–180, 1995.

[21] R. Posner. Lexical change. In *Linguistic Change in French*, pages 143–184. Clarendon Press, 1997.

[22] L. Sainéan. *Le Langage Parisien au XIXe siècle: facteurs sociaux, contingents linguistiques, faits sémantiques, et influences littéraires [Parisian French in the 19^{th} Century: social factors, linguistic elements, semantic facts, and literary origins]*. Champion, Paris, 1912. Available online: http://gallica.bnf.fr/ark:/12148/bpt6k5433966k.

[23] V. Sood, T. Antal, and S. Redner. Voter models on heterogeneous networks. *Phys Rev E*, 77:041121, 2008.

[24] V. Sood and S. Redner. Voter model on heterogeneous graphs. *Phys. Rev. Lett.*, 94:178701, 2005.

[25] J. R. Taylor. Near-synonyms as co-extensive categories: 'high' and 'tall' revisited. *Language Sciences*, 25(3):263–284, 2003.

[26] S. G. Thomason and T. Kaufman. *Language Contact, Creolization, and Genetic Linguistics*. University of California Press, Berkeley, 1988.

[27] U. Weinreich, W. Labov, and M. I. Herzog. Empirical foundations for a theory of language change. In W. P. Lehmann and Y. Malkiel, editors, *Directions for Historical Linguistics: A Symposium*, pages 95–188. University of Texas Press, Austin, 1968.

Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations

Laurent Navarro
Thales / UPMC – LIP6
4 place Jussieu
75005 Paris, France
laurent.navarro@lip6.fr

Fabien Flacher
Thales
20-22 rue Grange Dame Rose
78140 Vélizy, France
fabien.flacher@thalesgroup.com

Vincent Corruble
UPMC – LIP6
4 place Jussieu
75005 Paris, France
vincent.corruble@lip6.fr

ABSTRACT
Large scale agent-based simulations typically face a trade-off between the level of detail in the representation of each agent and the scalability seen as the number of agents that can be simulated with the computing resources available. In this paper, we aim at bypassing this trade-off by considering that the level of detail is itself a parameter that can be adapted automatically and dynamically during the simulation, taking into account elements such as user focus, or specific events. We introduce a framework for such a methodology, and detail its deployment within an existing simulator dedicated to the simulation of urban infrastructures. We evaluate the approach experimentally along two criteria: (1) the impact of our methodology on the resources (CPU use), and (2) an estimate of the dissimilarity between the two modes of simulation, i.e. with and without applying our methodology. Initial experiments show that a major gain in CPU time can be obtained for a very limited loss of consistency.

Categories and Subject Descriptors
D.3.3 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Multiagent systems*

General Terms
Algorithms, , Performance, Experimentation.

Keywords
Agent-based simulations – Simulation techniques – Tools and environments – Level of Detail.

1. INTRODUCTION
Agent-based simulation of credible actors in large-scale urban environments is a growing research domain, with numerous applications ranging from security to crisis management, entertainment, urban planning and virtual training. Those simulations share broadly speaking the same high-level goal: provide a powerful analytical tool which can animate a large number of individuals, with complex, credible – sometimes realistic – behavior, within a large world. Ideally, they would work in real time in a continuous space, on a standard machine and with intensive and rich interactions with one or several users.

However, simulating hundreds of thousands of individual agents within a very large environment like an airport, a crowded train station or a whole megacity, with credible behaviors, requires important computational power. This is mainly due to the complexity of the microscopic models used for instance for navigation, or decision processes that result in large states and actions spaces. Indeed, most of them require that each agent perceive its environment, update its internal variables, choose the most appropriate action and eventually communicate and learn. Reducing the complexity of the underlying algorithms is then a significant challenge.

A similar issue has already been tackled by the field of computer graphics, where Level of Detail (LOD) techniques have been investigated [1] in order to find a good balance between visual credibility and computational requirements. Those techniques tend to adapt the complexity of the 3D models based on the viewpoint of the observer. Our approach proposes a similar idea adapted to the agent models.

In this paper, we define an agent model as a computational abstraction of the behavior or the cognitive capabilities of a synthetic actor. Thus, this definition either applies to the processes and behaviors dealing with navigation, decisions, emotions, communication or social interactions. All those models take as input a representation of the agent being driven and a representation of its environment, and output an action or a modification of the internal state.

We present here a novel approach of dynamic LOD for large scale simulations, which can apply to all agent models. Moreover, instead of using predefined LOD levels, our approach is able to determine by itself the most suitable representation level for each agent, regarding the simulation context, in real time and within a continuous environment. To do so, we first introduce the generic notions of dynamic change of representation and spatial aggregation. Then, we define a concrete sub-problem and we evaluate the approach experimentally along two criteria: the impact of our methodology on the computational resources, and an estimate of the dissimilarity between a full microscopic simulation and a simulation with our methodology. Finally we discuss the results obtained and propose enhancements for future works.

2. RELATED WORK
Generating realistic behavior for virtual humans has been the subject of numerous studies in various communities. Systems like SOAR [11], ACT-R [3], ICARUS [2] or LIDA [19] are excellent examples of cognitive architectures that provide a complex modeling of extremely advanced human reasoning capabilities at microscopic scale, based on studies about human memory, problem solving and skill acquisition [4]. However, though these systems are applicable in scenes with a reasonable number of

Cite as: Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations, L. Navarro, F. Flacher and V. Corruble, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 701 - 708.

actors, they are inefficient to handle applications involving large populations of virtual humans on a standard computer. This limitation is also a disadvantage to the use of multi-agent platforms such as Cougaar [8], JADE [17] and ZEUS [26] which offer specific architectures able to distribute the virtual entities on different machines depending on the required computational load.

Attempts have been made to increase the number of simulated entities on a single computer by tuning the update time length given to each agent. To reach the amount of 200.000 vehicles simulated as individual autonomous agents with specific action selection mechanisms, SUMO [23] uses discrete calculation time steps of 1 second. Similarly, the crowd simulation proposed in [18] reduces the update times of non-visible agents and adapts their behavior to more simplified but less accurate microscopic agent models. Finally, the Process Manager described in [10] dynamically chooses between several AI update processes – full, time-sliced, postponed or replaced with simplified behavior – depending on the needs in computational resources. While those systems share the same philosophy, the first one sacrifices its real-time component for the benefit of an accurate result whereas the others elected to decrease the realism of the simulation to maintain its believability.

Some systems are able to simulate a very large number of agents using only macroscopic models. Crowd Patches [9] can handle up to 3.700 actors by dividing the world into small convex areas where agents can navigate, and using offline computed paths and animations stored within each patch to steer them. Other approaches have been attempted through the simultaneous use of macroscopic and microscopic models to define the individual behaviors of each agent. Thus, YaQ [25] uses offline predefined macroscopic paths across the world to steer up to 35.000 pedestrians using various microscopic algorithms, depending on their position: potential fields on significant areas, Craig Reynolds's seeking behavior on lower interest spots and linear steering toward their destination without collisions on unimportant regions. Similarly, Continuum Crowds [7] represents agents as particles which are subjected to three fields – one for their destination, one for their speed and one for their discomfort caused by the proximity of other agents – that guide them to their destination. Thereby, those systems combine global path planning and local collision avoidance within a single global steering model. However, they focus on navigation issues and are not easily transposed to other levels of behavior models such as ones dealing with decisions or emotions. Moreover, they do not provide the expected level of interactivity.

Some approaches also exploit the principle of simultaneous use of microscopic and macroscopic models, but choose to partition the environment and implement a model type for each zone. [22] describes a top-down approach for simulating pedestrians within a large city, which uses high level flows and distributions models to steer non-visible agents along a network of nodes that describe the accessible areas of a city, and a microscopic collision avoidance model with speed adjustment for visible actors. Similarly, the systems presented in [20] and [21] simulate vehicles navigating in a static predesigned world. The entities use a macroscopic model based on the flow theory for low interest areas without crossroads, and a microscopic multi-agent car-following model for high interest areas. Those architectures can handle several thousand agents with high consistency level and offer a good interactivity with the agents' behavior within both macroscopic and microscopic areas. But they require a preprocessed environment and predefined transition functions between the agent models.

A last approach, IVE [16], is of particular interest to our work, since it is one that introduces level of detail techniques on human decision and behavior. This framework utilizes a hierarchical reactive planning mechanism to control the agents, which uses a tree structure. Those agents are placed within a 2D world that is split into atomic cells which are hierarchically organized within a topology tree. Each level of this topology tree is linked to one of the behavioral tree, defining accessible LOD ranks. Thus, IVE can adapt the level of detail of the simulation in order to simplify the behaviors of the unobserved agents – and then reduce the computational needs – hence dealing with more than 10.000 agents simultaneously. But it requires the use of a discrete hierarchical world statically linked with the tree structure used by the decision process.

The field of multi-agent systems is not the only one to be relevant in the context of this study. Thus, Multi-Resolution Modeling (MRM), which is the joint execution of different models of the same phenomenon within the same simulation or across several heterogeneous systems, provides several relevant approaches. In selective viewing [12], only the most detailed model is executed, and all other ones are emulated by selecting information, or views, from the representation of the most detailed model. In aggregation / disaggregation techniques, one model is executed at a given time, but instead of being the most detailed one like in selective viewing, the choice of the model depends on the user needs. This approach has several variants, such as full disaggregation [15], partial disaggregation [6], playboxes [14] and pseudo-disaggregation [13]. Variable Resolution Modeling allows the construction of families of models which support dynamic changes in resolution [12] by introducing constraints during their creation, such as the standardization of all the parameters in a dictionary, the creation of a hierarchical structure for the variables or the definition of calibration rules between models.

Multiple Representation Entities [5] is a final example from the MRM field which is of particular interest here. It uses concurrent representations to ensure simulation consistency and reduce computation costs. Its approach is to maintain, at all time, all representations through all available models of a given phenomenon, using appropriate mapping functions to translate changes between two representations. The goal is to permit constant interactions between all the representations, to avoid loss of resources or time when scaling from one model to another. This approach is a powerful way to deal with complex MRM, which offers a remedy for the weakness of aggregation / disaggregation methods and requires lower resources than simultaneous execution of multiple models. But it only gives mathematical requirements for mapping functions, through the use of attributes dependency graphs. Also, it does not identify the representation at any level nor relationships between representations.

3. DYNAMIC LEVEL OF DETAIL FOR AGENT MODELS

Our approach aims to mix the philosophy of graphical level of detail with the use of multiple agent models at different resolutions. The goal is to simulate precisely the behavior of actors in areas of high level of interest with microscopic models and to simulate less precisely but more economically (resource-

wise) behavior of actors located elsewhere with macroscopic models.

Several criteria have motivated the choice of using multiple models. Firstly, it allows the capture of all the aspects of a given phenomenon. Indeed, low resolution models allow a better overall understanding, by focusing on the big picture rather than on the details, whereas high resolution models give an accurate comprehension of a specific phenomenon and tend to simulate reality. Secondly, such a choice allows the finding of a good balance between computing resources and simulation properties, such as realism, coherence and complexity. Indeed, although high resolution models are very accurate for modeling individual behaviors, they often have high computational and memory needs. On the other hand, low resolution models can save resources but tend to give less accurate results. Mixing both types of models can hopefully lead to the best of both worlds. Finally, using multi models helps design systems by mimicking the human reasoning ability – which already works at different levels of understanding – and simplifies the calibration of the models by allowing the use of available data matching at least one of the implemented models.

However, this fundamental choice leads to several challenges which can be classified along two axes. The first one relates to the models themselves. One must define the way they will be used (a model at a time, one model per areas of interest, all models simultaneously, etc…) and the way they will interact, using some of the Multi Resolution Modeling methods described above. The second axis relates to the physical agents. One must define how to manage a continuous 3D environment with complex moving agents, and how the physical position of the agents will have an impact on the model used.

3.1 Dynamic change of representation

This chapter focuses on the scalability aspect of the implemented agent models. It attempts to provide an efficient method for navigating dynamically from one model to another. The primary decision made is the choice of the aggregation / disaggregation technique to define how the models are used. This way, several agents are aggregated into a group of agents, then several groups are aggregated into a crowd, and finally several crowds are aggregated into a flow. The different agent models (agent, group, crowd and flow) are linked to each aggregation / disaggregation step.

Let M_1 be an agent model. The representation of an agent A_1 in M_1 at time t is denoted by $Rep(A_1;\ M_1; t)$ and is the vector of inner attributes of A_1 required by M_1 to operate. The number of such attributes is denoted by $|M_1|$. Then:

$$Rep(A_1;\ M_1; t) = \begin{pmatrix} a_{1;\,1}(t) \\ a_{1;\,2}(t) \\ \vdots \\ a_{1;\,|M_1|}(t) \end{pmatrix}$$

Let M_2 be another agent model. We assume that M_2 is more abstract than M_1, which also means that the representation level of M_1 is higher than the one of M_2. Finally, let $A = \{A_1;\ A_2;\ \dots;\ A_N\}$ be a set of N agents, driven by the model M_1. The goal is to find the aggregation function F_{Ag} able to transform the representation of A in M_1 at time t, into the representation of the aggregate A' controlled by the model M_2 at the same time:

$$Rep(A;\ M_1; t) = \big(Rep(A_1;\ M_1; t);\dots;\ Rep(A_N;\ M_1; t)\big) = \begin{pmatrix} a_{1;\,1}(t) & \cdots & a_{N;\,1}(t) \\ \vdots & \ddots & \vdots \\ a_{1;\,|M_1|}(t) & \cdots & a_{N;\,|M_1|}(t) \end{pmatrix}$$

$$F_{Ag}[Rep(A;\ M_1; t)] = Rep(A';\ M_2; t)$$

As is, such function is difficult to define – or to learn – because it attempts to aggregate parameters which are a priori not semantically connected, such as the velocity of the agents and their thirst level. Our approach is to split F_{Ag} into several sub functions, each operating on parameters with a similar meaning, therefore likely to share a common dynamic. In this end, we classify each agent's attributes in two categories, physical and psychological, and several subcategories, like physical traits, resources or spatial data for the first group and emotions, internal variables or knowledge for the second. Then, we partition the representation of the agents in each model. The goal is then to find the aggregation sub functions corresponding to each class of attributes, which guarantees the consistency of the models and allows a future disaggregation.

The notion of consistency is central in such an approach because it symbolizes the amount of essential information lost during the aggregation / disaggregation process and is linked to the global coherence of the simulation. A relevant definition of consistency between a high level model M and a low level model M' has been given in [12] by the comparison between the projected state of an aggregate of high level entities which have followed M, and the projected state of the same aggregate initially controlled by M'. The projection symbolizes that only a part of the final states is relevant to define the consistency. Our approach uses this notion to determine which kind of sub function fits best with which class of attributes. Thus, machine learning techniques would allow the system to find the best sub function for each attributes class between two agent models among a group of predefined operators such as SUM, MIN, MAX, MEDIAN or MEAN, by optimizing the consistency of both models.

In parallel to the definition of the aggregation sub functions, we must find the associated disaggregation operator, F_{Disag}, which aims to recreate A from A' at time t' with respect to the evolution of A' between t and t'. To do so, we define memory functions whose goal is to save data at aggregation time to facilitate the disaggregation process:

$$Mem(A;\ M_1; t) = \big(Mem(A_1;\ M_1; t);\dots;\ Mem(A_N;\ M_1; t)\big) = \begin{pmatrix} m_{1;\,1}(t) & \cdots & m_{N;\,1}(t) \\ \vdots & \ddots & \vdots \\ m_{1;\,|M_1|}(t) & \cdots & m_{N;\,|M_1|}(t) \end{pmatrix}$$

$$F_{Disag}[Rep(A';\ M_2; t');\ Mem(A;\ M_1; t)] = Rep(A;\ M_1; t')$$

There is a strong link between an aggregation function, its opposite disaggregation operator and the associated memory function. As an example, let us consider the resources of an agent. An intuitive aggregation operator would be the SUM as we may consider that a group of agents disposes of the sum of the resources of each individual. In this case, the memory function would be, for each resource attribute, a RATIO operator between the initial amount of the aggregated agent and the amount of the aggregate. Then, the disaggregation function would be a simple MULTIPLY between the new amount of the aggregate and the

memory of the agent, plus a random distribution of surplus between the agents.

Finally, such method allows our approach to tune the memory consumption by controlling the quantity of data stored by the memory functions for each aggregated agents. Thus, gradual forgetting methods can be implemented, which keeps all the data of $Mem(A;\ M_1; t)$ just after the aggregation, then creates a statistical distribution for each attribute among all the aggregated agents after a predefined period of time and finally erase all stored data if the agents have been aggregated after a long period. In this last case, random attributes are generated for the disaggregation process.

3.2 Spatial aggregation

This section focuses on the spatial aggregation of agents and addresses the issue of finding which agents should be aggregated to form a representation at a less detailed level. The philosophy employed here is to consider a group of humans as a set of individuals with similar psychological profiles and a common physical space.

To this end, two distances are defined based on the two main attributes classes defined before: a spatial distance D_Θ, and a psychological distance D_Ψ. The first one can be a trivial Euclidean distance or a more complex computation taking into account the physical path between the two agents. The second distance represents how two actors share the same thoughts (for example the same goal, the same dominant emotion or the same desire). It can be the norm between the vectors of psychological attributes or the similarity between the long term goals chosen by the agent. Those distances are combined to define the affinity between two agents A_1 and A_2.

$$Aff(A_1;\ A_2) = f[D_\Theta(A_1;\ A_2); D_\Psi(A_1;\ A_2)]$$

This affinity must be a continuous positive function, strictly decreasing as D_Θ or D_Ψ increase. It represents the connection between two agents within the simulation, only based on their individual states. Their environment is taken into account with the definition of events. Those symbolize points of particular attention which require the creation of an area of high level of interest to increase the overall consistency of the simulation. Thus, the observer's point of view, an accident or an evacuation can lead to the creation of simulation events. Let $E = \{E_1;\ E_2;\ \ldots;\ E_M\}$ be a set of M events generated by the simulation. The link between an agent and an event is characterized by a new pair of distances similar to those defined above. Although the meaning of the physical distance remains the same as the one between two agents, the signification of the psychological one is a bit different, and symbolizes how an actor is sensitive to the event. For example, if we consider an agent collapsing in the street, we can assume the impact of this event to be higher on a doctor walking nearby than on a child or an employee in a hurry. Those distances are combined to define the affinity between two agents A_1 and A_2 and an event $E_i \in E$:

$$\begin{cases} D_\Theta(A_1;\ A_2;\ E_i) = \text{Min}[D_\Theta(A_1; E_i);\ D_\Theta(A_2; E_i)] \\ D_\Psi(A_1;\ A_2;\ E_i) = \text{Min}[D_\Psi(A_1; E_i);\ D_\Psi(A_2; E_i)] \end{cases}$$

$$Aff(A_1;\ A_2;\ E_i) = f[D_\Theta(A_1;\ A_2;\ E_i); D_\Psi(A_1;\ A_2;\ E_i)]$$

Finally, we can define the link between the two agents A_1 and A_2 and E:

$$Aff(A_1;\ A_2;\ E) = \underset{i\in[1;M]}{\text{Max}}[Aff(A_1;\ A_2;\ E_i)]$$

This link is finally used to define the aggregation utility between two agents A_1 and A_2. This utility guides the choice of which agents to aggregate because they are close in their representation space and are not of interest for the simulation.

$$U_{Ag}(A_1;\ A_2) = f[Aff(A_1;\ A_2);\ Aff(A_1;\ A_2;\ E)]$$

The computation of the aggregation utilities between the agents leads to the creation of an aggregation graph, which vertices are the agents in the simulation. An edge of the graph is created when the value of the aggregation utility is greater than a given threshold. The weight of the edge is set to the value of the utility. Figure 1.A shows agents symbolized by circles with different colors representing their psychological states. The corresponding graph is shown in Figure 1.B. This structure allows optimizing the repartition of the agents within the created groups – Figure 1.C – with the use of specific graph algorithms.

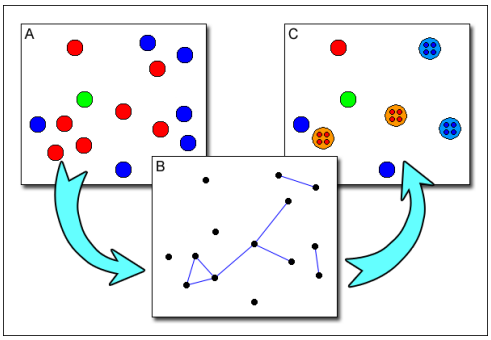


Figure 1: Example of spatial aggregation, with agents on the ground (A) used to create an aggregation graph (B) finally leading to the formation of groups (C).

The disaggregation of an aggregate A' proceeds of the same idea, although it just take into accounts the events defined in the simulation. Thus, we can define an affinity between A' and an event $E_i \in E$, then the affinity between A' and E, and finally the disaggregation utility which guides the choice of which aggregate to split because its representation grain is too coarse for the area of interest where it stands:

$$Aff(A';\ E_i) = f[D_\Theta(A';\ E_i); D_\Psi(A';\ E_i)]$$

$$Aff(A';\ E) = \underset{i\in[1;M]}{\text{Max}}[Aff(A';\ E_i)]$$

$$U_{Disag}(A') = f[Aff(A'; E)]$$

3.3 Implementation in SE-*

A large part of our approach has been implemented and evaluated within SE-*, a Thales proprietary multi-agent simulation. This system is a synthetic environment engine, designed to be highly scalable and capable of modeling complex adaptive behaviors, low-level navigation and interactions with the environment. Each agent has a motivational tree containing predefined attributes, internal variables, motivations and behaviors. A hierarchical plan is created from these different motivations and from the Smart Objects the agent may use. Currently, SE-* can animate up to 20,000 agents driven by more than 20 motivations within a complex environment.

This simulator has been used to test our approach on several scenarios. Due to the complexity of the model described above and its large number of parameters, we decided to focus on a sub problem for this first experiment, mostly by reducing the scalability of the agent models. The main simplification is the definition of two representation levels – individual and group – and the use of the same microscopic navigation and decision model for both levels. Thus, we assume that a group of a small number of agents perceive and act like a single actor. Then, we classify the attributes of the model into 3 physical categories (physical traits, resources and spatial data) and 3 psychological ones (motivations, internal variables and psychological traits). Finally, because our approach does not implement yet any automated learning mechanism for finding the aggregation operators, we defined them by hand. Thus, we use a simple MEAN operator for all the categories except for the resources which are aggregated using a SUM operator. The associated disaggregation and memory operators were also designed by hand.

To compute the affinity between two agents A_1 and A_2, we implemented a basic Euclidian distance as D_Θ and we set D_Ψ as being equal to zero if the agents have the same short-term goal, one if not. The affinity function is then defined as follow:

$$Aff(A_1;\ A_2) = \frac{1}{\alpha D_\Theta(A_1;\ A_2)^2 + \beta D_\Psi(A_1;\ A_2)^2}, (\alpha;\ \beta) \in {R_+^*}^2$$

The affinity between two agents A_1 and A_2 and an event $E_i \in E$ is defined similarly, except for D_Ψ which is always zero, symbolizing the fact that the agents are always affected by the events of the simulation. The aggregation utility between two agents A_1 and A_2 is then defined as follows:

$$U_{Ag}(A_1;\ A_2) = \frac{Aff(A_1;\ A_2)}{\gamma D_\Theta(A_1;\ A_2;E)^2}, \gamma \in R_+^*$$

Considering that D_Ψ is always zero, the definition of the disaggregation utility for an aggregate A' proceeds of the same idea:

$$U_{Disag}(A') = Aff(A';E) = \frac{1}{\delta D_\Theta(A';E)^2}, \delta \in R_+^*$$

4. EXPERIMENTAL EVALUATION

We designed 3 scenarios to evaluate our approach. Two of them take place in a subway station initially empty, including various objects such as ATMs, ticket vending machines, beverage dispensers and ticket barriers, and the last one occurs in a large city. In each scenario, the agents are driven by a dozen different motivations, such as going to work, drinking, destroying a machine, repairing a broken machine or fleeing.

Two subway stations have been designed for the two first scenarios, which share the same 3D model but have specific locations for the objects. Details are shown in Figure 2 and Figure 3. When entering the station, each agent aims to take the train and has random physical and psychological traits as well as 30% chance to own a ticket and another 30% chance to start with a small amount of money. To achieve its initial goal, and according to its inner attributes, an agent will have to get some cash at the ATM, buy a ticket, get a drink or directly go through the ticket barriers to the train doors. The first station contains 4 entries, 4 train doors, 8 ATMs (in green on the figures), 8 ticket vending machines (in yellow), 12 ticket barriers (in white), 12 exit barriers (in dark red) and 7 beverage dispensers (in red). In the second one, 4 ATMs were swapped with 4 ticket vending machines in order to see if a modification in the topology has an impact on the performances.

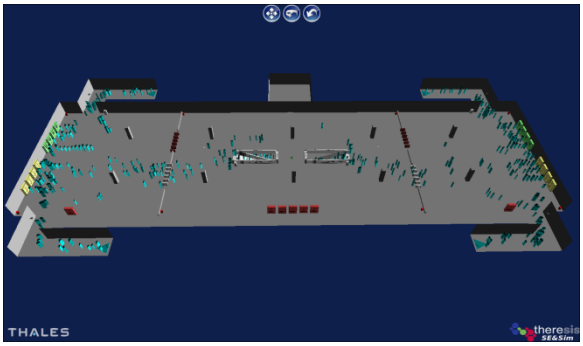


Figure 2: Top view of the first test subway station.

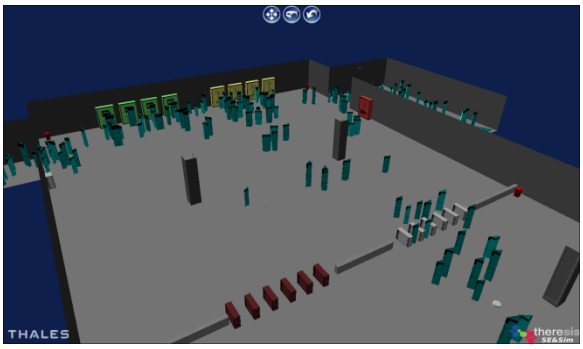


Figure 3: View of a part of the first test subway station.

The last scenario takes place in an entire city which includes the subway station, shown in Figure 4. The 3D mesh is larger and allows the simulation of thousands of agents. However, it does not contain any smart objects with which to interact. Thus, the agents only walk from entry points to exit gates without colliding, which is a typical navigation task.

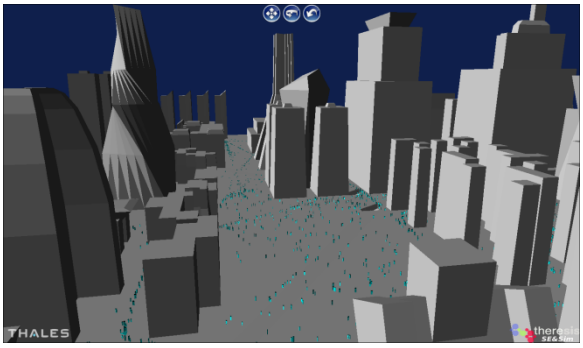


Figure 4: View of the test city.

Each scenario was run twice – one as a fully microscopic simulation without any LOD process and one with our dynamic aggregation method activated. The goal was to compare both runs to calculate both the CPU gain and the behavioral consistency. For the first criterion, we stored the total amount of time needed by the simulation to compute 60 frames within one second. For the second one, we aimed to find an estimate of the behavioral

distance between both runs. Thus, we used as objective abstract criterion: the number of uses of each object, from the start of the simulation to the measure time. Those cumulative values, taken every second, symbolize the throughput of each machine within the station. Because we already assume that the aggregation process has an impact on the simulation that is unavoidable and that may be significant, we choose to avoid using exact statistical hypothesis tests, such as Mann-Whitney's. Instead we defined, for each object, a local dissimilarity as the difference of the temporal means between the cumulative values obtained at both runs. Finally, we defined a global behavioral dissimilarity indicator as the mean of all the variations found for all objects. Let $U_o(t)$ be the cumulative number of uses of object o at time t during the microscopic simulation, and $U'_o(t)$ the cumulative uses of the same object at the same time during the simulation using our dynamic aggregation method. Then:

$$Dissimilarity = \frac{1}{N_{Objects}} \sum_{o=1}^{N_{Objects}} \left\{ \frac{\sum_{t=0}^{T} [U_o(t) - U'_o(t)]}{\sum_{t=0}^{T} U_o(t)} \right\}$$

Because it does not have any smart object to interact with, only the CPU gain was computed for the city scenario. For each scenario, we changed the maximum number of agents within the simulation and the maximum number of entities allowed inside an aggregate in order to study the impact of those parameters on the results. Finally, each experimentation has been run 5 times during 30 minutes on an Intel Core 2 Duo 2.26 GHz laptop with a memory of 2 Go. The results showed are the mean of the 5 runs.

Table 1: Experimentation results on both subway stations varying max group size and max number of entities.

Entities	Max Group Size	CPU Gain (%)		Dissimilarity (%)	
		1st Station	2nd Station	1st Station	2nd Station
100	5	43,4	43,0	3	5,4
100	10	47,5	45,9	7,1	6,9
100	15	50,3	46,5	10,9	8,3
100	20	49,4	46,9	9,9	7,6
100	25	50,5	47,6	8,7	9,7
300	5	59,9	56,7	4,9	6,3
300	10	66,7	60	4,5	5,5
300	15	67,9	65,6	7,5	8,5
300	20	67,7	66,2	5,7	6,3
300	25	69	66,9	8,6	7,1
500	5	61,5	56,8	21,5	20,1
500	10	67,4	64,5	19	19,1
500	15	69,6	67,2	18,7	18,4
500	20	70,7	66,5	17,2	17,5
500	25	72,6	69,1	14,2	16,2
1000	5	57,33	53,8	35,41	36,1
1000	10	63,97	59,4	33,68	32,4
1000	15	66,52	58,7	33,85	32,3
1000	20	67,79	60,7	31,51	31,4
1000	25	68,79	61,3	32,6	31,4

The results of the experimentation done on the first station are shown in Table 1. It appears that, for a given maximum number of agents within the station, the CPU gain is very encouraging (between 40% and 70% is saved) and logically increases with the maximum size of each aggregate. On the other hand, the behavioral dissimilarity appears to be acceptable (3-10% range for simulation inconsistency) for a maximum of 100 and 300 agents in the station. However, it becomes unsatisfactory (14 to 36% inconsistency) if the station is filled with 500 or 1000 agents. Moreover, there is no clear pattern in the dynamics of the behavioral dissimilarity as a function of the group size.

Table 1 also shows the results obtained when running the tests on the second station. The evolution of the CPU gain is the same as the one observed in the first experiment. However, the behavioral dissimilarity seems to be globally better at 300 agents even if it remains in the same range. Like before, it is difficult to detect a clear trend concerning this second criterion.

Table 2: Experimentation results for the city environment.

Entities	Max Group Size	CPU Gain (%)	Aggregation Cost (%)
10.000	5	38,0	5,3
10.000	10	48,0	7,1
10.000	15	54,2	8,7
10.000	20	56,0	9,0
10.000	25	54,5	8,8

The CPU gain observed in the city simulation is shown in Table 2. Like above, the CPU gain increases with the maximum size of each aggregate. This test demonstrates that the cost of the additional computations required by our approach (the *Aggregation Cost*) is limited and indeed remains much smaller than the total computation gain, even with a high number of agents. However, all the tests highlight a non linear variation of the CPU gain according to the group size. This can be explained by the actual number of agents within each aggregate during a simulation run. According to our observations, this number is generally between 10 and 15 agents, which coincides with the slowdown in growth of the CPU gain after a maximum of 15 agents per aggregate. The main explanation for this result lies in the choice of the psychological distance and the aggregation utility threshold. Because the first one is focused on the agents' short-term goal, it is sometimes too specific and greatly limits the size of the groups. The second one has been set high enough to trigger an aggregation if and only if both physical and psychological distances are low. Because of what has been said before, this induces the agents to be grouped only if they are also physically close enough. Finally, those fixed parameters lead to the small group sizes observed in our simulations.

The results of the two subway scenarios highlight an important variation of the behavioral dissimilarity between the experimentations involving a small number of entities – 100 and 300 – within the station, and those dealing with 500 to 1000 agents. Again, our observations showed that this difference is the direct result of the overcrowding of the station which becomes a key phenomenon when it contains more than 500 microscopic entities. In this situation, agents trying to pass the ticket barriers are colliding with the ones queuing at the ATMs and the ticket vending machines. The time required to access the objects is greatly increasing. Some agents even leave the station because they get upset to wait so long to use the machines or because they get stressed by the crowd. This situation does not appear during macroscopic simulations, because the aggregation itself greatly reduces the perceived density of agents in the station. Hence, our approach is not able to simulate properly specific microscopic

phenomena because the aggregation process is too coarse by grouping several entities into one agents and applying to it a microscopic agent model.

5. DISCUSSION AND FUTURE WORK

In this paper, we presented a novel approach of dynamic level of detail (LOD) for large scale simulations, which breaks from the general habit of using a single level of representation. Instead, we proposed the use of behavioral LOD and we introduced the notions of dynamic change of representation and spatial aggregation. Hence, our approach can be applied to various models governing agent behavior, dealing for example with navigation, decision, or emotions. Moreover, it is able to determine by itself the most suitable representation level for each agent, regarding the simulation context.

The results detailed in section 4 show an encouraging CPU gain between the microscopic simulation and the one implementing LOD techniques, even on experimentations involving a high number of agents. Moreover, this gain leads to an acceptable behavioral dissimilarity when the number of entities within the station does not lead to crowded situations.

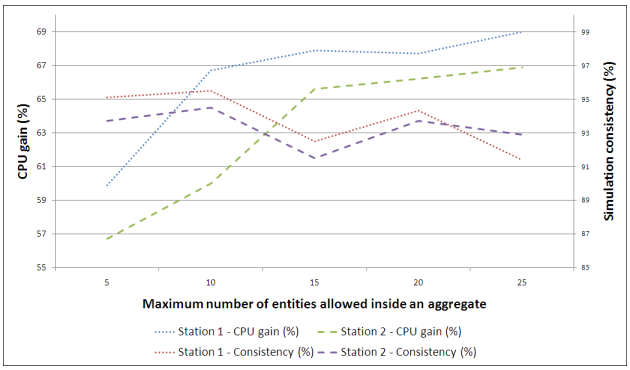


Figure 5: Evolution of CPU gain and simulation consistency for a maximum of 300 agents within each station.

However, when microscopic phenomena such as a very high density of agents are observed, the behavioral distance increases significantly. Thus, this result highlights two shortcomings of our current approach. The first one is the consequence of the assumptions made for the experimentations, where the same agent model has been used on each representation level. Doing so implies that the physical area of an aggregate to be equal to the one of an individual agent. This remark brings forward the need for a group model taking into account, at the minimum, a surface, a density and deformation factor. Of course, using a more complex model with specific group actions, knowledge and detailed internal state might help designing a more realistic simulation.

The second shortcoming of our approach results from the fact that the aggregation process, by merging several entities into a single one, may be too coarse in some situations. Although it may lead to visual inconsistencies, it can also create a strong behavioral difference between a model and another with a lower level of representation. A solution would be to use an intermediate level between several entities and an aggregate: the mesoscopic level [24]. The idea is to assume that, among the two main attributes categories defined in chapter 3.1 – physical and psychological – the first one is the most objective and observable. Thus, going from the microscopic to the mesoscopic level consists in aggregating only the psychological attributes. The mesoscopic agent will then have several bodies, corresponding to the physical microscopic bodies of the agents and driven by the low representation physical agent model (such as the navigational model), and one brain controlled by the low representation psychological model (such as decisional or emotional models). Doing so would decrease the CPU gain, because it only saves computation time on some agent models, but would also decrease the dissimilarity, in particular in crowded situations. However, many issues remain to be studied, especially the criteria for aggregating and disaggregating mesoscopic agents.

Another weakness highlighted by our experiments is the use of fixed parameters which leads to small aggregates size. This limitation could be lifted by a study on a more generic psychological distance between two agents and on the dynamicity of the most important parameters of the approach such as the aggregation and dissagregation thresholds. The first one has been arbitrarily defined and deserves to be made dependent on more subjectives parameters, such as those which are important for the user observing the simulation. For our experiments, we chose the short-term goal as criterion, but another user which may be particularly interested in the stress level of each agent might decide that two actors are psychologically close if they share the same stress level. Although this attribute cannot be used alone to define a coherent psychological distance, there is a need to give to the user some control over the weight of each psychological attribute in the computation of the distance. Secondly, the important parameters such as the thresholds were defined by hand for this first experimentation. An idea would be to set them dynamically, function of the number of representation levels, the number of agents in the simulation and the available CPU power. This way, the aggregation and desaggregation processes would adapt the context of the simulation and would provide the best CPU gain / dissimilarity ratio.

One of the most important limitations of the sub problem defined in section 3.3 is the simplification done to the scalability of the agent models part, especially for the definition of the aggregation, disaggregation and memory operators. Indeed, one of the major improvements of this work would come from the ability to obtain these operators through learning or search. As mentioned in section 3.1, the use of machine learning mechanisms can be promising. They may, for example, focus on minimizing the behavioral dissimilarity defined in section 4.

Finally, the issue of communications between agents – which relates more generally to the notion of scalability of the interactions between agents at all levels of representation – has not been directly studied in this work, as our agents do not communicate directly with each other. Considering such ability would require the definition (or the automatic search) of aggregation and disaggregation operators to transform the information emitted from an agent at a given level of representation to another at another level. If those operators are similar to the ones working on the agents' representation – except that they would work on the semantics – they would be called for each interaction and might increase the computational cost. This important point has yet to be investigated.

6. ACKNOWLEDGMENTS

We would like to thank Professor Jean-Daniel Zucker for some fruitful discussions around the notions of abstraction aggregators and meso-agents, which we intend to explore further.

7. REFERENCES

[1] D. Luebke. *A Developer's Survey of Polygonal Simplification Algorithms*. IEEE Computer Graphics and Applications. 2001.

[2] P. Langley Pat and D. Choi. *A Unified Cognitive Architecture for Physical Agents*. In proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06). 2006.

[3] J. Anderson, D. Bothell, M. Byrne, S. Douglass, C. Lebiere and Y. Qin. *An Integrated Theory of the Mind*. Psychological Review. 2004.

[4] P. Langley, J. Laird and S. Rogers. *Cognitive Architectures: Research Issues and Challenges*. Cognitive Systems Research. 2009.

[5] P. Reynolds and A. Natrajan. *Consistency Maintenance in Multiresolution Simulations*. ACM Transactions on Modeling and Computer Simulation. 1997.

[6] D. Hardy and M. Healy. *Constructive & Virtual Interoperation: A Technical Challenge*. In proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation (CGF-BR 1994). 1994.

[7] A. Treuille, S. Cooper and Z. Popovic. *Continuum crowds*. In proceedings of the 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH 2006). 2006.

[8] A. Helsinger and T. Wright. *Cougaar: A Robust Configurable Multi Agent Platform*. In proceedings of the 26th IEEE Aerospace Conference. 2005.

[9] B. Yersin, J. Maïm, J. Pettré and D. Thalmann. *Crowd Patches: Populating Large-Scale Virtual Environments for Real-Time Applications*. In proceedings of the 2009 Symposium on Interactive 3D Graphics and Games (I3D'09). 2009.

[10] I. Wright and J. Marshall. *Egocentric AI Processing for Computer Entertainment: A Real-Time Process Manager for Games*. In proceedings of the 1st International Conference on Intelligent Games and Simulation (GAME-ON 2000). 2000.

[11] J. Laird. *Extending the Soar Cognitive Architecture*. In proceedings of the 1st Conference on Artificial General Intelligence (AGI-08). 2008.

[12] P. Davis and R. Hillestad, *Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Managemen*t. In proceedings of the 25th Winter Simulation Conference (WSC'93). 1993.

[13] R. Calder, J. Peacock, B. Wise, T. Stanzione, F. Chamberlain and J. Panagos. *Implementation of a Dynamic Aggregation / Disaggregation Process in the JPSD CLCGF*. In proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representation (CGF-BR 1995). 1995.

[14] C. Karr and E. Root. *Integrating Aggregate and Vehicle Level Simulations*. In proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation (CGF-BR 1994). 1994.

[15] R. Calder, J. Peacock, J. Panagos and T. Johnson. *Integration of Constructive, Virtual, Live, and Engineering Simulations in the JPSD CLCGF*. In proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representation (CGF-BR 1995). 1995.

[16] C. Brom and Z. Vlckova. *IVE: Virtual Humans' AI Prototyping Toolkit*. International Journal of Computer and Information Science and Engineering. 2007.

[17] F. Bellifemine, G. Caire, A. Poggi and G. Rimassa. *JADE: A software framework for developing multi-agent applications*. Information and Software Technology. 2008.

[18] C. Niederberger and M. Gross. *Level-of-Detail for Cognitive real-time Characters*. The Visual Computer: International Journal of Computer Graphics. 2005.

[19] S. Franklin, U. Ramamurthy, S. D'Mello, L. MacCauley, A. Negatu, R. Silva and V. Datla. *LIDA: A Computational Model of Global Workspace Theory and Developmental Learning*. In proceedings of the AAAI Fall Symposium on AI and Consciousness: Theoretical Foundations and Current Approaches. 2007.

[20] E. Bourrel and V. Henn. *Mixing micro and macro representations of traffic flow: a first theoretical step*. In proceedings of the 9th Euro Working Group on Transportation Meeting (EWGT2002). 2002.

[21] M. El Hmam, D. Jolly, H. Abouaissa and A. Benasser. *Modélisation Hybride du Flux de Trafic*. Revue électronique Sciences et Technologies de l'Automatique. 2006.

[22] S. Stylianou, M. Fyrillas and Y. Chrysanthou. *Scalable Pedestrian Simulation for Virtual Cities*. In proceedings of the 11th ACM Symposium on Virtual Reality Software and Technology (ACM VRST 2004). 2004.

[23] D. Krajzewicz, M. Bonert and P. Wagner. *The Open Source Traffic Simulation Package SUMO*. In proceedings of the 10th RoboCup Infrastructure Simulation Competition (RoboCup 2006). 2006.

[24] A. Ruas. *The role of meso level for urban generalisation*. In proceedings of the 2nd Workshop of the ICA commission on Generalisation and Multiple Representation (GMR 1999). 1999.

[25] J. Maïm, B. Yersin, J. Pettré and D. Thalmann. *YaQ: An Architecture for Real-Time Navigation and Rendering of Varied Crowds*. IEEE Computer Graphics and Applications. 2009.

[26] H. Nwana, D. Ndumu and L. Lee. *ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems*. In proceedings of the 3rd International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'98). 1998

Logic-Based Approaches II

Reasoning about local properties in modal logic

Hans van Ditmarsch
Department of Logic
University of Seville
Spain
hvd@us.es

Wiebe van der Hoek
Department of Computer Science
University of Liverpool, UK
wiebe@csc.liv.ac.uk

Barteld Kooi
Faculty of Philosophy
University of Groningen
the Netherlands
B.P.Kooi@rug.nl

ABSTRACT

In modal logic, when adding a syntactic property to an axiomatisation, this property will semantically become true in all models, in all situations, under all circumstances. For instance, adding a property like $K_a p \rightarrow K_b p$ (agent b knows at least what agent a knows) to an axiomatisation of some epistemic logic has as an effect that such a property becomes *globally* true, i.e., it will hold in all states, at all time points (in a temporal setting), after every action (in a dynamic setting) and after any communication (in an update setting), and every agent will know that it holds, it will even be common knowledge. We propose a way to express that a property like the above only needs to hold *locally*: it may hold in the actual state, but not in all states, and not all agents may know that it holds. We can achieve this by adding relational atoms to the language that represent (implicitly) quantification over all formulas, as in $\forall p(K_a p \rightarrow K_b p)$. We show how this can be done for a rich class of modal logics and a variety of syntactic properties.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal Logic*

General Terms

Theory

Keywords

Modal logic, Correspondence theory, Canonicity, Local properties, Epistemic logic

1. INTRODUCTION

Modal logic has become *the* framework for formalising areas in computer science and artificial intelligence as diverse as distributed computing [10], reasoning about programs [11], verifying temporal properties of systems, game theoretic reasoning [18], and specifying and verifying multi-agent systems [21]. Regarding the latter example alone, since Moore's pioneering work [14] on knowledge and action, agent theories like intention logic [4] and BDI [15] use modal logic (where the modalities represent time, action, informational attitudes like knowledge or belief, or motivational attitudes like desires or intentions) to analyse interactions between modalities, like *perfect recall, no-learning, realism,* or different notions of *commitment.* As for epistemic modal logic, since the seminal work of Hintikka [12], modal epistemic logic has played a key role in knowledge representation, witnessed by its key role for reasoning about knowledge in computer science [7], and artificial intelligence [13]. The current activities in dynamic epistemic logic [1, 19] can be seen as providing a modal logical analysis in the area of belief revision, thereby providing it with a natural basis to do multi-agent belief revision, give an account of the change of higher order information, capture this all in one and the same object language: a modal language, indeed.

The popularity of modal logic in those areas is partly explained by its appealing *semantics*: the notion of state is a very powerful one when it comes to modeling computations of a machine, or describing possibilities that an agent thinks/desires/fears to be possible. Another strong feature of modal logic is its *flexibility*: the fact that temporal, dynamic, informational and motivational attitudes can be represented by modalities does not mean that they all satisfy the same laws. Rather, depending on the interpretation one has in mind, one can decide to either embrace or abandon certain principles for each of the modalities used. Syntactically, this means one assumes a number of axioms or inference rules for a modality or for the interaction of some modalities, and more often than not, this semantically corresponds to assuming some specific properties of the associated accessibility relations.

In the context of epistemic logic for instance, adding specific modal axioms allows one to specify that the knowing agent is *veridical* ($K_a p \rightarrow p$): if agent a knows that p, then p must be true), or that he is positively ($K_a p \rightarrow K_a K_a p$) or negatively ($\neg K_a p \rightarrow K_a \neg K_a p$) *introspective*. Those axioms happen to correspond (in a precise way: correspondence theory for modal logic is already some decades old, cf. [17]) to reflexivity, transitivity and euclidicity of the associated accessibility relation R_a, respectively. Moreover, the axioms are *canonical* for it: adding the syntactic axiom to a modal logic enforces the canonical model for the logic to have the corresponding property, which then in turn implies that completeness of the logic with respect to the class of models satisfying that relational property is guaranteed. At this point, is important to note the difference between $K_a p \rightarrow p$ as a *formula* and that as a *scheme*, or *axiom*: as a formula, it merely expresses that regarding the atom

Cite as: Reasoning about local properties in modal logic, van Ditmarsch, van der Hoek, and Kooi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 711-718.

p, agent a cannot know it without it being true. However, when we assume it as an axiom, or as a scheme, it means that we assume it to hold for every substitution instance of p, in other words, we assume that for all formulas φ, the implication $K_a\varphi \to \varphi$ holds.

It is often argued (indeed, already by Hintikka in [12]) that a distinguishing feature between knowledge and belief is that whereas knowledge is veridical, belief need not be, i.e., the scheme $B_ap \to p$ should not be assumed as an axiom for belief. This then simply entails that epistemic logics have veridicality as an axiom, and doxastic logics have not. Semantically speaking: the accessibility relations denoting knowledge are reflexive, those denoting belief need not be. But how then to deal with a situation where we want to express that "currently, a's beliefs happen to be true"? If we add $B_ap \to p$ as an axiom to our logic, the effect is that in all models (with respect to which the logic is complete), and in all states, all instances of that axiom are true, i.e., for all models M, for all states s *and for all formulas* φ, we then have $M, s \models B_a\varphi \to \varphi$. Given a model M and a state s we *can* express that a's belief that an individual proposition q holds is correct: $M, s \models B_aq \wedge q$. And we can express that a's belief about q is correct: $M, s \models (B_aq \to q) \wedge (B_a\neg q \to \neg q)$. But what we *cannot* express in modal logic is that $B_a\varphi \to \varphi$ holds for all φ in one state, without claiming at the same time it should hold throughout the model. As a consequence, we cannot express in the object language that agent b thinks that agent a's beliefs are correct, while agent c believes that a is wrong about a proposition q. The closest one gets to expressing that would be to say that for all φ, in M, s we have $M, s \models B_b(B_a\varphi \to \varphi) \wedge B_c((B_aq \wedge \neg q) \vee (B_a\neg q \wedge q))$ (but here, the quantification over φ is on a meta-level, and not in the scope of agent b). Neither can we say, in a temporal doxastic context, that a's beliefs now are correct, but tomorrow they need not be.

To give another example of the same phenomenon, suppose one adds the scheme $K_ap \to K_bp$ to a modal logic (b knows everything that a knows). Semantically, this means $R_b \subseteq R_a$. If the logic is about a set of agents A, then it becomes common knowledge among A that b knows at least what a knows! And if there is a notion of time, we have that it will always be the case that b knows at least what a knows, and, when having modalities for actions, it follows that no action can make it come about that a has a secret for b, in particular, it is impossible to inform a about something that b does not already know—this rules out dynamics that is, in contrast, very possible in DEL.

So, the general picture in modal logic that we take as our starting point is the following. One has a modal logic to which one adds an axiom scheme θ (say, $B_ap \to p$). If one is lucky, the scheme corresponds to a relational property $\Theta(x)$ (in the case above, Rxx). However, adding θ to the logic means having $\Theta(x)$ true everywhere, implying that θ is always true. What we are after is looking at ways to enforce the scheme θ *locally*. To do so, we will add a marker $\boxdot$ to the modal language, such that $\boxdot$ is true locally, in a state s, if and only if Θ is true, locally (i.e., Rss holds).

Doing so, we generalise work of [20], where a case study, in the context of a multi-agent logic **S5**, is given for 'knowing at least as much as', i.e., in our terminology, $\boxdot(a, b)$ in [20] equals $a \succeq b$, and our $\Theta(a, b)(x)$ is the property $\forall y(R_bxy \Rightarrow R_axy)$ in [20].

We will not only generalise the result of [20] to arbitrary modal logics $\mathbf{K}(+\varphi_1, \ldots, +\varphi_n)$ where φ_i are canonical axioms, but also we allow to add several markers at the same time. This then enables that we cannot only make global properties locally true, but it allows for far more subtle quantifications over formulas than is allowed in modal logic, enabling us to express properties like "If all of John's beliefs are correct, than so must Mary's beliefs be", or "If John knows now everything that Mary knows, then that must have been true yesterday as well" or "If John's beliefs are correct, then he must know that Mary's beliefs are correct as well" (for more examples of such quantification, see Section 2.1).

This paper is organised as follows. In Section 1.1 we sketch how our machinery will look like. Then, in Section 2 we formally introduce three languages and present an example. Section 3 provides an axiomatisation of our extended modal logic, we come back to the example and make a case for completeness. Finally, in Section 4 we summarise and conclude.

1.1 To a Modal Logic with Local Schemes

In this section we introduce three languages to reason about Kripke models. The place where these languages meet are important for our set-up. Let us outline the overall approach at the hand of an example: formal definitions follow later in this section. First of all, we are interested in a *modal scheme* $\theta(a, b, p) = [a]p \to [b]p$ in a modal language $\mathcal{L}$ (generally, we write $[a]\varphi$ for modal formulas, but for epistemic interpretations we may write $K_a\varphi$, and for doxastic ones $B_a\varphi$). To the modal language we add a relational atom $\boxdot(a, b)$, or, in this specific case $Sup(a, b)$, which will be true in a state s iff $\forall y(R_bsy \Rightarrow R_asy)$ holds. The latter property is a formula $\Theta(a, b)(s)$ in a first-order language $\mathcal{L}^1$. Our modal logic should now formalise the idea that $\theta(a, b, c)$ and $\boxdot(a, b)$ 'capture the same'. Rather than saying that the two are equivalent, the logic will take care that something along the following lines holds: consistency of a formula φ with an occurrence of $\neg\boxdot(a, b)$ is the same as consistency of φ with the occurrence of $\neg\boxdot(a, b)$ replaced by $\neg\theta(a, b, p)$ (if p is a fresh atom). For completeness of the logic, we then take care that in its canonical model, the truth of $\theta(a, b, p)$ in a specific world (i.e., maximal consistent set Δ) coincides with property $\Theta(\Delta)$. We show that our construction works because the second order formula $\forall P(\forall x(R_axy \Rightarrow Py) \Rightarrow \forall y(R_bxy \Rightarrow Py)) = \forall P\hat{\Theta}(a, b, P)(x)$ is equivalent to $\Theta(x)$. The formula $\forall P\hat{\Theta}(a, b, P)$ is an example of a formula from the third language that we use, i.e., a second-order language $\mathcal{L}^2$.

The languages that we define are simple extensions of languages usually studied in standard modal logic [3, 2]. More specifically, our modal logic extends that of modal logic with some relational atoms $\boxdot$, the first order language is the standard language to reason about properties of accessibility relations, and the second order language is similar to the one usually obtained by applying the so-called standard translation to modal formulas. Our completeness proof, in turn, is an extension of 'standard' completeness proofs in modal logic: we sometimes have to add fresh atoms p to ensure that $\theta(a, b, p)$ is satisfied. However, we have borrowed ideas from [5] to prove our Extension Lemma 2 and ideas from [16] to make this lemma work 'everywhere in the canonical model'. Space does not allow to include the proofs themselves, but we will make an effort to explain the overall idea and the construction of the canonical model.

2. LANGUAGE AND SEMANTICS

As outlined above, we deal with three languages, which are all interpreted over the same objects, i.e., Kripke models. The languages are an extended modal language $\mathcal{L}$, a first order language $\mathcal{L}^1$ and a second order language $\mathcal{L}^2$.

For all languages, we assume a set of modality labels $A = \{a_1, \ldots, a_{|A|}\}$. In the modal language, these will give rise to modalities $[a]$, and in the other two languages, we assume to have a binary relation R_a for each $a \in A$. For the latter two languages we also assume to have a set of variables $\mathcal{X} = \{x, y, \ldots\}$. The variables will range over possible worlds: note that neither in $\mathcal{L}^1$ nor in $\mathcal{L}^2$ we assume to have constants. For $\mathcal{L}^2$, we furthermore use a set $\Pi = \{P, P_1, P_2, \ldots, Q, Q_1, Q_2, \ldots\}$ of unary predicates. For each such predicate P in Π we assume to have an atomic proposition $p \in \pi$ that are building blocks for the modal language $\mathcal{L}$. On top of this, for this modal language $\mathcal{L}$ we assume a finite set $\rho = \{\boxdot_1, \boxdot_2, \cdots \boxdot_m\}$ of relational atoms: they are nothing else than syntactic atoms of which the truth depends on local properties of accessibility relations (see the function I in Definition 1). Therefore, we will often write $\boxdot(a_1, \ldots, a_n)$ rather than $\boxdot$ to make this dependence clear, and treat $\boxdot$ as if it were an n-ary relational *predicate* (rather than an *atomic* symbol). Our languages will be denoted $\mathcal{L}(A, \pi, \rho)$ (the modal language), $\mathcal{L}^1(A, \mathcal{X})$ (the first order language) and $\mathcal{L}^2(A, \Pi, \mathcal{X})$ (the second order language). If the parameters for the languages are clear, we will also write $\mathcal{L}$, $\mathcal{L}^1$ and $\mathcal{L}^2$, respectively.

DEFINITION 1 (MODAL LANGUAGE). *Let the sets A, π, and ρ be as described above. The modal language $\mathcal{L}(A, \pi, \rho)$ is defined as follows:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [a]\varphi \mid \boxdot(a_1, \cdots, a_n)$$

where $a, a_1, \ldots, a_n \in A$, $p \in \pi$ and $\boxdot$ is an n-ary relational atom in ρ. Formula $\langle a\rangle\varphi$ is shorthand for $\neg[a]\neg\varphi$ and we also assume the usual definitions for disjunction, implication and bi-implication. If the modality is an epistemic one, the labels are agents, and we write $K_a\varphi$ rather than $[a]\varphi$. For a doxastic interpretation we write $B_a\varphi$, etc.

A formula without occurrences of relational atoms is called a purely modal formula. Suppose we have a multi-modal formula $\theta(a_1, \ldots, a_n, p_1, \ldots p_k)$ where $a_1, \ldots, a_n$ are labels of modalities $[a_1], \ldots [a_n]$ and $p_1, \ldots, p_k$ are atoms. We will write $\vec{a}$ for the tuple $a_1, \ldots, a_n$ and $\vec{p}$ for $p_1, \ldots, p_k$. When we write $a \in \vec{a}$ we mean that a is one of the labels occurring in the tuple $\vec{a}$, likewise for p and $\vec{p}$. Finally, for any tuple $\vec{x} = x_1, \ldots, x_n$ with each x_i taken from some set X, we will write $\vec{x} \in \vec{X}$.

DEFINITION 2 (FIRST AND SECOND ORDER LANGUAGE). *Let A and $\mathcal{X}$ be given. First define a language $\mathcal{L}^+(A, \mathcal{X})$:*

$$\Theta := R_a xy \mid \forall y\Theta \mid \neg\Theta \mid \Theta \;\&\; \Theta$$

with $a \in A$, and $x, y \in \mathcal{X}$. Now, our first order language $\mathcal{L}^1(A, \mathcal{X})$ is the one-free-variable sublanguage of $\mathcal{L}^+$, i.e., the sublanguage of $\mathcal{L}^+$ consisting of all formulas with at most one variable not in the scope of a quantifier. If $\Theta \in \mathcal{L}^1(A, \mathcal{X})$ has x as its only free variable, and if $a_1, \ldots a_n$ are all the modality labels occurring in Θ, we will also write $\Theta(\vec{a})(x)$ for Θ.

Finally, given A, Π and $\mathcal{X}$ we define the second order language $\mathcal{L}^2(A, \Pi, \mathcal{X})$ as the one-free-variable fragment of

$$\hat{\Theta} := P(x) \mid R_a xy \mid \forall y\hat{\Theta} \mid \forall P\hat{\Theta} \mid \neg\hat{\Theta} \mid \hat{\Theta} \;\&\; \hat{\Theta}$$

with $P \in \Pi, x, y \in \mathcal{X}$ and $a \in A$. In $\mathcal{L}^1(A, \mathcal{X})$ and $\mathcal{L}^2(A, \Pi, \mathcal{X})$, existential quantification (using $\exists$) and implication (using $\Rightarrow$) are defined in a standard way.

We write Px for $P(x)$. As mentioned earlier, all languages will be interpreted over Kripke models.

DEFINITION 3 (KRIPKE MODELS AND FRAMES). *Given A, π and ρ, a Kripke model is a tuple $M = \langle W, R, I, V\rangle$ where*

- *W is a set of possible worlds;*
- *$R : A \to \wp(W \times W)$ assigns a binary relation to each modality label*
- *$I : \rho \to \mathcal{L}^1(A, \mathcal{X})$ assigns a first order property to each relational atom in ρ*
- *$V : \pi \to \wp(W)$ assigns a set of possible worlds to each propositional variable*

Rather then $(w, v) \in R(a)$ we will write $R_a wv$. A Kripke frame is a tuple $F = \langle W, R, I\rangle$ such that $\langle M, V\rangle = \langle W, R, I, V\rangle$ is a model. The 'arity' of a symbol $\boxdot \in \rho$ can be read off from its interpretation $I(\boxdot)$: if $I(\boxdot)$ refers to modalities $a_1, \ldots, a_n$, then we may write $\boxdot(\vec{a})$ for $\boxdot$.

DEFINITION 4 (SEMANTICS OF MODAL FORMULAS). *Let A and π be given. Also, let $M = \langle W, R, I, V\rangle$. Then we define, for $\varphi \in \mathcal{L}(A, \pi, \rho)$:*

$M, w \models p$	*iff*	$w \in V(p)$
$M, w \models \neg\varphi$	*iff*	$M, w \not\models \varphi$
$M, w \models \varphi \wedge \psi$	*iff*	$M, w \models \varphi$ *and* $M, w \models \psi$
$M, w \models [a]\varphi$	*iff*	*for all v if $R_a wv$, then* $M, v \models \varphi$
$M, w \models \boxdot(\vec{a})$	*iff*	$I(\boxdot(\vec{a}))(w)$ *holds*

The class of all models is denoted $\mathcal{K}(A, \pi, \rho)$. All models with interpretation I are denoted $\mathcal{K}(A, \pi, \rho, I)$. Validity in a model M is defined as usual. Moreover, $\mathcal{K}(A, \pi, \rho) \models \varphi$ means that for all I, $M = \langle W, R, I, V\rangle$, and all $w \in W$, we have $M \models \varphi$. Given I, we say that φ is I-satisfiable, if there is a model $M = \langle W, R, I, V\rangle$ and a $w \in W$ such that $M, w \models \varphi$. Formula φ is I-valid if $\neg\varphi$ is not I-satisfiable. If $F = \langle W, R, I\rangle$ is a frame, $F, w \models \varphi$ is defined as: for all valuations V, $\langle W, R, I, V\rangle, w \models \varphi$.

Interpretation of $\mathcal{L}^1(A, \mathcal{X})$-formulas in a model $M = \langle W, R, I, V\rangle$ is straightforward. For $\mathcal{L}^2(A, \Pi, \mathcal{X})$, we assume that Ps holds for a predicate P iff $s \in V(p)$. In other words, the link between a propositional atom and a unary predicate is implicit by using lower-case and upper-case notation.

EXAMPLE 1. *Let $\boxdot(a, b)$ be such that in M with interpretation I, we have $I(\boxdot(a, b)) = \Theta(a, b)$ where $\Theta(a, b)(x) = \forall y(R_b xy \Rightarrow R_a xy)$, saying that in the current world w, the set of a-successors of w is a superset of the set of b-successors of w. If this is the interpretation of $\boxdot(a, b)$, we will also write $Sup(a, b)$. As a second example, take $\boxdot = \boxdot(a)$ to be such that $I(\boxdot(a))(x) = R_a xx$. Note that $B_a \boxdot(a)$ can hence be interpreted as 'a believes that his beliefs are correct', since $M, w \models B_a \boxdot(a)$ does entail that for all φ, $M, w \models$*

Table 1: In this table, $\vec{a}$ is a sequence a or (a,b) or (a,b,c) of modality labels, and $\vec{p}$ is either the single atom p or the sequence p,q. $\Theta(\vec{a})(x)$ is a property of a state x, and $\boxdot(\vec{a})$ is a name in the object language such that $\boxdot(\vec{a})$ holds at w iff $\Theta(\vec{a})(w)$ holds of M.

$\theta(\vec{a},\vec{p})$	$\Theta(\vec{a})(x)$	$\boxdot(\vec{a})$
$[a]p \to [b]p$	$\forall y(R_b xy \Rightarrow R_a xy)$	$Sup(a,b)$
$[c]p \to [a][b]p$	$\forall y,z((R_a xy \& R_b yz) \Rightarrow R_c xz)$	$Trans(a,b,c)$
$\neg[a]\bot$	$\exists y R_a xy$	$Ser(a)$
$[a]p \to p$	$R_a xx$	$Refl(a)$
$\neg[a]p \to [b]\langle c\rangle p$	$\forall yz((R_a xy \& R_b xz) \Rightarrow R_c yz)$	$Eucl(a,b,c)$
$\langle a\rangle p \to \langle b\rangle\langle c\rangle p$	$\forall z(R_a xz \Rightarrow \exists y R_b xy \& R_c yz)$	$Dens(a,b,c)$

Table 2: For every modal formula $\theta(\vec{a},\vec{p})$ from Table 1, we give the second order translation $\hat{\Theta}(\vec{a},\vec{P})$.

$\theta(\vec{a},\vec{p})$	$\hat{\Theta}(\vec{a},\vec{P})(x)$
$[a]p \to [b]p$	$\forall y(R_b xy \Rightarrow Py) \Rightarrow \forall z(R_a xz \Rightarrow Pz)$
$[c]p \to [a][b]p$	$\forall w(R_c xw \Rightarrow Pw) \Rightarrow \forall y(R_a xy \Rightarrow \forall z(R_b yz \Rightarrow Pz))$
$\neg[a]\bot$	$\neg\forall y(R_a xy \Rightarrow \bot)$
$[a]p \to p$	$\forall y(R_a xy \Rightarrow Py) \Rightarrow Px$
$\neg[a]p \to [b]\langle c\rangle p$	$\neg\forall w(R_a xw \Rightarrow Pw) \Rightarrow \forall y(R_b xy \Rightarrow \exists z(R_c yz \& Pz))$
$\langle a\rangle p \to \langle b\rangle\langle c\rangle p$	$\exists w(R_a xw \& Pw) \Rightarrow \exists y(R_b xy \& \exists z(R_c yz \& Pz))$

$B_a(B_a\varphi \to \varphi)$ (but see Remark 1). As a final example, take $\boxdot(a,b,c)$ with $I(\boxdot(a,b,c))(x) = \forall y\forall z((R_a xy \ \& R_b yz) \Rightarrow R_c xz)$ we will write $Trans(a,b,c)$ for $\Theta(a,b,c)$. Of course, a special case of this is $\boxdot = \boxdot(a,a,a)$ saying that currently, at world w, the relation R_a is transitive. For more examples, see Table 1.

Remark 1. *Take $\boxdot(a)$ and M such that $I(\boxdot(a)) = \forall x R_a xx$. Note that although $M,w \models \boxdot(a)$ entails that agent a's beliefs are correct, the converse is not true, as the following example shows. Let $M = \langle W,R,I,V\rangle$ be such that $W = \{w,v\}$, and $R_a = \{(w,u),(u,w)\}$. Moreover, assume that for all p, $w \in V(p)$ iff $u \in V(p)$. Since M,w and M,v are bisimilar [2, Chapters 1 and 5] models, we have $M,w \models \varphi$ iff $M,u \models \varphi$, and hence $M,w \models B_a\varphi \to \varphi$, for all purely modal φ. However, since $(w,w) \notin R_a$, we have $M,w \models \neg\boxdot(a)$.*

Note that, since $\Theta(\vec{a})(w)$ does not refer to atomic propositions p (or, rather predicates P), we have that $\Theta(\vec{a})(w)$ holds in the model M iff $\Theta(\vec{a})(w)$ holds in the frame F.

Definition 5 (Standard Translation). *Fix sets A, π, Π, and ρ. Fix an interpretation I and write $I(\boxdot(\vec{a})) = \Theta_{\boxdot}(\vec{a})$. We define $\mathsf{ST}_I : \mathcal{L}(A,\pi,\rho) \times \mathcal{X} \to \mathcal{L}^2(A,\Pi,\mathcal{X})$ by*

$$\begin{array}{lcl} \mathsf{ST}_I(p)(x) & = & P(x) \\ \mathsf{ST}_I(\boxdot\vec{a})(x) & = & \Theta_{\boxdot}(\vec{a})(x) \\ \mathsf{ST}_I(\neg\varphi)(x) & = & \neg\mathsf{ST}_I(\varphi)(x) \\ \mathsf{ST}_I(\varphi\wedge\psi)(x) & = & \mathsf{ST}_I(\varphi)(x) \ \& \ \mathsf{ST}_I(\psi)(x) \\ \mathsf{ST}_I([a]\varphi)(x) & = & \forall y(Rxy \Rightarrow \mathsf{ST}_I(\varphi)(y)) \end{array}$$

In the last clause, y is assumed to be a fresh variable. If φ is purely modal (i.e., φ is $\boxdot$-free), $\mathsf{ST}_I(\varphi)$ does not depend on the interpretation I and we write $\mathsf{ST}(\varphi)$ in such a case.

Note that the standard translation $\mathsf{ST}(\theta(\vec{a},\vec{p}))$ of a modal formula involving modalities $\vec{a}$ and atoms $\vec{p}$ is typically a formula $\hat{\Theta}(\vec{a},\vec{P})(x)$ involving binary relations R_a (one for each $a \in \vec{a}$) and predicates P (one for each $p \in \vec{p}$) and a free variable x.

Example 2. *Take $\theta(a,b,p) = [a]p \to [b]p$. Then we have that $\mathsf{ST}_I(\theta(a,b,p))(x) =$*

$$\forall y(R_b xy \Rightarrow Py) \Rightarrow \forall z(R_a xz \Rightarrow Pz)$$

If $\theta(a,b,c,p) = [c]p \to [a][b]p$, we have $\mathsf{ST}_I(\theta(a,b,c,p)) =$

$$\forall w(R_c xw \Rightarrow Pw) \Rightarrow \forall y(R_a xy \Rightarrow \forall z(R_b yz \Rightarrow Pz))$$

The following is straightforward from classical modal theory, except for the case of $\theta(\vec{a},\vec{p}) = \boxdot(\vec{a})$, in which case it follows directly from the truth definition (Definition 4) for $\boxdot$-formulas.

Lemma 1. *Let I be an interpretation of relational symbols and let $\theta(\vec{a},\vec{p})$ be a modal formula. Let $\hat{\Theta}(\vec{a},\vec{P})(x)$ be its second order translation $\mathsf{ST}_I(\theta(\vec{a},\vec{p}))$. Then, for all models $M = \langle W,R,I,V\rangle$, all frames $F = \langle W,R,I\rangle$ and worlds $w \in W$ we have*

1. *$M,w \models \theta(\vec{a},\vec{p})$ iff $\hat{\Theta}(\vec{a},\vec{P})(w)$ holds in M*
2. *$M \models \theta(\vec{a},\vec{p})$ iff $\forall x\hat{\Theta}(\vec{a},\vec{P})(x)$ holds in M*
3. *$F,w \models \theta(\vec{a},\vec{p})$ iff $\forall\vec{P}\hat{\Theta}(\vec{a},\vec{P})(w)$ holds in F*
4. *$F \models \theta(\vec{a},\vec{p})$ iff $\forall x\forall\vec{P}\hat{\Theta}(\vec{a},\vec{P})(x)$ holds in F*

Table 2 provides the standard translation $\hat{\Theta}(\vec{a},\vec{P})(x)$ of the modal formulas $\theta(\vec{a},\vec{p})$ that we introduced in Table 1.

Definition 6. *Let $\theta(\vec{a},\vec{p})$ be a purely modal formula from $\mathcal{L}(A,\pi,\rho)$ and suppose that $\Theta \in \mathcal{L}^1(A,\mathcal{X})$ is such that $\Theta(\vec{a})(x)$ is equivalent with the second order formula $\forall\vec{P}\hat{\Theta}(\vec{a},\vec{P})(x)$ where $\hat{\Theta}(\vec{a},\vec{P})(x) = \mathsf{ST}(\theta(\vec{a},\vec{p}))(x)$. Then we say that $\theta(\vec{a},\vec{p})$* characterises *$\Theta(\vec{a})(x)$.*

If $\theta(\vec{a},\vec{p})$ characterises $\Theta(\vec{a})(x)$ then we have that $F, w \models \theta(\vec{a},\vec{p})$ if $\Theta(\vec{a})(w)$ holds. In other words, $\theta(\vec{a},\vec{p})$ corresponds with $\Theta(\vec{a})$. There are many well known classes of modal formulas $\theta(\vec{a},\vec{p})(x)$ for which it is guaranteed that the second order formula $\forall\vec{P}\mathsf{ST}(\theta(\vec{a},\vec{p}))$ is equivalent to a formula $\Theta(\vec{a})(x) \in \mathcal{L}^1(A,\mathcal{X})$. A large set of formulas for which this is true is the set of so-called *Sahlqvist formulas*. Moreover, given such a Sahlqvist formula $\theta(\vec{a},\vec{p})$, its first order equivalent $\Theta(\vec{a})(x)$ can be effectively computed from it [3, Theorem 3.54]. So for Sahlqvist formulas, we can effectively find the first-order formula that it characterises. All the formulas $\theta(\vec{a},\vec{p})$ from Table 1 are (equivalent to) Sahlqvist formulas.

Take the specific example in a doxastic context where $\Theta(a)(x)$ is $\forall x R_a x x$, and $I(\boxdot(a)) = \Theta(a)$, note that $\theta(a,p) = (B_a p \to p)$ characterises $\Theta(a)(x)$ but still, as shown in Remark 1, the formulas $\boxdot(a)$ and $\theta(a,p)$ are not *equivalent*. Still, the two should be strongly connected, in a sense we will explain in Section 3. We first look at an example, involving our extended modal language.

2.1 A Simple Example

Consider five friends, Joey, Chandler, Ross, Monica and Phoebe (or j, c, r, m and p, for short). In this example, we use 'think' and 'believe' for the same thing. Joey believes that Monica's beliefs are at least as accurate as Ross' beliefs, i.e., Joey believes that if Ross' beliefs are correct, so must Monica's be (A). Joey also believes that Monica thinks that Chandler believes anything that Monica believes (B). Although Joey does not think that he believes everything he knows (he thinks that he knows he cannot find a job as an actor, but at the same time cannot believe it), he actually believes anything he knows (C). Moreover, Joey thinks that Chandler's beliefs are consistent (D). Finally, Joey happens to know that Monica believes that Phoebe is in competition with her for Chandler's attention, but at the same time Joey thinks that Chandler believes that Phoebe is not in competition with Monica for his attention (E). Then, we conclude that Joey believes that Ross' beliefs are not guaranteed to be correct (F), or, better, that Joey believes he may assume that some formula is believed by Ross, but not true (F').

We first give a (semi-formal) formalisation of our assumption using a modal logic that allows for quantification over formulas. Let z represent the proposition that Joey cannot find himself a job as an actor, and let q be the proposition that Phoebe is in competition with Monica for Chandler's attention. This formalisation is given in Table 3, where assumption A in the episode is represented as a, etc. The formalisation in our language $\mathcal{L}(A,\pi,\rho)$ follows in Table 4.

We can now be more precise about what it means that our language can do more than just formalising a local version of a global property. For instance, the global property $B_a p \to p$ will have a local counterpart $\mathit{Refl}(a)$. Locally, this will denote something that is similar to $\forall\varphi(B\varphi \to \varphi)$. But if one looks at the 'translation' a in Table 3 of the assumption A above, i.e., $B_j(\forall\varphi(B_r\varphi \to \varphi) \to \forall\varphi(B_m\varphi \to \varphi))$, it becomes clear that this is different from the quantification $g : \forall\varphi B_j((B_r\varphi \to \varphi) \to (B_m\varphi \to \varphi))$, which one would get as a local counterpart of an axiom $B_j((B_r p \to p) \to (B_m p \to p))$. That a and g are not equivalent, can be seen in the model M, w of Figure 1, where a is true in M, w, but g is not: for the latter, $\varphi = p$ provides a counterexample. That a is true in M, w is easily seen from realising that a is formalised by a'.

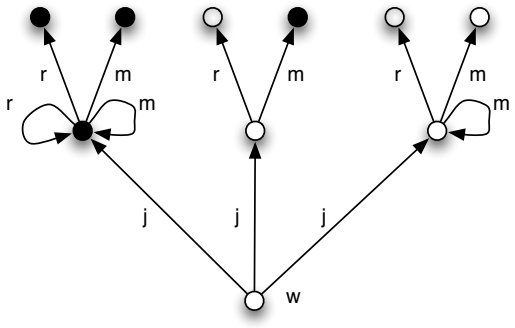


Figure 1: A model M, w. The atom p is true exactly in the worlds that are filled black.

Table 3: A semi-formal translation of the episode

a	$B_j(\forall\varphi(B_r\varphi \to \varphi) \to \forall\varphi(B_m\varphi \to \varphi))$
b	$B_j B_m(\forall\varphi(B_m\varphi \to B_c\varphi))$
c	$\neg B_j(K_j z \to B_j z) \wedge \forall\varphi(K_j\varphi \to B_j\varphi)$
d	$B_j\forall\varphi(\neg(B_c\varphi \wedge B_c\neg\varphi))$
e	$K_j B_m q \wedge B_j B_c \neg q$
f	$B_j\neg\forall\varphi(B_r\varphi \to \varphi)$

We then formalise the same episode using the relational atoms $\boxdot(\vec{a})$ introduced in Table 1, which results in Table Table 4. Abusing the language somewhat, we write $\mathit{Sup}(kj, j)$ for the relational atom corresponding to $K_j\varphi \to B_j\varphi$—from a language point of view, K_j and B_j are simply two different modal operators, say $[i]$ and $[kj]$.

Table 4: A formalisation of the episode

a'	$B_j(\mathit{Refl}(r) \to \mathit{Refl}(m))$
b'	$B_j B_m \mathit{Sup}(m, c)$
c'	$\neg B_j(K_j z \to B_j z) \wedge \mathit{Sup}(kj, k)$
d'	$B_j \mathit{Ser}(c)$
e'	$K_j B_m q \wedge B_j B_c \neg q$
f'	$B_j\neg\mathit{Refl}(r)$

3. AXIOMATIZATION

The aim of this section is to provide an axiomatisation for modal logics that are enriched with some relational atoms $\boxdot_1(\vec{a}_1), \ldots, \boxdot_m(\vec{a}_m)$, such that for every $\boxdot_k (k \leq m)$, there is a modal formula $\theta_{\boxdot_k}(\vec{a},\vec{p})$ such that, at least on frames, the two 'mean the same thing'. In fact, the logic $\mathbf{K}(A,\pi,\rho,I)$ that we define should be sound and complete with respect to $\mathcal{K}(A,\pi,\rho,I)$, so our aim for our logic is that for all formulas $\varphi \in \mathcal{L}(A,\pi,\rho)$, the notions $\mathbf{K}(A,\pi,\rho,I) \vdash \varphi$ and $\mathcal{K}(A,\pi,\rho,I) \models \varphi$ coincide. The idea to achieve this is as follows. First of all, suppose that for every relational atom $\boxdot(\vec{a})$ and fixed interpretation I we have a formula $\theta_{\boxdot}(\vec{a},\vec{p})$ such that $\theta_{\boxdot}(\vec{a},\vec{p})$ characterises $I(\boxdot(\vec{a}))$. Then, for each $\boxdot(\vec{a})$ and related $\theta_{\boxdot}(\vec{a},\vec{p})$ we add an axiom $\boxdot(\vec{a}) \to \theta_{\boxdot}(\vec{a},\vec{p})$ to our logic $\mathbf{K}(A,\pi,\rho,I)$.

Adding the other direction as an implication does not work, as the example $\boxdot(\vec{a}) = \boxdot(a,b) = Sup(a,b)$ and $\theta(\vec{a},\vec{p}) = [a]p \to [b]p$ shows: $([a]p \to [b]p) \to Sup(a,b)$ is not a validity: the antecedent may be true due to some specific choice of p. However, what we semantically *do* have is the following: suppose that we have $\mathcal{K}(A,\pi,\rho,I) \models \varphi \to ([a]p \to [b]p)$, where p does not occur in φ. This then means that φ must entail that (locally) all b successors are a successors, i.e., $\mathcal{K}(A,\pi,\rho,I) \models \varphi \to Sup(a,b)$, because if the latter would not hold, there would be a model $M = \langle W,R,I,V\rangle$ such that $M,w, \models \varphi \wedge \neg Sup(a,b)$. But since p does not occur in φ, we can change the valuation V for p freely without changing that of φ, in particular we can choose V' such that $x \in V'(p) \leftrightarrow R_a wx$ (and $V'(q) = V(q)$ for atoms $q \neq p$). It is easy to see that in the resulting model $M' = \langle W,R,I,V'\rangle$ we have $M', w \models \varphi \wedge \neg([a]p \to [b]p)$: a contradiction. This means that we need to be able to infer the following in $\mathcal{K}(A,\pi,\{Sup(a,b)\},I)$:

$$\begin{array}{l}\text{If } \mathcal{K}(A,\pi,\{Sup(a,b)\},I) \models \varphi \to ([a]p \to [b]p) \\ \text{then } \mathcal{K}(A,\pi,\{Sup(a,b)\},I) \models \varphi \to Sup(a,b), \\ \qquad\qquad\qquad\qquad \text{where } p \notin \varphi\end{array} \quad (1)$$

The rule (1) can be understood as follows. If p does not occur in φ, and $\varphi \to ([a]p \to [b]p)$ is true at a state s, then φ must carry sufficient information such that $[a]p \to [b]p$ must hold (it will not be because of specific requirements on p imposed by φ) and hence we must have $\varphi \to Sup(a,b)$ holding at s as well. But in fact we can do the same reasoning that involves *successors* of s: Suppose φ implies that in all R_c successors t of s, we have $M,t \models [a]p \to [b]p$. Then (in the same way as for s), we must have $M,t \models \varphi \to Sup(a,b)$. In other words, the following should hold for $\mathcal{K}(A,\pi,\{Sup(a,b)\},I)$:

$$\begin{array}{l}\text{If } \mathcal{K}(A,\pi,\{Sup(a,b)\},I) \models \varphi \to [c]([a]p \to [b]p) \\ \text{then } \mathcal{K}(A,\pi,\{Sup(a,b)\},I) \models \varphi \to [c]Sup(a,b), \\ \qquad\qquad\qquad\qquad \text{where } p \notin \varphi\end{array} \quad (2)$$

And the same should hold for all R_d successors u of all R_c-successors t of s, i.e., we also have a valid rule if we replace $[c]$ in (2) by $[c][d]$. But also, we have the following. Suppose that p does not occur in φ or ψ. If $M,s \models \varphi \to [c](\psi \to ([a]p \to [b]p))$, it means that if φ is true in s, then in all R_c successors t of s we have $M,t \models \psi \to ([a]p \to [b]p)$, and we have argued above that we then should also have $M,t \models \psi \to Sup(a,b)$. I.e., we have:

$$\begin{array}{l}\text{If } \mathcal{K}(A,\pi,\{Sup(a,b)\},I) \models \varphi \to [c](\psi \to ([a]p \to [b]p)) \\ \text{then } \mathcal{K}(A,\pi,\{Sup(a,b)\},I) \models \varphi \to [c](\psi \to Sup(a,b)), \\ \qquad\qquad\qquad\qquad \text{where } p \notin \varphi,\psi\end{array} \quad (3)$$

To formalise that a property $\theta_{\boxdot}(\vec{a},\vec{p})$ holds after arbitrary sequences $\varphi_1 \to [a_1](\varphi_2 \to \ldots [a_{n-1}](\varphi_n \to \theta_{\boxdot}(\vec{a},\vec{p}))\ldots)$, we follow [20] and introduce *pseudo modalities*: we will then present an inference rule $\mathbf{R}_{\boxdot}$ for every $\boxdot \in \rho$ to our axiomatisation $\mathbf{K}(A,\pi,\rho,I)$.

Definition 7 (pseudo modalities). *We define the following* pseudo modalities*, which are (possibly empty) sequences $s = ()$ or $s = (s_1,\ldots,s_n)$, where each s_i is a formula or a modality label. The formula $\langle s\rangle\varphi$ represents an $\mathcal{L}(A,\Pi,\rho)$ formula, as follows:*

$$\begin{array}{lcl}\langle()\rangle\varphi & = & \varphi \\ \langle\psi,s_2,\ldots,s_n\rangle\varphi & = & \psi \wedge \langle s_2,\ldots,s_n\rangle\varphi \\ \langle a,s_2,\ldots,s_n\rangle\varphi & = & \langle a\rangle(\langle s_2,\ldots,s_n\rangle\varphi)\end{array}$$

We also define $[s]\varphi$ as $\neg\langle s\rangle\neg\varphi$. We say that $\vec{p}$ does not occur in s (and write $\vec{p} \notin s$) if none of the atoms p occurring in $\vec{p}$ does occur in any of the formulas s_i in s.

So, for instance $\langle a,\psi,b\rangle\varphi$ is an abbreviation of $\langle a\rangle(\psi \wedge \langle b\rangle\varphi)$, while $[a,\psi,b]\varphi$ is $[a](\psi \to [b]\varphi)$.

Definition 8 (proof system). *Fix A,π and ρ. Moreover, fix an interpretation $I : \rho \to \mathcal{L}^1(A,\mathcal{X})$ such that for every $\boxdot \in \rho$, there is a $\theta_{\boxdot}(\vec{a},\vec{p})$ such that the modal formula $\theta_{\boxdot}(\vec{a},\vec{p})$ characterises the first order formula $I(\boxdot)$. Then, the following comprises the axioms and inference rules of the logic $\mathbf{K}(A,\pi,\rho,I)$*

Prop *All instances of propositional tautologies*

K $[a](\varphi \to \psi) \to ([a]\varphi \to [a]\psi)$

Ax$_{\boxdot}$ $\boxdot(\vec{a}) \to \theta_{\boxdot}(\vec{a},\vec{p})$

MP *From $\varphi \to \psi$ and φ, infer ψ*

Nec *From φ, infer $[a]\varphi$*

R$_{\boxdot}$ *From $\langle s\rangle\neg\theta_{\boxdot}(\vec{a},\vec{p}) \to \varphi$, infer $\langle s\rangle\neg\boxdot(\vec{a}) \to \varphi$, where $\vec{p}$ does not occur in φ or s.*

US *From φ infer $\varphi[\psi/p]$.*

MP *stands for Modus Ponens,* **Nec** *for Necessitation, and* **US** *for Uniform Substitution ($\varphi[\psi/p]$ stands for substitution of ψ for every occurrence of p in φ). If $\boxdot(\vec{a})$ and $\theta_{\boxdot}(\vec{a},\vec{p})$ are connected through the axiom* **Ax**$_{\boxdot}$ *and inference rule* **R**$_{\boxdot}$*, we say they are axiomatically linked (through axiom* **Ax**$_{\boxdot}$ *and rule* **R**$_{\boxdot}$*). If there is a derivation of a formula φ from a set of formulas Γ using Γ and the axioms and inference rules from $\mathbf{K}(A,\pi,\rho,I)$ we write $\Gamma \vdash_{\mathbf{K}(A,\pi,\rho,I)} \varphi$, or $\Gamma \vdash_{\mathbf{K}} \varphi$, for short.*

Theorem 1 (Soundness). *For all $\varphi \in \mathcal{L}(A,\pi,\rho)$, if $\mathbf{K}(A,\pi,\rho,I) \vdash \varphi$ then $\mathcal{K}(A,\pi,\rho,I) \models \varphi$.*

3.1 Back to Our Example

To formalise the derivation of Table 4, let the set of modalities $A = \{c,j,m,p,r\}$, let $\pi = \{q,z\}$ and let $\rho = \{Refl(r), Refl(m), Sup(c,m), Ser(c), Sup(kj,j)\}$ and those atoms are axiomatically linked with their 'natural' modal counterparts (see Table 1 and for $KimplB(j)$ we take $K_j p \to B_j p$). Let the resulting logic be $\mathbf{K}(A,\pi,\rho,I)$.

First of all, from c' and $\mathbf{Ax}_{KimplB(j)}$ we derive $K_j B_m q \to B_j B_m q$. Together with e' this gives e'': $B_j B_m q \wedge B_j B_c \neg q$. From d', i.e., $B_j Ser(c)$ and $\mathbf{Ax}_{Ser(c)}$, we get $B_j(B_c\neg q \to \neg B_c q)$. Combining this with e'' gives $B_j B_m q \wedge B_j \neg B_c q$, which is equivalent to $B_j\neg(B_m q \to B_c q)$ (*).

From b' and $\mathbf{Ax}_{Sup(m,c)}$ we derive $B_j B_m(B_m p \to B_c p)$, for any p (**). Now, take the formula $\psi = (B_m q \to B_c q)$. From (*) we have $B_j\neg\psi$, and from (**) we conclude $B_j B_m \psi$. In other words, we found a formula ψ for which $B_j\neg(B_m\psi \to \psi)$. Now using the contrapositive of axiom $\mathbf{Ax}_{Refl(m)}$, we obtain $B_j\neg Refl(m)$, which is our conclusion f'.

Now one may wonder whether this also warrants the conclusion f, but as should be clear from Remark 1, $B_j\neg Refl(m)$ and $B_j\neg\forall\varphi(B_m\varphi \to \varphi)$ are not the same thing. However, what we *do* have is the following. Let φ be $a' \wedge b' \wedge c' \wedge d' \wedge e'$, and let s be B_j, then what we have proven now is

$$\mathbf{K}(A,\pi,\rho,I) \vdash \varphi \to B_j\neg Refl(m) \quad (4)$$

But from this, if not *derive*, we can safely *assume* that, given φ, there is some formula ψ for which $B_j(B_m\psi \wedge \neg\psi)$, because if this were not the case, we would have, for some atom p not occuring in φ:

$$\mathbf{K}(A,\pi,\rho,I) \vdash \varphi \rightarrow B_j(B_m p \rightarrow p) \quad (5)$$

From which, using rule $\mathbf{R}_{\boxdot}(\vec{a})$, we would conclude

$$\mathbf{K}(A,\pi,\rho,I) \vdash \varphi \rightarrow B_j Refl(m) \quad (6)$$

which either means we have a derivation for $B_j\bot$ (Joey believes anything), or, if we assume the conjunct $g' = Ser(j)$ to be also part of φ (expressing, that actually, Joey's beliefs are consistent), that we have derived a contradiction with (4).

It is worth noting how the axiomatisation makes it possible that some relational atoms (and hence some first-order frame properties) only hold in the scope of a modal operator (like in property a' and b' for example): the axiom $\mathbf{Ax}_{\boxdot}$ and rule $\mathbf{R}_{\boxdot}$ do not require that some relational properties hold, they only specify what should be the case *if* they hold.

3.2 Completeness

DEFINITION 9. *A* theory Γ *is a set of formulas. For* π *a set of propositional atoms,* Γ *is a* π*-theory if all propositional atoms in* Γ *are from* π*. Given a logic* $\mathbf{L}$*, a theory* Γ *is* $\mathbf{L}$*-consistent if* $\bot$ *cannot be derived from* Γ *using the axioms and inference rules of* $\mathbf{L}$*. A theory* Γ *is a maximal* $\mathbf{L}$*-consistent* π*-theory if it is consistent and no* π*-theory* Δ *is* $\mathbf{L}$*-consistent while at the same time* $\Gamma \subset \Delta$*. For a logic* $\mathbf{K}(A,\pi,\rho,I)$*, a set of formulas* Γ *is a* witnessed π*-theory if for every* $\langle s\rangle\neg\boxdot(\vec{a}) \in \Gamma$*, there are atoms* $\vec{p}$ *such that* $\langle s\rangle\neg\theta_{\boxdot}(\vec{a},\vec{p}) \in \Gamma$*, where* $\boxdot(\vec{a})$ *and* $\theta_{\boxdot}(\vec{a},\vec{p})$ *are axiomatically linked. If* Γ *is not witnessed, then a formula* $\langle s\rangle\neg\boxdot(\vec{a})$ *for which there is no* $\langle s\rangle\neg\theta_{\boxdot}(\vec{a},\vec{p}) \in \Gamma$*, is called a* defect *for the theory* Γ*. Finally,* Γ *is said to be* fully witnessed*, if it is witnessed and for every formula of the form* $\langle s\rangle\varphi$*, either that formula or its negation is in* Γ*.*

LEMMA 2 (EXTENSION LEMMA). *Let* Σ *be a* $\mathbf{K}(A,\pi,\rho,I)$*-consistent* π*-theory. Let* $\pi' \supseteq \pi$ *be an extension of* π *by a countable set of propositional variables. Then there is a maximal* $\mathbf{K}(A,\pi',\rho,I)$*-consistent, witnessed* π'*-theory* Σ' *extending* Σ*.*

Before we outline a proof, we first define some languages.

DEFINITION 10. *Let the set of agents* A*, the set of atoms* π *and the set of relational atoms* ρ *be fixed. Let* $\mathcal{L}(A,\pi,\rho)$ *be as in Definition 1. Let* $\pi^0 = \{p_0, p_1, \dots\}$ *be a set of fresh atomic variables, i.e.,* $\pi \cap \pi^0 = \emptyset$ *and let* $\pi' = \pi \cup \pi^0$*. Let* $\pi_n = \pi \cup \{p_i \mid i \leq n\}$*. Define* $\mathcal{L}_n$ *to be* $\mathcal{L}(A,\pi_n,\rho)$*, and let* $\mathcal{L}_\omega$ *be* $\mathcal{L}(A,\pi',\rho)$*. A theory* $\Delta \subseteq \Sigma$ *is called an* approximation *if for some* n *it is a consistent* π_n*-theory. For such a theory, and any number* k*, the sequence* $\vec{p} = \langle p_{n+1}, \dots, p_{n+k}\rangle$ *is a new sequence* $\vec{p}$ *for* Δ *if* n *is the least number such that* Δ *is a* π_n*-theory.*

PROOF OF LEMMA 2 (SKETCH). Assume an enumeration of $\psi_0, \psi_1, \dots$ of all formulas of the form $\langle s\rangle\neg\boxdot(\vec{a})$, where s is a pseudo modality and $\boxdot(\vec{a}) \in \rho$. Define

$$\Delta^+ = \begin{cases} \Delta \cup \{\langle s\rangle(\neg\theta_{\boxdot}(\vec{a},\vec{p}))\} & \text{where } \vec{p} \text{ is a new sequence for } \Delta, \text{ and } \langle s\rangle\neg\boxdot(\vec{a}) \text{ is the first defect for } \Delta, \text{ if this exists} \\ \Delta & \text{otherwise} \end{cases}$$

Clearly, by $\mathrm{Ax}_{\boxdot}$, the set Δ^+ is consistent when Δ is and hence, if Δ is an approximation, so is Δ^+. To define the extension Σ' of Σ, assume $\varphi_0, \varphi_1, \dots$ to be an enumeration of the formulas in $\mathcal{L}_\omega$, and define $\Sigma_0 = \Sigma$, and

$$\begin{aligned} \Sigma_{2n+1} &= \begin{cases} \Sigma_{2n} \cup \{\varphi_n\} & \text{if this is consistent} \\ \Sigma_{2n} \cup \{\neg\varphi_n\} & \text{else} \end{cases} \\ \Sigma_{2n+2} &= (\Sigma_{2n+1})^+ \end{aligned}$$

Finally, let $\Sigma' = \bigcup_{n\in\omega} \Sigma_n$. By construction, $\Sigma' \supset \Sigma$ is a maximal $\mathbf{K}(A,\pi,\rho,I)$-consistent, witnessed π'-theory. □

DEFINITION 11 (CANONICAL MODEL). *The canonical model* $M^c = (W^c, R^c, I, V^c)$ *for the logic* $\mathbf{K}(A,\pi,\rho,I)$ *has:*

- $W^c = \{\Gamma \mid \Gamma$ *is a maximal* $\mathcal{L}_\omega$*-consistent witnessed* π'*-theory*$\}$*;*
- $R^c_a\Gamma\Delta$ *iff for all* $\varphi \in \mathcal{L}_\omega$ *it holds that if* $[a]\varphi \in \Gamma$*, then* $\varphi \in \Delta$*;*
- I *as given as a parameter of the logic;*
- $V^c_p = \{\Gamma \mid p \in \pi' \cap \Gamma\}$*.*

LEMMA 3. *Suppose the following holds:*

1. $\theta_{\boxdot}(\vec{a},\vec{p})$ *is a purely modal formula;*
2. $\theta_{\boxdot}(\vec{a},\vec{p})$ *and* $\boxdot(\vec{a})$ *are linked through the rule* $R_{\boxdot}$ *and the axiom* $\mathbf{Ax}_{\boxdot}$*;*
3. *The first order formula* $I(\boxdot(\vec{a})) = \Theta(\vec{a})(x)$ *is equivalent with the second order formula* $\forall\vec{P}\hat{\Theta}(\vec{a},\vec{P})(x)$ *where* $\hat{\Theta}(\vec{a},\vec{P})(x)$ *is* $\mathsf{ST}(\theta_{\boxdot}(\vec{a},\vec{p}))(x)$*.*

Then, in the canonical model, $\boxdot(\vec{a})$ *and* $\Theta(\vec{a})(x)$ *are connected as in Definition 4, i.e., for all* $\Delta \in M^c$ *we have* $M^c, \Delta \models \boxdot(\vec{a})$ *iff in* M^c *it holds that* $I(\boxdot(\vec{a}))(\Delta)$*.*

Lemma 3 paves the way for a coincidence lemma that guarantees that membership and truth in the canonical model coincide. We then get:

THEOREM 2. *If the assumptions of Lemma 3 hold, the logic* $\mathbf{K}(A,\pi,\rho,I)$ *is sound and complete with respect to the class of* $\mathcal{K}(A,\pi,\rho,I)$ *models.*

DEFINITION 12. *A purely modal formula* φ *is* canonical for *a first order property* Φ*, if the canonical model for the modal logic* $(\mathbf{K}+\varphi)(A,\pi,\rho,I)$ *has the property* Φ*.*

There are many examples of canonical formulas: all Sahlqvist formulas are canonical [8].

THEOREM 3. *Let* φ_i *be canonical for* Φ_i*,* $i \leq n$*. Then the logic* $(\mathbf{K}+\varphi_1, \dots, \varphi_n)(A,\pi,\rho,I)$ *is sound and complete with respect to all models in* $\mathcal{K}(A,\pi,\rho,I)$ *that satisfy* $\Phi_1, \dots, \Phi_n$*.*

4. CONCLUSION

First, note that our modelling of locality is different from local *frame* correspondence as defined in [3], and quite distant from the use of *local propositions* in epistemic logic [6], propositions that do not change truth value within an agent's equivalence class .

We have so far assumed that the properties of $\boxdot(\vec{a})$ are those specified by the axiom $\mathbf{A}_{\boxdot}$ and rule $\mathbf{R}_{\boxdot}$. However, one

can add connections between $\boxdot(\vec{a})$ and modal formulas, or between different $\boxdot_1(\vec{a}_1)$ and $\boxdot_2(\vec{a}_2)$ atoms. For instance

$$\mathit{Refl}(a,a) \rightarrow \mathit{Trans}(a,a,a) \tag{7}$$

added to an epistemic logic has the effect that whenever a's knowledge is veridical, a is also positively introspective. I.e., we would have, semantically, that whenever $M, s \models K_a\varphi \rightarrow \varphi$, for all φ, then also $M, s \models K_a\varphi \rightarrow K_aK_a\varphi$, for all φ. This again is a property that cannot be expressed in standard, 'global' modal logic. As a second example, in an epistemic temporal modal logic, one could add an axiom

$$\mathit{Trans}(a,a,a) \rightarrow F(\mathit{Trans}(a,a,a) \wedge \mathit{Eucl}(a,a,a)) \tag{8}$$

saying that whenever agent a is positively introspective, he will eventually also become negatively introspective. As a third example, a simple axiom like

$$\mathit{Ser}(a) \rightarrow \mathit{Ser}(b) \tag{9}$$

in a doxastic setting would mean that whenever a's beliefs are consistent, those of b must be consistent as well.

It is possible to view some standard results in modal logic concerning completeness of modal systems as obtained as special cases from our local logic. If the conditions of Theorem 2 are satisfied, and one adds a $\boxdot(\vec{a})$ as an *axiom*, one immediately gets completeness with respect to the class of models that satisfy $I(\boxdot(\vec{a}))$. For instance, in a logic with axioms and rules for $\mathit{Refl}(a)$, adding $\mathit{Refl}(a)$ itself as an axiom gives a modal system that is sound and complete with respect to the class of reflexive Kripke models! Of course, this amounts to the same thing as adding $\theta_{\boxdot}(\vec{a},\vec{p})$, as is directly clear from rule $\mathbf{R}_{\boxdot}$ (take $\varphi = \bot$ and s the empty sequence).

Finally, it is important to realise that, although we presented the axioms for the underlying logic (the formulas φ_i that we assumed to be canonical) and the relational atoms as two independent layers, let us recall that [20] showed that interaction properties between the modalities and the relational atoms may be automatically 'imported'. For the case of epistemic logic **S5** with at least two agents and the $\mathit{Sup}(a,b)$ atom, one can derive that $\mathit{Sup}(a,b) \rightarrow K_b\mathit{Sup}(a,b)$ and $\neg\mathit{Sup}(a,b) \rightarrow K_b\neg\mathit{Sup}(a,b)$, in other words, in such a logic, it is derivable that if agent a considers at least the states possible that b considers possible, then b knows this! Similarly, if there is a state considered possible by b but not by a, then agent b knows this as well!

To summarise, we have presented a flexible way to deal locally with quantification over formulas. In particular, we have shown how, under some mild conditions, in a modal logic that extends **K** with some canonical axioms, one can add a number of relational atoms, for each of them an axiom and an inference rule, such that the logic is complete for the class of models that interpret the atom as a first order property of the underlying frame. We argued that this presents many opportunities to express properties concerning the knowledge or beliefs of agents in a local way, so that they are only true now, or as a belief or knowlege of some specific agents. Although we focussed on epistemic and doxastic logics, our technique is applicable in temporal and dynamic settings as well. On our agenda is to study how our framework behaves in a dynamic epistemic logic setting. For instance, one might consider the effect of publicly announcing relational atoms, like $\mathit{Sup}(a,b)$, which would mean that it is announced that a knows at least what b knows. After such an announcement, one would expect that the local property becomes global again, in many cases it would become a validity in the resulting model that $\forall y(R_bxy \Rightarrow R_axy)$.

Like we explained, our completeness proof borrows ideas from both [16] and [5]. Also, the inference rule $\mathbf{R}_{\boxdot}$ is reminiscent of an inference rule for irreflexivity [9]. However, it is important to stress that the approaches mentioned aim to axiomatise *global* properties. As far as we know, the work presented in this paper is a first general approach to *local* properties in models.

We thank the AAMAS reviewers for their helpful comments.

5. REFERENCES

[1] A. Baltag, L. S. Moss, and S. Solecki. The logic of common knowledge, public announcements, and private suspicions. *TARK 98*, pp. 43–56, 1998.

[2] P. Blackburn, J. van Benthem, and F. Wolter, (eds). *Handbook of Modal Logic*. Elsevier, Amsterdam, 2007.

[3] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53. CUP, Cambridge, 2001.

[4] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[5] M. de Rijke. *Extending Modal Logic*. PhD thesis, ILLC, University of Amsterdam, 1993.

[6] K. Engelhardt, R. van der Meyden and Y. Moses. Knowledge and the logic of local propositions. *TARK 98*, pp. 29–41, 1998.

[7] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[8] M. Fitting. Modal proof theory. In Blackburn et al. [2], pp. 85 – 138.

[9] D. Gabbay. An irreflexivity lemma with applications to axiomatisations of conditions on tense frames. In *Aspects of Philosophical Logic*, pp. 67–89. Reidel, 1981.

[10] J. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *JACM*, 37(3):549–587, 1990.

[11] D. Harel. Dynamic logic. In *Handbook of Philosophical Logic Vol. II*, pp. 497–604, 1984.

[12] J. Hintikka. *Knowledge and Belief*, 1962.

[13] J.-J. C. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. CUP, 1995.

[14] R. C. Moore. Reasoning about knowledge and action. In *IJCAI-77*, Cambridge, MA, 1977.

[15] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In *KR&R-91*, pp. 473–484, 1991.

[16] G. Renardel de Lavalette, B. Kooi, and R. Verbrugge. Strong completeness for PDL. In *AiML2002*, pp. 377–393, 2002.

[17] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, University of Amsterdam, 1976.

[18] W. van der Hoek and M. Pauly. Modal logic for games and information. [2], pages 1077–1148.

[19] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, Berlin, 2007.

[20] H. van Ditmarsch, W. van der Hoek, and B. Kooi. Knowing more — towards a local correspondence theory. In *IJCAI-09*, pp. 955–960, 2009.

[21] M. Wooldridge. *An Introduction to Multi-agent Systems*. John Wiley & Sons, 2002.

Knowledge and Control

Wiebe van der Hoek
Dept of Computer Science
University of Liverpool, UK
wiebe@csc.liv.ac.uk

Nicolas Troquard
School of Computer Science
and Electronic Engineering
University of Essex, UK
ntroqu@essex.ac.uk

Michael Wooldridge
Dept of Computer Science
University of Liverpool, UK
mjw@csc.liv.ac.uk

ABSTRACT

Logics of propositional control, such as van der Hoek and Wooldridge's CL-PC [14], were introduced in order to represent and reason about scenarios in which each agent within a system is able to exercise unique control over some set of system variables. Our aim in the present paper is to extend the study of logics of propositional control to settings in which these agents have incomplete information about the society they occupy. We consider two possible sources of incomplete information. First, we consider the possibility that an agent is only able to "read" a subset of the overall system variables, and so in any given system state, will have partial information about the state of the system. Second, we consider the possibility that an agent has incomplete information about which agent controls which variables. For both cases, we introduce a logic combining epistemic modalities with the operators of CL-PC, investigate its axiomatization, and discuss its properties.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

epistemic logic, propositional control, partial observability

1. INTRODUCTION

The *Coalition Logic of Propositional Control* (CL-PC) was introduced by van der Hoek and Wooldridge as a formalism for reasoning about how agents and coalitions can exercise control in multiagent environments [14]. The logic models situations in which each agent has control over some set of propositions; that is, each agent is associated with some set of propositions, and has the ability to assign a (truth) value to each of its propositions. In this way, valuations become possible worlds (see e.g., [15] for an early treatment of such modelings). The language of CL-PC provides modal constructs $\Diamond_i\varphi$ to express the fact that, under the assumption that the rest of the system remains unchanged, agent i can assign values to the propositions under its control in such a way that φ becomes true; these operators are closely related to the strategic ability operators in cooperation logics such as ATL [2] and Coalition Logic [11]. Since the logic was originally presented, a number of variants of CL-PC have been developed. For example: the logic DCL-PC is an extension to CL-PC in which agents are able to transfer the control of their variables by executing *transfer programs* [12]; and Gerbrandy studied generalisations of CL-PC, allowing for instance situations in which agents have "partial" control of propositions [7].

Our aim in this paper is to study one rather obvious aspect of propositional control logics that has hitherto been neglected: *the interaction between knowledge and control*. It is indeed surprising that this aspect of propositional control logics has not been previously studied in the literature. After all, the interaction between knowledge and ability has a venerable history in the artificial intelligence community, going back at least to the work of Moore in the late 1970s [10]. Moore was interested in *knowledge pre-conditions*: what an agent needs to know in order to be able to do something. To use a standard example, in order to be able to open a safe, you need to know the combination. He formalised a notion of ability that was able to capture such subtleties in a logic that combined elements of dynamic and epistemic logic. More recently, the interplay between ability and knowledge has been studied with respect to cooperation logics such as ATL and Coalition Logic. For example, van der Hoek and Wooldridge proposed ATEL, a variant of ATL extended with epistemic modalities [13]; and various authors developed variants of ATEL intended to rectify some counterintuitive properties of the original ATEL proposal [8, 1].

Epistemic logic is, ultimately, a logic modelling (un)certainty [6]. When we say an agent knows φ, we typically mean that the agent is certain about φ. This notion of uncertainty is elegantly captured in possible worlds semantics, where knowing φ means that φ is true in all worlds that the agent considers possible. If we turn to CL-PC, we can identify several different sources of uncertainty, as follows.

First, and most obviously, an agent may be uncertain about the value of the variables in the system. We call this type of uncertainty *partial observability*, and it is very naturally modelled by assigning to every agent a set of variables that the agent is able to "see". Partial observability interacts with control in several important ways. For example, if I control the variable q and my goal is to achieve the formula $p \leftrightarrow \neg q$, then if I can observe the value of p, I can readily choose a value for q that will result in my goal being achieved: I simply choose the opposite to the value of p. However, if I cannot see the value of p, then I am in trouble. Second, and perhaps more unusually, there may be uncertainty about *which agent controls which variables*. Here, we might conceivably have a situation in which an agent is able to bring about some state of affairs, but does not know that they are able to bring it about, because it is not

Cite as: Knowledge and Control, W. van der Hoek, N. Troquard and M. Wooldridge, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 719-726.

aware that it controls the appropriate variables.

The aim of the present paper is to develop extensions to CL-PC that are able to capture these two types of uncertainty. The remainder of the paper is structured as follows. After presenting some definitions that will be used throughout the remainder of the paper, in Section 2, we present the epistemic extension to CL-PC for the case that agents have complete knowledge about how the control of variables is actually distributed over the agents, but they may lack information about what is factually true. Subsequently, in Section 3, we then look at formalising the case where agents have full knowledge about factual truth, partial knowledge about who controls what, and are completely ignorant about other's information regarding control. We also sketch an even more general setting where both factual truth and control may be uncertain. We conclude in Section 4.

We begin with some definitions, which are used throughout the remainder of the paper. First, let $\mathbb{B} = \{true, false\}$ be the set of Boolean truth values. We assume that the domains we model contain a (finite, non-empty) set $N = \{1, \ldots, n\}$ of agents ($|N| = n$, $n > 0$). The environment is also assumed to contain a (fixed, finite) set $\mathbb{A} = \{p, q, \ldots\}$ of *Boolean variables*. Each agent $i \in N$ will be assumed to *control* some subset $\mathbb{A}_i$ of atoms $\mathbb{A}$, with the intended interpretation that if $p \in \mathbb{A}_i$, then i has the unique ability to assign a value ($true$ or $false$) to p. We require that the sets $\mathbb{A}_i$ form a partition of $\mathbb{A}$, i.e., $\mathbb{A}_i \cap \mathbb{A}_j = \emptyset$ for $i \neq j$, and $\mathbb{A}_1 \cup \cdots \cup \mathbb{A}_n = \mathbb{A}$. Thus every variable is controlled by some agent; and no variable is controlled by more than one agent. A *coalition* is simply a set of agents, i.e., a subset of N. We typically use $C, C', \ldots$ as variables standing for coalitions. Where $C \subseteq N$, we denote by $\mathbb{A}_C$ the set of variables under the collective control of the agents in C: $\mathbb{A}_C = \bigcup_{i \in C} \mathbb{A}_i$. A *valuation* is a total function $\theta : \mathbb{A} \to \mathbb{B}$, which assigns a truth value to every Boolean variable. Let Θ denote the set of all valuations. Where C is a coalition, a C-valuation is a function $\theta_C : \mathbb{A}_C \to \mathbb{B}$; thus a C-valuation is a valuation to variables under the control of the agents in C. Given a set X of atoms and two valuations θ_1 and θ_2, we write $\theta_1 \equiv_X \theta_2$ to mean that θ_1 and θ_2 agree on the value of all variables in X, i.e., $\theta_1(p) = \theta_2(p)$ for all $p \in X$.

2. PARTIAL OBSERVABILITY

In this section, we develop an *Epistemic Coalition Logic of Propositional Control with Partial Observability* – ECL-PC(PO) for short. This logic is essentially CL-PC extended with epistemic modalities K_i, one for each agent $i \in N$. These epistemic modalities have a conventional (S5) possible worlds semantics. The interpretation we give to epistemic accessibility relations is as follows. We assume each agent $i \in N$ is able to see a subset $V_i \subseteq \mathbb{A}$ of the overall set of Boolean variables; that is, it is able to correctly perceive the value of these variables. A valuation θ' is then i-accessible from valuation θ if θ and θ' agree on the valuation of variables visible to i, i.e., $\theta \equiv_{V_i} \theta'$. Formally, the language of ECL-PC(PO) is defined by the following BNF grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_i \varphi \mid K_i \varphi$$

where $p \in \mathbb{A}$, and $i \in N$. As in CL-PC [14], a formula $\Diamond_i \varphi$ means that i can assign values to the variables under its control in such a way that, assuming no other variables are changed, φ becomes true. As in epistemic logic [6], a formula $K_i \varphi$ means that the agent i knows φ.

The remaining operators of classical logic ("$\wedge$" – and, "$\rightarrow$" – implies, "$\leftrightarrow$" – iff) are assumed to be defined as abbreviations in terms of $\neg, \vee$ as usual. We define the box dual operator of $\Diamond_i$ as: $\Box_i \varphi \equiv \neg\Diamond_i \neg\varphi$. We also assume the existential dual M_i ("maybe") of the K_i operator is defined as: $M_i \varphi \equiv \neg K_i \neg\varphi$. For coalitions, we define (this definition is justified in [14]):

$$\Box_{\{1,\ldots,k\}} \varphi \equiv \Box_1 \ldots \Box_k \varphi.$$

Coming to the semantics, a *frame* for CL-PC is simply a structure $\langle N, \mathbb{A}_1, \ldots, \mathbb{A}_n \rangle$, where N is the set of agents in the system, and each $\mathbb{A}_i$ is the set of variables under the control of agent i; a model for CL-PC combines such a frame with a valuation $\theta \in \Theta$, which gives an initial value for every Boolean variable [14]. Frames for ECL-PC(PO) extend CL-PC frames with a set of variables $V_i \subseteq \mathbb{A}$ for each agent $i \in N$. Formally, an ECL-PC(PO) frame, F, is a $(2n+1)$-tuple

$$F = \langle N, \mathbb{A}_1, \ldots, \mathbb{A}_n, V_1, \ldots, V_n \rangle, \text{ where}$$

- $N = \{1, 2, \ldots, n\}$ is a (finite, nonempty) set of agents.
- The sets $\mathbb{A}_i$ form a partition of $\mathbb{A}$.
- $V_i \subseteq \mathbb{A}$ is the set of atoms whose values are visible to i.

It will often make sense to assume $V_i \supseteq \mathbb{A}_i$, i.e., each agent can see the value of the variables it controls; however, we will not impose this as a requirement. We leave aside the question for now of what settings there are in which this assumption does not hold.

The truth value of an ECL-PC(PO) formula is inductively defined wrt. a frame F and a valuation θ by the following rules ($\models^d$ stands for a 'direct semantics', [14]):

$$\begin{array}{lll}
F, \theta \models^d p & \text{iff} & \theta(p) = true \qquad (p \in \mathbb{A}) \\
F, \theta \models^d \neg\varphi & \text{iff} & F, \theta \not\models^d \varphi \\
F, \theta \models^d \varphi \vee \psi & \text{iff} & F, \theta \models^d \varphi \text{ or } F, \theta \models^d \psi \\
F, \theta \models^d \Diamond_i \varphi & \text{iff} & \exists \theta' \in \Theta : \theta' \equiv_{\mathbb{A} \setminus \mathbb{A}_i} \theta \text{ s.t. } M, \theta' \models^d \varphi \\
F, \theta \models^d K_i \varphi & \text{iff} & \forall \theta' \in \Theta : \theta' \equiv_{V_i} \theta \Longrightarrow M, \theta' \models^d \varphi
\end{array}$$

We denote the fact that φ is true in all models by $\models^d \varphi$. We let $\Lambda_1 = \{\varphi \mid \models^d \varphi\}$ be the logic of all the formulas valid in all ECL-PC(PO) models.

EXAMPLE 1. *Suppose we have a frame F with two agents, $N = \{1, 2\}$ and two Boolean variables, $\mathbb{A} = \{p, q\}$, with $\mathbb{A}_1 = V_1 = \{p\}$ and $\mathbb{A}_2 = \{q\}$ and $V_2 = \{p, q\}$. Thus agent 1 can only see the value of the variable it controls, while agent 2 can see the values of both variables. Let $\theta(p) = \theta(q) = true$. Now, we have:*

- $F, \theta \models^d \Diamond_1 (p \leftrightarrow \neg q)$

  Agent 1 can set his variable p in such a way that p and q have different values.

- $F, \theta \models^d \neg K_1 q \wedge \neg K_1 \neg q \wedge K_1 (K_2 q \vee K_2 \neg q)$

  Agent 1 does not know the value of variable q, but he does know that 2 knows the value of q.

- $F, \theta \models^d K_1 \Diamond_1 (p \leftrightarrow \neg q) \wedge \neg\Diamond_1 K_1 (p \leftrightarrow \neg q)$

  Agent 1 knows that he can make p and q take on different values (because he controls p, and hence can make it different to q in any given state). However, agent 1 cannot choose values for the variables he controls in such a way that he knows that p and q take on different values.

- $F, \theta \models^d K_2 \Box_1 ((K_2 p \vee K_2 \neg p) \wedge (K_2 q \vee K_2 \neg p))$

  Agent 2 knows that whatever truth values 1 chooses for her variables, 2 will know the value of p and of q.

- $F, \theta \models^d K_2 ((p \wedge q) \wedge \Diamond_1 (\neg p \wedge \Diamond_2 (\neg p \wedge \neg q)))$ *Agent 2 knows that $(p \wedge q)$ and that 1 can bring about that $\neg p$ which 2 can further narrow down to $(\neg p \wedge \neg q)$.*

CLPC	
$(Prop)$	φ , where φ is a propositional tautology
$(K(\Box))$	$\Box_i(\varphi \to \psi) \to (\Box_i\varphi \to \Box_i\psi)$
$(T(\Box))$	$\Box_i\varphi \to \varphi$
$(B(\Box))$	$\varphi \to \Box_i\Diamond_i\varphi$
$(empty)$	$\Box_\emptyset\varphi \leftrightarrow \varphi$
$(comp\cup)$	$\Box_{C_1}\Box_{C_2}\varphi \leftrightarrow \Box_{C_1\cup C_2}\varphi$
$(confl)$	$\Diamond_i\Box_j\varphi \to \Box_j\Diamond_i\varphi$
$(exclu)$	$(\Diamond_i p \wedge \Diamond_i\neg p) \to (\Box_j p \vee \Box_j\neg p)$, where $j \neq i$
$(actual)$	$\bigvee_{i\in N} \Diamond_i p \wedge \Diamond_i\neg p$
$(full\Box)$	$(\bigwedge_{p\in X} \Diamond_i p \wedge \Diamond_i\neg p) \to \Diamond_i\varphi_X$
Knowledge	
$(K(K))$	$K_i(\varphi \to \psi) \to (K_i\varphi \to K_i\psi)$
$(T(K))$	$K_i\varphi \to \varphi$
$(B(K))$	$\varphi \to K_iM_i\varphi$
$(4(K))$	$K_i\varphi \to K_iK_i\varphi$
$(incl)$	$\Box_N\varphi \to K_i\varphi$
$(unif)$	$M_ip \wedge M_i\neg p \to \Box_N(M_ip \wedge M_i\neg p)$
$(fullK)$	$(\bigwedge_{p\in X} M_ip \wedge M_i\neg p) \to M_i\varphi_X$
Rules	
(MP)	from $\vdash \varphi \to \psi$ and $\vdash \varphi$ infer $\vdash \psi$
$(Nec(\Box))$	from $\vdash \varphi$ infer $\vdash \Box_i\varphi$

Figure 1: Axiomatics of Λ_1. The meta-variable i ranges over N, C_1 and C_2 over 2^N, φ represents an arbitrary formula of ECL-PC(PO), p ranges over $\mathbb{A}$. φ_X is the conjunction of literals true in any valuation of $X \subseteq \mathbb{A}$.

When studying a new logic, there are two key computational problems we must consider: the *model checking* problem and the *satisfiability* problem. For ECL-PC(PO), the model checking problem is the problem of determining, for a given frame and valuation F, θ and formula φ, whether or not $F, \theta \models^d \varphi$. The satisfiability problem is the problem of determining whether, for a given formula φ there exists a frame F and valuation θ such that $F, \theta \models^d \varphi$. It was proved in [14] that both the model checking and satisfiability problems for the underlying logic CL-PC are PSPACE-complete. The fact that the model checking problem is PSPACE complete in fact yielded a decision problem for satisfiability: because the frames F are "small", we can exhaustively search the set of possible frames and valuations for a formula, checking each pair in turn to see whether it satisfies the formula. Observe that the model checking problem for ECL-PC(PO) is trivially seen to be solvable in polynomial space. Then basically the same approach for satisfiability checking of CL-PC also works for ECL-PC(PO): the truth of a formula only depends on at most one more agent than is named in the formula, as with CL-PC [14], and so we can exhaustively examine each F, θ pair to see whether $F, \theta \models^d \varphi$. We may conclude:

THEOREM 1. *The model checking and satisfiability problems for* ECL-PC(PO) *are both PSPACE-complete.*

An axiomatization for ECL-PC(PO) is provided in Figure 1. Several points are in order with respect to this axiomatization. First, note that K_i is an S5 modality, and that the axiom 4 for the modality $\Box_i$ is an instance of axiom $(comp\cup)$. With $K(\Box)$, $T(\Box)$ and $B(\Box)$ this implies that $\Box_i$ is also an S5 modality.

LEMMA 1. *The axiomatization for Λ_1 in Figure 1 is sound.*

We now prove that this axiomatization is complete. This will be done using a normal form for ECL-PC(PO)-formulas.

DEFINITION 1. *We define $ctrls(i,p)$ as $(\Diamond_i p \wedge \Diamond_i\neg p)$ and $sees(i,p)$ as $(K_ip \vee K_i\neg p)$. Let $CTRL = \{ctrls(i,p) \mid i \in N \,\&\, p \in \mathbb{A}\}$ and $VIEW = \{sees(i,p) \mid i \in N \,\&\, p \in \mathbb{A}\}$. The elements of $\mathbb{A}$, $CTRL$ and $VIEW$ are called basic propositions. For any set Φ of basic propositions, call $L(\Phi) = \{x, \neg x \mid x \in \Phi\}$ the set of literals over Φ. For a basic proposition x, let $\ell(x) \in \{x, \neg x\}$. So e.g., $\ell(p) \to \Diamond_i\ell(p)$ stands both for $p \to \Diamond_i p$ and for $\neg p \to \Diamond\neg p$. A* propositional description *π is a conjunction over $L(\mathbb{A})$ where each $p \in \mathbb{A}$ occurs exactly once. Let Π be the set of propositional descriptions. A* control description *γ is a conjunction over $CTRL$ such that for every $p \in \mathbb{A}$, there is exactly one $i \in N$ such that $ctrls(i,p)$ occurs in γ. Let Γ be the set of control descriptions. Finally, a* visibility description *ς is a conjunction over $L(VIEW)$, such that for every agent i and every atom $p \in \mathbb{A}$, either $sees(i,p)$ or $\neg sees(i,p)$ occurs in ς. Let Σ be the set of visibility descriptions. A* full description *is a conjunction $\pi \wedge \gamma \wedge \varsigma$, where π, γ and ς are as explained above.*

Given a propositional description $\pi \in \Pi$, we shall note $\hat{\pi}^i$ the conjunction of literals in π that are under the control of agent i and $\check{\pi}^i$ the conjunction of literals in π that are not under its control. Of course $\pi \leftrightarrow \hat{\pi}^i \wedge \check{\pi}^i$. In the same vein, we shall note $\ddot{\pi}^i$ the conjunction of literals in π that are seen by agent i and $\dot{\pi}^i$ the conjunction of literals in π that are not seen by it. Again $\pi \leftrightarrow \ddot{\pi}^i \wedge \dot{\pi}^i$.

As its name suggests, a full description $(\pi \wedge \gamma \wedge \varsigma)$ fully characterises a situation: it specifies which atoms are true and which are false (this is π), it specifies which agents control which variables (through γ) and it specifies exactly which propositional variables each agent can see (through ς). So semantically, it is immediately clear that any formula will be a disjunction of such full descriptions (namely, descriptions of those situations where φ is true), but our task is now to show that this is derivable in the logic.

The next Lemma states a few theorems derivable within our axiomatic system, all of which are instrumental in the proofs of Theorem 2 and of Theorem 3.

LEMMA 2. *Let π, γ and ς be propositional, control and visibility descriptions, respectively (and so are their 'primed' version). For $P \subseteq \mathbb{A}$, let $\pi_1(L(P))$ be a conjunction over $L(P)$ and let $\pi_2(L(\mathbb{A} \setminus P))$ be a conjunction over $L(\mathbb{A} \setminus P)$.*

Then, the following are derivable in Λ_1:

1. $\neg ctrls(i,p) \to (\ell(p) \to \Box_i\ell(p))$
2. $sees(i,p) \to (\ell(p) \to K_i\ell(p))$
3. $\ell(ctrls(i,p)) \leftrightarrow \Box_N\ell(ctrls(i,p))$
4. $\ell(sees(i,p)) \leftrightarrow \Box_N\ell(sees(i,p))$
5. $\bigwedge_{p\in P} ctrls(i,p) \wedge \bigwedge_{p\notin P} \neg ctrls(i,p) \to \Diamond_i\pi_(L(P)) \wedge (\pi_2(L(\mathbb{A} \setminus P)) \to \Box_i\pi_2(L(\mathbb{A} \setminus P)))$
6. $\Diamond_i(\hat{\pi}^i \wedge \check{\pi}^i) \leftrightarrow \check{\pi}^i$
7. $\bigwedge_{p\in P} sees(i,p) \wedge \bigwedge_{p\notin P} \neg sees(i,p) \to M_i\pi_2(L(\mathbb{A} \setminus P)) \wedge (\pi_1(L(P)) \to K_i\pi_1(L(P)))$
8. $M_i(\ddot{\pi}^i \wedge \dot{\pi}^i) \leftrightarrow \ddot{\pi}^i$
9. $\Box_N\varphi \leftrightarrow \Box_i\Box_N\varphi$
10. $\Box_N\varphi \leftrightarrow K_i\Box_N\varphi$
11. $(\pi \wedge \gamma \wedge \varsigma) \leftrightarrow (\pi \wedge \Box_N\gamma \wedge \Box_N\varsigma)$

THEOREM 2 (NORMAL FORM). *Every formula φ is provably equivalent to a disjunction of full descriptions, i.e., for every φ there exists a k and π_j, γ_j and ς_j $(1 \leq j \leq k)$ such that*

$$\vdash \varphi \leftrightarrow \bigvee_{j\leq k} (\pi_j \wedge \Box_N\gamma_j \wedge \Box_N\varsigma_j) \qquad (1)$$

PROOF. By Lemma 2.11, it follows from

$$\vdash \varphi \leftrightarrow \bigvee_{j \leq k} (\pi_j \wedge \gamma_j \wedge \varsigma_j)$$

which we prove now by induction on the structure of φ.

We will make use of the fact that the sets of propositional (Π), control (Γ) and visibility (Σ) descriptions are finite. Roughly speaking, a triple (π, γ, ς) represents a state. The idea behind the normal form is, that a formula can be represented by a subset $X \subseteq \Pi \times \Gamma \times \Sigma$, which translates in the language as a (typically large) disjunction of formulas of the form $\pi \wedge \gamma \wedge \varsigma$.

One base case is for φ being a basic proposition in Φ.

$$\vdash p \leftrightarrow \bigvee_{\substack{\pi_i \in \Pi \\ \pi_i \vdash p}} \bigvee_{\gamma_j \in \Gamma} \bigvee_{\varsigma_k \in \Sigma} (\pi_i \wedge \gamma_j \wedge \varsigma_k)$$

The statement $\pi_i \vdash p$ means that p appears as a positive literal in π_i. The two other base cases $\varphi = ctrls(i, p)$ and $\varphi = sees(i, p)$ are analogous.

Now we suppose for induction that φ can be transformed into an equivalent formula $\bigvee_{j \leq k}(\pi_j \wedge \gamma_j \wedge \varsigma_j)$.

Case $\psi = \neg\varphi$: "ψ is represented by the complement of the states representing φ".

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} \bigvee_{\substack{(\pi,\gamma,\varsigma) \in \Pi \times \Gamma \times \Sigma \\ (\pi,\gamma,\varsigma) \neq (\pi_j, \gamma_j, \varsigma_j)}} (\pi \wedge \gamma \wedge \varsigma)$$

Case $\psi = \varphi_1 \vee \varphi_2$: since the normal form itself is a disjunction, this case is straightforward.

Case $\psi = \Diamond_i \varphi$: similar to $\psi = M_i \varphi$.

Case $\psi = M_i \varphi$: by induction hypothesis

$$\vdash \psi \leftrightarrow M_i \bigvee_{j \leq k} (\pi_j \wedge \gamma_j \wedge \varsigma_j)$$

By modal logic

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} M_i(\pi_j \wedge \gamma_j \wedge \varsigma_j)$$

By Lemma 2.10 and Lemma 2.11

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} M_i(\pi_j \wedge K_i \Box_N \gamma_j \wedge K_i \Box_N \varsigma_j)$$

By $S5(K)$

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} (M_i \pi_j \wedge K_i \Box_N \gamma_j \wedge K_i \Box_N \varsigma_j)$$

Applying our notation and Lemma 2.11 and Lemma 2.10

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} (M_i(\ddot{\pi}^i_j \wedge \dot{\pi}^i_j) \wedge \gamma_j \wedge \varsigma_j)$$

By Lemma 2.8

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} (\ddot{\pi}^i_j \wedge \gamma_j \wedge \varsigma_j)$$

Finally,

$$\vdash \psi \leftrightarrow \bigvee_{j \leq k} \bigvee_{\tilde{\pi}_j \in \Pi(\dot{\pi}^i_j)} ((\ddot{\pi}^i_j \wedge \tilde{\pi}_j) \wedge \gamma_j \wedge \varsigma_j)$$

where $\Pi(\dot{\pi}^i_j)$ is the set of propositional descriptions restricted to the set of atoms occurring in $\dot{\pi}^i_j$, that is, that are not seen by i. □

We require some subsidiary definitions. We begin by defining an alternative, possible worlds semantics for ECL-PC(PO). Given a frame F, a *Kripke model for* ECL-PC(PO) is a structure

$$K = \langle W, R^\Diamond_1, \ldots, R^\Diamond_n, R^K_1, \ldots, R^K_n, \pi \rangle$$

where $W = \Theta$ is a *set of worlds*, which correspond to possible valuations to $\mathbb{A}$, $R^\Diamond_i \subseteq W \times W$, and $R^K_i \subseteq W \times W$, where these latter relations are defined as:

$$R^\Diamond_i(w, w') \text{ iff } w \equiv_{\mathbb{A} \setminus \mathbb{A}_i} w', \text{ and } R^K_i(w, w') \text{ iff } w \equiv_{V_i} w'.$$

Finally, $\pi : W \to 2^{\mathbb{A}}$ gives the set of Boolean variables true at each world. The key clauses for $\models^k$ ('Kripke semantics) are then as follows:

$$\begin{array}{lll} K, w \models^k p & \text{iff} & p \in \pi(w) \qquad (p \in \mathbb{A}) \\ K, w \models^k \Diamond_i \varphi & \text{iff} & \exists w' \in W \text{ s.t. } R^\Diamond_i(w, w') \text{ and } K, w' \models^k \varphi \\ K, w \models^k K_i \varphi & \text{iff} & \forall w' \in W \text{ s.t. } R^K_i(w, w') \text{ and } K, w' \models^k \varphi \end{array}$$

LEMMA 3. *Let F, θ be an* ECL-PC(PO) *frame and associated valuation, let K, w be the corresponding Kripke model and world, and let φ be an arbitrary* ECL-PC(PO) *formula. Then:*

$$F, \theta \models^d \varphi \text{ iff } K, w \models^k \varphi.$$

We assume the standard definitions of maximally consistent sets and their existence via Lindenbaum's lemma (see, e.g., [4, p.196]). We proceed to construct a canonical model

$$\hat{K} = \langle \hat{W}, \hat{R}^\Diamond_1, \ldots, \hat{R}^\Diamond_n, \hat{R}^K_1, \ldots, \hat{R}^K_n, \hat{\pi} \rangle, \text{ where:}$$

- $\hat{W}$ is the set of all Λ_1 maximally consistent sets;
- $\hat{R}^\Diamond_i(w, w')$ iff $\varphi \in w'$ implies $\Diamond_i \varphi \in w$;
- $\hat{R}^K_i(w, w')$ iff $\varphi \in w'$ implies $M_i \varphi \in w$; and
- $\hat{\pi}(w) = \mathbb{A} \cap w$.

The following is a standard result for canonical models:

LEMMA 4 (TRUTH LEMMA.). *Let $\hat{K} = \langle \hat{W}, \hat{R}^\Diamond_1, \ldots, \hat{R}^\Diamond_n, \hat{R}^K_1, \ldots, \hat{R}^K_n, \hat{\pi} \rangle$ be a canonical model, $w \in \hat{W}$ be a world in $\hat{K}$, and φ be an arbitrary* ECL-PC(PO) *formula. Then:*

$$\hat{K}, w \models^k \varphi \text{ iff } \varphi \in w.$$

The truth lemma above gives rise to completeness wrt. a set of models, but it is not the kind of models we have associated with ECL-PC(PO). In the intended models, the modalities K_i and $\Diamond_i$ are defined with respect to valuations that are 'similar' with respect to the appropriate sets of atoms, while in the canonical model, those modal operators are defined as necessity operators with respect to a relation between maximal consistent sets that is defined in terms of membership of formulas in these sets. We now have to show that, in the canonical model, these two ways of looking at the modalities coincide. For this, our normal form Theorem 2 will be crucial.

But first we restrict ourselves to a generated submodel of $\hat{K}$. To be more precise, for the canonical model $\hat{K}$ just obtained, and $w \in \hat{W}$, let $\hat{K}_{\vec{w}}$ be the model generated by w in the following sense. Let $\hat{R}^\Diamond_N$ be $\hat{R}^\Diamond_1 \cup \cdots \cup \hat{R}^\Diamond_n$. Then, define $\hat{W}_{\vec{w}} = \{v \mid \hat{R}^\Diamond_N(w, v)\}$, and all relations $\hat{R}^\Diamond_{\vec{w}_i}$ and $\hat{R}^K_{\vec{w}}$ and valuation $\hat{\pi}_{\vec{w}}$ are the old relations and valuation restricted to the new set $\hat{W}_{\vec{w}}$. The following is a known result about generated submodels:

$$\forall \varphi \forall v \in \hat{W}_{\vec{w}} \;\; \hat{K}, v \models \varphi \text{ iff } \hat{K}_{\vec{w}}, v \models \varphi$$

THEOREM 3 ($\hat{K}_{\vec{w}}$ SIMULATES AN ECL-PC(PO) FRAME.). *Let $\hat{K}$ be as defined above, and take $w \in \hat{W}$. Consider the model $\hat{K}_{\vec{w}}$. Define, for every $i \in N$, the sets $\mathbb{A}_i = \{p \mid ctrls(i,p) \in w\}$, and $V_i = \{p \mid sees(i,p) \in w\}$. Then, in $\hat{K}_{\vec{w}}$, the accessibility relations satisfy the following properties:*

1. $\hat{R}^{\Diamond}_{\vec{w}_i}(v,v')$ *iff* $\pi_{\vec{w}}(v) \equiv_{\mathbb{A}\setminus\mathbb{A}_i} \pi_{\vec{w}}(v')$.
2. $\hat{R}^{K}_{\vec{w}_i}(v,v')$ *iff* $\pi_{\vec{w}}(v) \equiv_{V_i} \pi_{\vec{w}}(v')$.

PROOF. Consider the first item. Suppose $\hat{R}^{\Diamond}_{\vec{w}_i}(v,v')$, which means that for any φ, $\varphi \in v' \Rightarrow \Diamond_i\varphi \in v$. Take any $p \in \mathbb{A} \setminus \mathbb{A}_i$. We show that $p \in v$ iff $p \in v'$. Suppose $p \in v$. By definition of $\mathbb{A}_i$, we have $ctrls(i,p) \notin w$, and, since w is a maximal consistent set, $\neg ctrls(i,p) \in w$. By Lemma 2, item 4 (take $\ell(ctrls(i,p) = ctrls(i,p))$ we have $\Box_N \neg ctrls(i,p) \in w$, and, since v is $\hat{R}^{\Diamond}_N$-reachable from w, we have $\neg ctrls(i,p) \in v$. This gives $(\neg ctrls(i,p) \wedge p) \in v$, which, by Lemma 2, item 1 gives us $\Box_i p \in v$. Now for contradiction, if $p \notin v'$, we would have $\neg p \in v'$, and by definition, $\Diamond_i \neg p \in v$, which contradicts $\Box_i p \in v$. The reasoning for $p \notin v$ goes similar.

For the converse, suppose $\pi_{\vec{w}}(v) \equiv_{\mathbb{A}\setminus\mathbb{A}_i} \pi_{\vec{w}}(v')$, i.e., $v \cap (\mathbb{A} \setminus \mathbb{A}_i) = v' \cap (\mathbb{A} \setminus \mathbb{A}_i)$. Take an arbitrary $\varphi \in v'$, we have to show that $\Diamond_i\varphi \in v$. By Theorem 2, we know that φ is equivalent to a disjunction as specified in (1), and since v' is a maximal consistent set, there must be (uniquely) a propositional description π, a control description γ and a visibility description ς such that $(\pi \wedge \Box_N\gamma \wedge \Box_N\varsigma) \in v'$. Since v and v' are both reachable from the same generating world w, we have $(\Box_N\gamma \wedge \Box_N\varsigma) \in v$ and hence, by ($comp\cup$) and ($T(\Box)$)

$$(\Box_i\gamma \wedge \Box_i\varsigma) \in v \tag{2}$$

Let us decompose π into $\pi_1 \wedge \pi_2$, where π_1 uses all the atoms p from $\mathbb{A}_i$, and π_2 uses all the atoms from $\mathbb{A} \setminus \mathbb{A}_i$. By Lemma 2, item 5, we have

$$\Diamond_i\pi_1 \in v \tag{3}$$

Moreover $\pi \in v'$ implies that $\pi_2 \in v'$. Moreover by assumption $v \cap (\mathbb{A}\setminus\mathbb{A}_i) = v' \cap (\mathbb{A}\setminus\mathbb{A}_i)$. Hence, $\pi_2 \in v$. By Lemma 2, item 5, we then have that

$$\Box_i\pi_2 \in v \tag{4}$$

Collecting equations (2), (3) and (4), and using the modal validity $\vdash (\Box\alpha \wedge \Diamond\beta) \rightarrow \Diamond(\alpha \wedge \beta)$, we obtain $\Diamond_i(\pi_1 \wedge \pi_2 \wedge \gamma \wedge \varsigma) \in v$. By Lemma 2.11, we conclude $\Diamond_i(\pi_1 \wedge \pi_2 \wedge \Box_N\gamma \wedge \Box_N\varsigma) \in v$ which means that $\Diamond_i\varphi \in v$.

We now prove the second item. Suppose $\hat{R}^{K}_{\vec{w}_i}(v,v')$, which means that for any φ, $\varphi \in v' \Rightarrow M_i\varphi \in v$. Take any $p \in V_i$. We show that $p \in v$ iff $p \in v'$. Suppose $p \in v$. By definition of V_i, we have $sees(i,p) \in w$. By Lemma 2, item 4 we have $\Box_N sees(i,p) \in w$, and, since v is $\hat{R}^{\Diamond}_N$-reachable from w, we have $sees(i,p) \in v$. This gives $(sees(i,p) \wedge p) \in v$, which, by Lemma 2, item 2 gives us $K_i p \in v$. Now for contradiction, if $p \notin v'$, we would have $\neg p \in v'$, and by definition, $M_i\neg p \in v$, which contradicts $K_i p \in v$. The reasoning for $p \notin v$ goes similar.

For the converse, suppose $\pi_{\vec{w}}(v) \equiv_{V_i} \pi_{\vec{w}}(v')$, i.e., $v \cap (V_i) = v' \cap (V_i)$. Take an arbitrary $\varphi \in v'$, we have to show that $M_i\varphi \in v$. By Theorem 2, we know that φ is equivalent to a disjunction as specified in (1), and since v' is a maximal consistent set, there must be (uniquely) a propositional description π, a control description γ and a visibility description ς such that $(\pi \wedge \Box_N\gamma \wedge \Box_N\varsigma) \in v'$. Since v and v' are both reachable from the same generating world w, we have $(\Box_N\gamma \wedge \Box_N\varsigma) \in v$ and hence, by ($incl$)

$$(K_i\gamma \wedge K_i\varsigma) \in v \tag{5}$$

Let us decompose π into $\pi_1 \wedge \pi_2$, where π_1 uses all the atoms p from $\mathbb{A} \setminus V_i$, and π_2 uses all the atoms from V_i. By Lemma 2, item 7, we have

$$M_i\pi_1 \in v \tag{6}$$

Moreover $\pi \in v'$ means trivially that $\pi_2 \in v'$. Moreover by assumption $v \cap (V_i) = v' \cap (V_i)$. Hence, $\pi_2 \in v$. By Lemma 2, item 7, we then have that

$$K_i\pi_2 \in v \tag{7}$$

Collecting equations (5), (6) and (7), and using the modal validity $\vdash (\Box\alpha \wedge \Diamond\beta) \rightarrow \Diamond(\alpha \wedge \beta)$, we obtain $M_i(\pi_1 \wedge \pi_2 \wedge \gamma \wedge \varsigma) \in v$. By Lemma 2.11, we conclude $M_i(\pi_1 \wedge \pi_2 \wedge \Box_N\gamma \wedge \Box_N\varsigma) \in v$ which means that $M_i\varphi \in v$. □

THEOREM 4 (COMPLETENESS OF Λ_1.). *Λ_1 is sound and complete with respect to the class of* ECL-PC(PO) *frames.*

PROOF. Soundness is observed in Lemma 1. For completeness, take a Λ_1-consistent formula φ. Consider a maximal consistent set w with $\varphi \in w$. We know that $\hat{K}, w \models \varphi$. Take the generated model $\hat{K}_{\vec{w}}$. We know that again $\hat{K}_{\vec{w}}, w \models \varphi$, and moreover, by Theorem 3, $\hat{K}_{\vec{w}}$ simulates an ECL-PC(PO) frame. □

3. UNCERTAINTY ABOUT OWNERSHIP

The next type of uncertainty we consider relates to which agents control which variables. We refer to the logic we develop to capture such situations as the ECL-PC(UO), where "UO" stands for "uncertainty of ownership". The syntax of ECL-PC(UO) is identical to that of ECL-PC(PO), and so we will not present the syntax again here. In the semantics however, we substitute for every agent the set of propositions that it can see the value of, with a set of propositions which it sees the ownership of.

Given a set of agents N, atomic variables $\mathbb{A}$, and control partition $\mathbb{A}_1, \ldots, \mathbb{A}_n$, a *controls observation* for agent i is as set $\Omega_i \subseteq \mathbb{A}$. The interpretation of Ω_i is that $p \in \Omega_i$ means that agent i *knows who has control over the variable* p, that is, the agent $j \in N$ such that $p \in \mathbb{A}_j$. Given this, we define a frame F for ECL-PC(UO) as:

$$F = \langle N, \mathbb{A}_1, \ldots, \mathbb{A}_n, \Omega_1, \ldots, \Omega_n \rangle \text{ where:}$$

- N and $\mathbb{A}_i \subseteq \mathbb{A}$ are as before, and
- Ω_i is the controls observation for agent i.

We now define a relation on *frames*, which will be used to give a semantics to our epistemic modalities. Let

$$F = \langle N, \mathbb{A}_1, \ldots, \mathbb{A}_n, \Omega_1, \ldots, \Omega_i, \ldots, \Omega_n \rangle, \text{ and}$$

$$F' = \langle N, \mathbb{A}'_1, \ldots, \mathbb{A}'_n, \Omega'_1, \ldots, \Omega'_i, \ldots, \Omega'_n \rangle$$

be two frames that contain the same agents and the same base set of propositional variables. Then we write $F \simeq_i F'$ to mean that (1) $\Omega_i = \Omega'_i$ and (2) for all $p \in \Omega_i$ and for all $j \in N$ we have $\mathbb{A}_j \cap \Omega_i = \mathbb{A}'_j \cap \Omega_i$. Thus, roughly, $F \simeq_i F'$ means that F' and F' agree on the variables that i can see the ownership of, and moreover, for each of those variables, the control is assigned to the same agents in both frames.

Formally, the key steps in the semantics are defined as follows:

$$\begin{array}{lll} F, \theta \models^d p & \text{iff} & \theta(p) = true \qquad (p \in \mathbb{A}) \\ F, \theta \models^d \Diamond_i\varphi & \text{iff} & \exists \theta' \in \Theta : \theta' \equiv_{\mathbb{A}\setminus\mathbb{A}_i} \theta \text{ s.t. } M, \theta' \models^d \varphi \\ F, \theta \models^d K_i\varphi & \text{iff} & \forall F' : F' \simeq_i F \Longrightarrow F', \theta \models^d \varphi \end{array}$$

EXAMPLE 2. *Suppose we have a frame F in which $N = \{1, 2\}$, $\mathbb{A}_1 = \{p\}$, $\mathbb{A}_2 = \{q\}$, $\Omega_1 = \emptyset$, $\Omega_2 = \{p, q\}$. In this case, agent 1 has* no information at all *about which agent controls which variable: As far as this agent is concerned,* any *partition of controlled variables to agents is possible. Let $\theta(p) = \theta(q) = true$. We have:*

- $F, \theta \models^d K_1(p \wedge q) \wedge K_2(p \wedge q)$

  Unlike ECL-PC(PO)*, agents have no uncertainty about the actual value of variables. Thus both agents know that both variables are true in the valuation θ.*

- $F, \theta \models^d \Diamond_1(\neg p \wedge q) \wedge \neg K_1 \Diamond_1(\neg p \wedge q)$

  In fact, agent 1 can bring about $\neg p \wedge q$: he controls the variable p and he can choose $\neg p \wedge q$. However, because he is uncertain about whether he controls p, he does not know that he has the ability to choose $\neg p \wedge q$.

- $F, \theta \models^d \Diamond_2(p \wedge \neg q) \wedge K_2 \Diamond_2(p \wedge \neg q)$

  Agent 2 can choose a value for q so as to bring about $p \wedge \neg q$ (assuming agent 1 leaves p unchanged). Moreover, since 2 knows that she controls q, she knows that she can choose $p \wedge \neg q$.

- $F, \theta \models^d K_2((p \wedge q) \wedge \Diamond_1(\neg p \wedge \Diamond_2 \neg q))$

  Agent 2 knows that actually $p \wedge q$ holds, and that 1 can choose a situation where p is false and in which agent 2 furthermore can set q to false.

- $F, \theta \models^d K_1 \Box_{\{1,2\}} \Diamond_{\{1,2\}}(p \leftrightarrow \neg q) \wedge K_2 \Box_1 \Diamond_2(p \leftrightarrow \neg q)$

  Agent 1 knows that together, the agents can always make the values of p and q different, but agent 2 even knows that, no matter which values 1 chooses for his variables, 2 can achieve a situation such that p and q are different.

Note that, by the same arguments as given for ECL-PC(PO), we may conclude that:

THEOREM 5. *The model checking and satisfiability problems for* ECL-PC(UO) *are both PSPACE-complete.*

We give an axiomatization for ECL-PC(UO) in Figure 2. Derivability $\vdash$ in this section refers to that axiomatization. The following definitions and notations are useful.

DEFINITION 2. *Define $seeswho(i, p)$ as $\bigvee_{j \in N} K_i ctrls(j, p)$. Let $SW = \{seeswho(j, p) \mid j \in N,\ p \in \mathbb{A}\}$. The elements of $\mathbb{A}$, $CTRL$ and SW are our new basic propositions. A* controls observation description *ω is a full conjunction over SW. We note Ω the set of such controls observation descriptions. A new* full description *is a conjunction $\pi \wedge \gamma \wedge \omega$, where π, γ and ω are as explained above.*

Let $P \subseteq \mathbb{A}$. We define $CTRL(P) = \{\bigwedge ctrls(i, p) \mid i \in N, p \in P$, every p appears only once$\}$. Finally let $\hat{\omega}^i$ be of the form $\bigwedge_{p \in \mathbb{A}} \ell(seeswho(i, p))$ and let the formula $\check{\omega}^i$ be of the form $\bigwedge_{p \in At, j \neq i} \ell(seeswho(j, p))$ such that $\hat{\omega}^i \wedge \check{\omega}^i$ is a controls observation description.

As with ECL-PC(PO), a full description $(\pi \wedge \gamma \wedge \omega)$ fully characterises a situation: it specifies which atoms are true and which are false (this is π), it specifies which agents control which variables (through γ) and it specifies exactly which agent is aware of who owns which variables (through ω). So semantically, it is immediately clear that any formula will be a disjunction of such full descriptions (namely, descriptions of those situations where φ is true), but our task is now to show that this is derivable in the logic.

CLPC	φ	where φ is a CLPC tautology
Knowledge		
$(K(K))$	$K_i(\varphi \to \psi) \to (K_i\varphi \to K_i\psi)$	
$(T(K))$	$K_i\varphi \to \varphi$	
$(B(K))$	$\varphi \to K_i M_i \varphi$	
$(4(K))$	$K_i\varphi \to K_i K_i \varphi$	
Ax1	$\psi \to K_i\psi$	when ψ objective
Ax2	$\Diamond_N \psi \to K_i \Diamond_N \psi$	when ψ objective
Ax3	$\ell(seeswho(i, p)) \to K_i \ell(seeswho(i, p))$	
Ax4	$seeswho(i, p) \wedge \ell(ctrls(j, p)) \to K_i \ell(ctrls(j, p))$	
Ax5	$\bigwedge_{p \in P} \neg seeswho(i, p) \to M_i(\gamma \wedge \check{\omega}^i)$	
Ax6	$\bigwedge_{p \in P} seeswho(i, p) \to (\gamma \to K_i\gamma)$	
Ax7	$M_i \check{\omega}^i \wedge K_i \hat{\omega}^i$	
Rules		
(MP)	from $\vdash \varphi \to \psi$ and $\vdash \varphi$ infer $\vdash \psi$	
$(Nec(\Box))$	from $\vdash \varphi$ infer $\vdash \Box_i \varphi$	
$(Nec(K_i))$	from $\vdash \varphi$ infer $\vdash K_i\varphi$	

Figure 2: Axiomatics of Λ_2. The meta-variable i ranges over N, φ represents an arbitrary formula of ECL-PC(UO), p ranges over $\mathbb{A}$. Finally, $\hat{\omega}^i$, and $\check{\omega}^i$ are as specified in Definition 2, and $\gamma \in CTRL(P)$. Objective formulas have no modal operators.

LEMMA 5. *The axiomatization for Λ_2 in Figure 2 is sound.*

We now prove that this axiomatization is complete.

THEOREM 6 (NORMAL FORM). *Every formula φ is provably equivalent to a disjunction of full descriptions, i.e., for every φ there exists a k and π_j, γ_j and ω_j $(1 \leq j \leq k)$ such that*

$$\vdash \varphi \leftrightarrow \bigvee_{1 \leq j \leq k} \pi_j \wedge \gamma_j \wedge \omega_j$$

The proof of Theorem 6 is omitted for reasons of space. We now define an alternative, possible worlds semantics for ECL-PC(UO). Given a frame $F = \langle N, \mathbb{A}_1, \ldots, \mathbb{A}_n, \Omega_1, \ldots, \Omega_i, \ldots, \Omega_n \rangle$, a *corresponding pointed Kripke model for* ECL-PC(UO) is a structure

$$K, w_{(F,\theta)} = \langle W, R_1^{\Diamond}, \ldots, R_n^{\Diamond}, R_1^K, \ldots, R_n^K, \pi \rangle, w_{(F,\theta)}$$

where $W = \Pi \times \Gamma \times \Omega$ is a *set of worlds* that correspond to a frame and a propositional valuation. For every $w \in W$, we note $w(\pi)$ the propositional description it contains, $w(\gamma)$ the control description, and $w(\omega)$ the controls observation description. Given two states w and w', a set of propositions X, we have already defined $w(\pi) \equiv_X w'(\pi)$. We define $w(\gamma) \equiv_X^i w'(\gamma)$ to mean that for every $p \in X$, $w(\gamma) \vdash ctrls(i, p)$ iff $w'(\gamma) \vdash ctrls(i, p)$. Similarly, we define $w(\omega) \equiv_X^i w'(\omega)$ to mean that for every $p \in X$, $w(\omega) \vdash seeswho(i, p)$ iff $w'(\omega) \vdash seeswho(i, p)$. Finally, the world $w_{(\theta,F)}$ is such that $w_{(\theta,F)}(\pi)$ describes θ, $w_{(\theta,F)}(\gamma)$ describes $\mathbb{A}_1, \ldots \mathbb{A}_n$ and $w_{(\theta,F)}(\omega)$ describes $\Omega_1, \ldots, \Omega_n$.

The relations $R_i^{\Diamond} \subseteq W \times W$, and $R_i^K \subseteq W \times W$, are defined as follows:

$$R_i^{\Diamond}(w, w') \text{ iff } \begin{cases} w(\pi) \equiv_{\mathbb{A} \setminus \mathbb{A}_i} w'(\pi) \\ w(\omega) = w'(\omega) \\ w(\gamma) = w'(\gamma) \end{cases}$$

and

$$R_i^K(w, w') \text{ iff } \begin{cases} w(\pi) \equiv_{\mathbb{A}} w'(\pi) \\ w(\omega) \equiv_{\Omega_i}^i w'(\omega) \\ w(\gamma) \equiv_{\mathbb{A}_j \cap \Omega_i}^i w'(\gamma) \quad \text{for all } j \in N \end{cases}.$$

Finally, $\pi : W \to 2^{\mathbb{A}}$ gives the set of Boolean variables true at each world. We can then define a Kripke semantics for our language,

with the key clauses defined via the satisfiability relation $\models^k$ as follows:

$$\begin{array}{lll} K,w \models^k p & \text{iff} & p \in \pi(w) \qquad (p \in \mathbb{A}) \\ K,w \models^k \Diamond_i \varphi & \text{iff} & \exists w' \in W \text{ s.t. } R_i^{\Diamond}(w,w') \text{ and } K,w' \models^k \varphi \\ K,w \models^k K_i \varphi & \text{iff} & \forall w' \in W \text{ s.t. } R_i^{K}(w,w') \text{ and } K,w' \models^k \varphi \end{array}$$

The following is immediate.

LEMMA 6. *Let F,θ be an* ECL-PC(UO) *frame and associated valuation, let $K, w_{(\theta,F)}$ be the corresponding Kripke model and world, and let φ be an arbitrary* ECL-PC(UO) *formula. Then:*

$$F,\theta \models^d \varphi \text{ iff } K, w_{(\theta,F)} \models^k \varphi.$$

The definition of a canonical model $\hat{K}$ for the logic is as before (although the model of course will be different, since the axioms are different!), and the truth lemma holds for this language as well. But in this case, we do not need to restrict ourselves to a generated submodel.

THEOREM 7 ($\hat{K}$ SIMULATES AN ECL-PC(UO) FRAME.). *Let $\hat{K}$ be as defined as above. Define, for every $i \in N$ and $v \in \hat{W}$, the sets $\mathbb{A}_{v_i} = \{p \mid ctrls(i,p) \in v\}$, and $\Omega_{v_i} = \{p \mid \exists j \in N, K_i ctrls(j,p) \in v\}$. Then, in $\hat{K}$, the accessibility relations satisfy the following properties:*

1. $\hat{R}_i^{\Diamond}(v,v')$ *iff* $\begin{cases} \pi(v) \equiv_{\mathbb{A}\setminus\mathbb{A}_i} \pi(v') \\ v(\omega) = v'(\omega) \\ v(\gamma) = v'(\gamma) \end{cases}$

2. $\hat{R}_i^{K}(v,v')$ *iff* $\begin{cases} \pi(v) \equiv_{\mathbb{A}} \pi(v') \\ v(\omega) \equiv^i_{\Omega_{v_i'}} v'(\omega) \\ v(\gamma) \equiv^i_{\mathbb{A}_{v_j'} \cap \Omega_{v_i'}} v'(\gamma) \quad \textit{for all } j \in N \end{cases}$

PROOF. We prove the second item. Suppose that $\hat{R}_i^K(v,v')$. By definition, it means that for all φ, $\varphi \in v'$ implies $M_i\varphi \in v$. We now prove the three properties of the right side of the item. We first show that $p \in v$ iff $p \in v'$. Suppose that $p \in v'$. Then $K_i p \in v'$ by Ax1. By hypothesis we obtain $M_i K_i p \in v$, which by $S5$ yields $p \in v$. The case $p \notin v'$ is similar.

We now show that $K_i ctrls(j,p) \in v$ iff $K_i ctrls(j,p) \in v'$. First, suppose that $K_i ctrls(j,p) \in v$. Then by hypothesis we have $M_i K_i ctrls(j,p) \in v$ and $K_i ctrls(j,p) \in v$ by $S5$. Second, suppose that $K_i ctrls(j,p) \notin v'$. Since v' is a m.c. set, $\neg K_i ctrls(j,p) \in v'$. Then, $M_i M_i \neg ctrls(j,p) \in v$ which by $S5$ is equivalent to $M_i \neg ctrls(j,p) \in v$ and $\neg K_i ctrls(j,p) \in v$. And since v is a m.c. set, we have $K_i ctrls(j,p) \notin v$.

Now, take any $j \in N$ and any $p \in \mathbb{A}_{v_j'} \cap \Omega_{v_i'}$. We show that $ctrls(j,p) \in v$ iff $ctrls(j,p) \in v'$. First, suppose that $ctrls(j,p) \in v'$. By definition of $\Omega_{\vec{v'}_i}$, we have $K_i ctrls(j,p) \in v'$. By hypothesis, we have $M_i K_i ctrls(j,p) \in v$ which in $S5$ is equivalent to $ctrls(j,p) \in v$. Second, suppose that $ctrls(j,p) \notin v'$. Since v' is an m.c. set, $\neg ctrls(j,p) \in v'$. Also, by definition of $\Omega_{\vec{v'}_i}$, we have $seeswho(i,p) \in v'$. Hence, by Axiom Ax4 we have $K_i \neg ctrls(j,p) \in v'$. Hence, we have $M_i K_i \neg ctrls(j,p) \in v$ which in $S5$ is equivalent to $\neg ctrls(j,p) \in v$, and since v is a m.c. set we obtain $ctrls(j,p) \notin v$.

We now prove the right to left direction of item 2. To do so, suppose that (hyp1) $\pi(v) \equiv_{\mathbb{A}} \pi(v')$, (hyp2) $v(\omega) \equiv^i_{\Omega_{v_i'}} v'(\omega)$ and (hyp3) $v(\gamma) \equiv^i_{\mathbb{A}_{v_j'} \cap \Omega_{v_i'}} v'(\gamma)$ for all $j \in N$. We need to show that $\hat{R}_i^K(v,v')$, that is, for all φ we have $\varphi \in v'$ implies $M_i\varphi \in v$.

Take an arbitrary $\varphi \in v'$. By Theorem 6, we assume w.l.o.g. that for some k we have $\varphi \leftrightarrow \bigvee_{1 \le j \le k} (\pi_j \wedge \gamma_j \wedge \omega_j)$.

Since v' is an m.c. set, there is (uniquely) a full description $\pi \wedge \gamma \wedge \omega$ such that $(\pi \wedge \gamma \wedge \omega) \in v'$.

From (hyp1) we have $\pi \in v$ and by Ax1 we obtain

$$K_i \pi \in v \tag{8}$$

Let us write ω as $\omega_1 \wedge \omega_2$ such that ω_1 contains the $\ell(seeswho(i,p))$ literals (those concerning i's observations) and ω_2 contains all the other literals in ω. Since by (hyp2) we have $v(\omega) \equiv^i_{\Omega_{v_i'}} v'(\omega)$, we have $\omega_1 \in v$ and by Axiom Ax3 we get $K_i\omega_1 \in v$. Hence

$$K_i \omega_1 \in v \tag{9}$$

Let us now decompose γ into $\gamma_1 \wedge \gamma_2$ such that γ_1 contains all the $ctrls(j,p)$ appearing in γ such that $p \in \Omega_{v_i'}$ and ω_2 contains all the other control atoms appearing in γ.

From (hyp3) we know that for all $j \in N$ we have $v(\gamma) \equiv^i_{\mathbb{A}_{v_j'} \cap \Omega_{v_i'}} v'(\gamma)$. Then for all $p \in \mathbb{A}_{v_j'} \cap \Omega_{v_i'}$ and all $j \in N$, we have that $ctrls(j,p) \in v$ iff $ctrls(j,p) \in v'$.

Then we have $\gamma_1 \in v$ and by Axiom Ax6 we obtain

$$K_i \gamma_1 \in v \tag{10}$$

Finally, using Axiom Ax5 we obtain

$$M_i(\omega_2 \wedge \gamma_2) \in v \tag{11}$$

Combining (8), (9), (10), and (11) we then obtain $M_i(\pi \wedge \omega \wedge \gamma) \in v$, i.e., $M_i\varphi \in v$. □

THEOREM 8 (COMPLETENESS OF Λ_2.). *Λ_2 is sound and complete with respect to the class of* ECL-PC(UO) *frames.*

Let us finally sketch a general setup, in which:

1. not every atom $p \in \mathbb{A}$ needs to be in control of an agent;
2. agent i does not necessarily know what j sees (if $i \neq j$) and does not have complete ignorance either;
3. agent i does not necessarily know what j knows about control (if $i \neq j$) and does not have complete ignorance either.

To cater for this, let $\Upsilon_i = \langle \Omega_i, V_i \rangle$, where $\Omega_i \subseteq \mathbb{A}$ and $V_i \subseteq \mathbb{A}$. The idea is that for every atom in Ω_i, agent i knows who controls it, and for every atom in V_i, agent i knows what its truth value is. Now, a model M is of the form

$$M = \langle N, S, R^{\Delta}, \simeq \rangle, \text{ where}$$

1. S is a set of states $\langle \mathbb{A}_1, \ldots, \mathbb{A}_n, \Upsilon_1, \ldots, \Upsilon_n, \theta \rangle$;
   (a) $\cup_{i \in N} \mathbb{A}_i \subseteq \mathbb{A}$ and $\mathbb{A}_i \cap \mathbb{A}_j \neq \emptyset$
   (b) $\Upsilon_i = \langle \Omega_i, V_i \rangle$ with $\Omega_i, V_i \subseteq \mathbb{A}$
2. $R^{\Delta} : N \to S \times S$ is a binary relation. This relation satisfies the following: for every $\langle \mathbb{A}_1, \ldots, \mathbb{A}_n, \Upsilon_1, \ldots, \Upsilon_n, \theta \rangle \in S$, and every θ' such that $\theta \equiv_{\mathbb{A}\setminus\mathbb{A}_{i_s}} \theta'$, there is a state $t = \langle \mathbb{A}_1, \ldots, \mathbb{A}_n, \Upsilon_1, \ldots, \Upsilon_n, \theta' \rangle$;
3. Given two states $s = \langle \mathbb{A}_1, \ldots, \mathbb{A}_n, \Upsilon_1, \ldots, \Upsilon_n, \theta \rangle$ and $s' = \langle \mathbb{A}'_1, \ldots, \mathbb{A}'_n, \Upsilon'_1, \ldots, \Upsilon'_n, \theta' \rangle$, define

$$s \simeq_i s' \text{ iff } \begin{cases} \Upsilon_i = \Upsilon'_i \\ \forall p \in V_i\, \theta(p) = \theta'(p) \\ \forall p \in \Omega_i \forall j \in N\ (p \in \mathbb{A}_j \text{ iff } p \in \mathbb{A}'_j) \end{cases}$$

The semantics is very general and allows for a number of specialisations. Examples of such specialisations are:

1. For all states s and every agent i, $\Omega_{i_s} = \mathbb{A}$ (complete knowledge about control)
2. For all states s and t, and every agent i, the components Ω_{i_s} and Ω_{i_t} are the same.
3. For all states s and t, and every agent i, the components V_{i_s} and V_{i_t} are the same.

These properties entail some validities:

1. $\models ctrls(j,p) \leftrightarrow K_i\, ctrls(j,p)$
2. $\models K_i\, ctrls(j,p) \leftrightarrow (K_h K_i\, ctrls(j,p) \wedge \Box_N K_i\, ctrls(j,p))$
3. $\models sees(i,p) \leftrightarrow (K_h\, sees(i,p) \wedge \Box_N\, sees(i,p))$

In fact, all those specialisations apply to ECL-PC(PO). Other natural assumptions would be that for instance $\mathbb{A}_i \subseteq \Omega_i$ (corresponding to $ctrls(i,p) \rightarrow K_i\, ctrls(i,p)$) and $\mathbb{A}_i \subseteq V_i$ (corresponding to $ctrls(i,p) \rightarrow sees(i,p)$).

We give one simple scenario that can be modelled in this setup, that of *Voting*. All agents either desire something (p_i) or not. They can reveal their preference through q_i: if $p_i \leftrightarrow q_i$, agent i is truthful, otherwise it lies. Here, $\mathbb{A}_i = \{q_i\}$, $\Omega_i = \{q_j \mid j \in N\}$ and $V_i = \{p_i\} \cup \{q_j \mid j \in N\}$. In other words, we assume agents cannot control what they prefer, although what they can do is choose their vote. We have here

$$\ell(p_i) \rightarrow K_i(\Diamond_i(\ell(p_i) \wedge q_i) \wedge \Diamond_i(\ell(p_i) \wedge \neg q_i))$$

i.e., i knows that it can vote truthfully but it can also lie. We also get $K_i q_j \rightarrow \neg(K_i p_j \vee K_i \neg p_j)$: even if i knows j's vote, it does not know j's real preference. Note that the information about what agents see and what they know about controls is still *global*, we have e.g. $K_i K_j\, ctrls(h, q_h)$.

4. CONCLUSION

As noted before, we added an information component to the logic of propositional control CL-PC ([14]). From a technical perspective, like in [7], our logic ECL-PC(PO), even if we would require that all agents see all propositional variables, is an extension of CL-PC, since as presented in [14], the distribution of propositional variables $\mathbb{A}$ over agents is assumed as *given*. In ECL-PC(PO), it is not given, but it is *fixed*, implying that a specification φ may leave room for different distributions of the atoms, but once it is chosen, there is no way to refer to other distributions, not in terms of what agents can imagine, nor in terms of what they can achieve.

There are close connections between propositional logics of control and other logics that facilitate reasoning about the powers of coalitions, like Coalition Logic [11] and ATL [2]. In fact, CL-PC was partially motivated by the way the model checking system MOCHA for ATL [3] is designed, in which the system is divided in a number of modules (agents, in our terminology), each controlling its own set of Boolean variables. And indeed, there have been several attempts to add an epistemic component to ATL [13, 8, 1]. However, what those extensions all have in common is that the uncertainty of the agents is specified in an abstract way: in the Kripke models for the logics for cooperation and knowledge, the accessibility relations corresponding to knowledge are just given, abstract, equivalence relations. In our logic CL-PC(PO) the knowledge is determined by the variables of which the agent can see the truth value, and in ECL-PC(UO) this accessibility relation is determined by the variable of which the agent can see the ownership. In this sense, we provide a *computationally grounded semantics* [16] for knowledge, which brings our approach closer to the interpreted systems approach to epistemic logic [5, 6]. Interestingly enough, the key idea of interpreted systems (two states are the same for agent i if the atoms that it sees have the same value) does not only apply to the epistemic dimension in our logics, but also to the control dimension: two states are reachable in terms of i's control, if the values of the atoms not in i's control is the same.

Future work should study how to combine our two approaches, as suggested at the end of Section 3, and to weaken some of the underlying assumptions regarding the agents' knowledge. Related to this, we would like to provide a completeness proof for our systems that does not rely on a normal form (and on the assumption that the number of propositional atoms is finite). Doing this, one needs to find a way of juggling with the two types of definitions of 'access' we are dealing with here: on the one hand, the canonical model in modal logic defines this in terms of membership of formulas in the states, whereas the interpreted systems approach would to this in terms of 'similarity' of the states. We hope that work of Lomuscio [9], connecting general $S5$ semantics with that of interpreted systems, may give some first steps in this search. Another natural direction to be explored is to emphasize the group aspect of both dimensions: when forming a coalition C to bring about φ, i.e., $\Diamond_C \varphi$ gives rise to interesting questions from cooperative game theory, and epistemic logic provides the tools and results to combine this with interesting notions of *group knowledge*.

5. REFERENCES

[1] T. Ågotnes. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):377–409, 2006.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *JACM*, 49(5):672–713, 2002.

[3] R. Alur *et al*. Mocha: Modularity in model checking. In *CAV*, LNCS 1427, pp. 521–525, 1998.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[5] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. In M. Y. Vardi, (ed.), *TARK*, 1988.

[6] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[7] J. Gerbrandy. Logics of propositional control. In *AAMAS06*, pp. 193–200, 2006.

[8] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[9] A. Lomuscio. *Knowledge Sharing among Ideal Agents*. PhD thesis, University of Birmingham, UK, 1999.

[10] R. C. Moore. Reasoning about knowledge and action. In *IJCAI-77*, Cambridge, MA, 1977.

[11] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.

[12] W. van der Hoek, D. Walther, and M. Wooldridge. Reasoning about the transfer of control. *JAIR*, 37:437–477, 2010.

[13] W. van der Hoek and M. Wooldridge. Time, knowledge, and cooperation. *Studia Logica*, 75(1):125–157, 2003.

[14] W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *AI*, 164:81–119, 2005.

[15] Y. Venema. A modal logic of substitution and quantification. Logic Colloquium '92, CSLI Pub., pp 293–309, 1996.

[16] M. Wooldridge and A. Lomuscio. A computationally grounded logic of visibility, perception, and knowledge. *IGPL*, 9(2):273–288, 2001.

Strategic Games and Truly Playable Effectivity Functions

Valentin Goranko
Informatics and Mathematical Modelling, Technical University of Denmark
vfgo@imm.dtu.dk

Wojciech Jamroga
Computer Science and Communication, University of Luxembourg
wojtek.jamroga@uni.lu

Paolo Turrini
Information and Computing Sciences, Utrecht University
paolo@cs.uu.nl

ABSTRACT

A well known (and often used) result by Marc Pauly states that for every playable effectivity function E there exists a strategic game that assigns to coalitions exactly the same power as E, and vice versa. While the latter direction of the correspondence is correct, we show that the former does not always hold in the case of infinite game models. We point out where the proof of correspondence goes wrong, and we present examples of playable effectivity functions in infinite models for which no equivalent strategic game exists. Then, we characterize the class of *truly playable* effectivity functions, that does correspond to strategic games. Moreover, we discuss a construction that transforms any playable effectivity function into a truly playable one while preserving the power of most (but not all) coalitions. We also show that Coalition Logic is not expressive enough to distinguish between playable and truly playable effectivity functions, and we extend it to a logic that can make this distinction while enjoying finite axiomatization and finite model property.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Theory

Keywords

Strategic reasoning, cooperative games, correspondence

1. INTRODUCTION

Several logics for reasoning about coalitional power have been proposed and studied in the last two decades. Eminent examples are: Alternating-time Temporal Logic (ATL) [1], Coalition Logic (CL) [11], and Seeing To It That (STIT) [2], used in computer science and philosophy to reason about properties of multi-agent systems. A crucial feature of these logics is the correspondence between their models and the game structures they are meant to reason about.

In particular, the connection between the semantics of Coalition Logic and games relies on *Pauly's representation theorem* [11] which states that playable effectivity functions correspond exactly to strategic games. Moreover, the correspondence has been used to obtain further results for CL: if the semantics can be defined equivalently in terms of strategic games and playable effectivity functions, they can be used interchangeably when proving properties of the logic. A similar remark applies to ATL and STIT, connected to Coalition Logic by a number of simulation results [4, 6, 7].

The correspondence between strategic games and effectivity functions is important even without the logical context. Effectivity functions generalize basic models of cooperative game theory, whereas strategic games are models of non-cooperative game theory. Pauly's result is relevant as it puts forward a characterization of *strategic games in terms of coalitional games*, therefore establishing a connection between the two families of game models.

In this paper, we show that the representation theorem is not correct as it stands. More precisely, we show that there are some playable effectivity functions with no corresponding strategic games. We point out where Pauly's proof of correspondence goes wrong, and we present examples of playable effectivity functions , for which no equivalent strategic games exist. Then, we define a more restricted class of effectivity functions, that we call *truly playable*, and we show that they correspond precisely to strategic games. We discuss several alternative characterizations of truly playable functions. Moreover, we present a construction that recovers the correspondence in the sense that it transforms any playable function into a truly playable one while preserving the power of most (but not all) coalitions. Finally, we discuss the ramifications for the above mentioned logics. On the one hand we show that the complete axiomatization of Coalition Logic from [11] is not affected if we change the class of models from playable to truly playable. On the other hand, we propose more expressive languages that can characterize the property of true playability, thus drawing a logical distinction with Pauly's playability.

2. PRELIMINARIES

2.1 Strategic Games

Strategic games are basic models of non-cooperative game theory [10]. After [11], we focus on abstract game forms, where the effect of strategic interaction between players is

Cite as: Strategic Games and Truly Playable Effectivity Functions, Goranko, Jamroga, Turrini, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 727-734.

represented by abstract outcomes from a given set and players' preferences are not specified. For simplicity we refer to them as strategic games.

DEFINITION 1 (STRATEGIC GAME). *A* strategic game G *is a tuple* $(N, \{\Sigma_i | i \in N\}, o, S)$ *that consists of a nonempty finite set of players* N*, a nonempty set of strategies* Σ_i *for each player* $i \in N$*, a nonempty set of outcomes* S*, and an outcome function* $o : \prod_{i \in N} \Sigma_i \to S$ *which associates an outcome with every strategy profile.*

Additionally, we follow [11] and define coalitional strategies σ_C in G as tuples of individual strategies σ_i for $i \in C$, i.e., $\Sigma_C = \prod_{i \in C} \Sigma_i$. Note that this definition allows for only one strategy $\sigma_\emptyset$ when $C = \emptyset$, namely the empty function.

2.2 Effectivity Functions

Effectivity functions have been introduced in cooperative game theory [9] to provide an abstract representation of the powers of coalitions to influence the outcome of the game.

DEFINITION 2 (EFFECTIVITY FUNCTION). *An* effectivity function *is a function* $E : 2^N \to 2^{2^S}$*, that associates a family of sets of states from* S *with each set of players.*

Intuitively, elements of $E(C)$ are *choices* available to coalition C: if $X \in E(C)$ then by choosing X the coalition C can force the outcome of the game to be in X. Effectivity functions are usually required to satisfy additional properties, consistent with this interpretation.

DEFINITION 3 (PLAYABILITY [11]). *An effectivity function* E *is* playable *iff the following conditions hold:*

Outcome Monotonicity $X \in E(C)$ *and* $X \subseteq Y$ *implies* $Y \in E(C)$*;*

N-maximality $\overline{X} \notin E(\emptyset)$ *implies* $X \in E(N)$*;*

Liveness $\emptyset \notin E(C)$*;*

Safety $S \in E(C)$*;*

Superadditivity *if* $C \cap D = \emptyset$*,* $X \in E(C)$ *and* $Y \in E(D)$*, then* $X \cap Y \in E(C \cup D)$*.*

Looking at playable effectivity functions, we can observe that their representation contains some redundancy. In particular, the fact that $E(C)$ is outcome monotonic suggests that one could succinctly represent it in terms of minimal sets, i.e., the elements of $E(C)$ that form an antichain under set inclusion. The nonmonotonic core, introduced in [11], is aimed at providing such a representation.

DEFINITION 4 (NONMONOTONIC CORE). *Let* E *be an effectivity function. The* nonmonotonic core $E^{nc}(C)$ *for* $C \subseteq N$ *is the set of minimal sets in* $E(C)$*:*

$$E^{nc}(C) = \{X \in E(C) \mid \neg\exists Y (Y \in E(C) \text{ and } Y \subsetneq X)\}.$$

We will show in Section 3.1 that not all sets in an effectivity function need to contain a set from the nonmonotonic core. Thus, E^{nc} does not always behave well as a representation of the effectivity function, unless it is "complete" in the following sense.

DEFINITION 5 (COMPLETE NONMONOTONIC CORE). *The nonmonotonic core* $E^{nc}(C)$ *is* complete *iff for every* $X \in E(C)$ *there exists* $Y \in E^{nc}(C)$ *such that* $Y \subseteq X$*.*

The nonmonotonic core of the empty coalition is of particular interest to us. For it, the following holds.

PROPOSITION 1. *For every playable effectivity function* E*:*

1. $E(\emptyset)$ *is a filter.*
2. $E^{nc}(\emptyset)$ *is either empty or a singleton.*

PROOF. (1) $E(\emptyset)$ is non-empty by Safety; closed under supersets by Outcome Monotonicity, and under intersections by Superadditivity (with respect to the empty coalition).
(2) Suppose $E^{nc}(\emptyset)$ is non-empty, and let $X, Y \in E^{nc}(\emptyset)$. Then, coalition $\emptyset$ is effective for each of X and Y, hence, by superadditivity, it is effective for $X \cap Y$. By the definition of $E^{nc}(\emptyset)$, it follows that $X = X \cap Y = Y$. □

Each strategic game $\mathcal{G}$ can be canonically associated with an effectivity function, called the α-effectivity function of $\mathcal{G}$ and denoted with E^α_G.

DEFINITION 6 (α-EFFECTIVITY IN STRATEGIC GAMES). *For a strategic game* G *the* α-effectivity function $E^\alpha_G : 2^N \to 2^{2^S}$ is defined as follows: $X \in E^\alpha_G(C)$ if and only if there exists σ_C such that for all $\sigma_{\overline{C}}$ we have $o(\sigma_C, \sigma_{\overline{C}}) \in X$.

PROPOSITION 2. *For every* α*-effectivity function* $E^\alpha_G : 2^N \to 2^{2^S}$*, the following hold:*

1. *The nonmonotonic core of* $E^\alpha_G(\emptyset)$ *is the singleton set* $\{Z\}$ *where* $Z = \{\{x\} \in S \mid x = o(\sigma_N)$ *for some* $\sigma_N\}$*.*
2. $E^\alpha_G(\emptyset)$ *is the principal*[1] *filter generated by* Z*.*

PROOF. For both claims it suffices to observe that $Z \in E^\alpha_G(\emptyset)$ and that $Z \subseteq U$ for every $U \in E^\alpha_G(\emptyset)$. Therefore, $E^{nc}(\emptyset) = \{Z\}$ for $E = E^\alpha_G$ and $E^\alpha_G(\emptyset)$ is the principal filter generated by Z. □

3. PROBLEM WITH CORRESPONDENCE

In this section we show that playability is not sufficient to make effectivity functions correspond to strategic games.

3.1 Counterexample to Pauly's Representation Theorem

THEOREM 3 (PAULY'S REPRESENTATION THEOREM [11]). *A coalitional effectivity function* E α*-corresponds to a strategic game if and only if* E *is playable.*

Thus, the theorem states that every playable effectivity function is equal to the α-effectivity function of some game (Pauly calls this equivalence relation α*-correspondence*), and that each game has an α-effectivity function that is playable. While the latter is true, the former turns out incorrect.

PROPOSITION 4. *There is a playable effectivity function* E *for which* $E \neq E^\alpha_G$ *for all strategic games* G*.*

PROOF. Consider an effectivity function E ranging on a set N consisting of a single player a and on the set of natural numbers $\mathbb{N}$ (i.e., $N = \{a\}, S = \mathbb{N}$), and defined as follows:

- $E(\{a\}) = \{X \subseteq \mathbb{N} \mid X \text{ is infinite}\}$;

[1]Filter F on domain Ω is *principal* iff there exists $X \subseteq \Omega$ such that F is the set of all supersets of X. Then, F is said to be *generated* by X.

- $E(\emptyset) = \{X \subseteq \mathbb{N} \mid \overline{X} \text{ is finite}\}$.

We claim that E is playable and that it does not correspond to any strategic game. Let us first verify the playability conditions. Outcome monotonicity, N-maximality, liveness and safety are straightforward to check. For superadditivity, notice that we have only two cases to verify:

1. $C = \{a\}, D = \emptyset$. Superadditivity holds here because intersection of an infinite and a cofinite set is infinite.
2. $C = \emptyset, D = \emptyset$. Superadditivity in this case holds because intersection of two cofinite sets is cofinite.

On the other hand, $E^{nc}(\emptyset) = \emptyset$ because there are no minimal cofinite sets. This implies, by Proposition 2, that $E \neq E^{\alpha}_{G}$ for all strategic games G. □

3.2 Tracing the Problem

Below, we summarize the relevant part of the proof of Theorem 2.27 from [11], and show where it goes wrong. We outline the construction of a strategic game $\mathcal{G}$ given an effectivity function E (Steps 1–4); then, the argument supposed to show that E α-corresponds to $\mathcal{G}$ (Steps 5–6).

Step 1: the players and the domain remain the same. The game $\mathcal{G} = (N, S, \Sigma_i, o)$ inherits the set of outcomes and the set of players as in the effectivity function E.

Step 2: coalitions choose a set from their effectivity function. Now, a family of functions is defined:

$$F_i = \{f_i : \mathcal{C}_i \to 2^S \mid \text{ for all } C \text{ we have that } \ f_i(C) \in E(C)\}$$

where $\mathcal{C}_i = \{C \subseteq N \mid i \in C\}$. Each function f_i assigns choices to all coalitions of which i is a member. F_i simply collects all such assignments.

Step 3: coalitions are partitioned according to their choices. Let $f = (f_i)_{i \in N}, f_i \in F_i$, be a tuple of such assignments, one per player. The next step is to define the set $P_\infty(f)$ which results from iterative partitioning of the set of players in the coarsest possible way such that players in the same partition are assigned same coalitional choices:

$$\begin{aligned}
P_0(f) &= \langle N \rangle \\
P_1(f) &= P(f, N) = \langle C^1_1, \ldots, C^1_{k_1} \rangle \\
P_2(f) &= \langle P(f, C^1_1), \ldots, P(f, C^1_{k_1}) \rangle = \langle C^2_2, \ldots, C^2_{k_2} \rangle \\
&\ldots \\
P_\infty(f) &= P_r(f) \text{ such that } P_i(f) = P_{i+1}(f) \text{ for all } i \geq r,
\end{aligned}$$

where each $P(f, C)$ returns the coarsest partitioning $\langle C_1, \ldots, C_m \rangle$ of coalition C such that for all $l \leq m$ and for all $i, j \in C_l$ it holds that $f_i(C) = f_j(C)$.

Step 4: an outcome is chosen in the intersection of coalitional choices. Strategies and outcome function are defined as follows. Each player in N is given a set of strategies of the form (f_i, t_i, h_i) where $f_i \in F_i$ is an assignment of coalitional choices for player i (see Step 2), t_i is a player (possibly different from i), and $h_i : 2^S \setminus \emptyset \to S$ is a *selector* function that picks up an arbitrary element from each nonempty subset of S.

The outcome of strategy σ_N is now defined as:

$$o(\sigma_N) = h_{i_0}(\mathcal{G}(f)), \quad \mathcal{G}(f) = \bigcap_{l=1}^{k} f(C_l),$$

where i_0 is a uniquely chosen player, h_{i_0} is the outcome selector from i_0's strategy, and C_l is one of the k coalitions in $P_\infty(f)$.

This concludes the construction of a game $\mathcal{G}$ which should α-correspond to the effectivity function E. Steps 5–6 are supposed to prove that $E = E^{\alpha}_{\mathcal{G}}$.

Step 5: choices are not removed by the construction. First, an attempt to prove $E(C) \subseteq E^{\alpha}_{\mathcal{G}}(C)$ for arbitrary coalition C is presented [11, p.29]:

> For the inclusion from left to right, assume that $X \in E(C)$. Choose any C-strategy $\sigma_C = (f_i, t_i, h_i)_{i \in C}$ such that for all $i \in C$ and for all $C' \supseteq C$ we have $f_i(C') = X$.(*) By coalition monotonicity, such f_i exists.(**) Take now any $\overline{C}$-strategy, $\sigma_{\overline{C}} = (f_i, t_i, h_i)_{i \in \overline{C}}$. We need to show that $o(\sigma_C, \sigma_{\overline{C}}) \in X$. To see this, note that C must be a subset of one of the partitions C_l in $P_\infty(f)$. Hence, $o(\sigma_N) = h_{i_0}(\mathcal{G}(f)) = h_{i_0} \bigcap_{l=1}^{k} f(C_l) \in X$.

The deduction of the last sentence is where the proof goes wrong. The problem is that, for $C = \emptyset$, the only available strategy is the empty strategy $\sigma_\emptyset$ which vacuously satisfies condition (*). And, for any agent i, a choice assignment f_i satisfying the condition must exist. However, *there is no guarantee that any i will indeed choose f_i in its strategy* since the coalition C for which we can fix its strategy does not include any players. In consequence, one cannot deduce that $h_{i_0}\big(\bigcap_{l=1}^{k} f(C_l)\big) \in X$; this could be only concluded if the intersection contains at least one player whose choice $f_i(C_l)$ is X (or a subset of X).

Another case where the reasoning fails is $C = N$. Consider a state space S with $x \in S$, and an effectivity function E such that $\{x\} \notin E(N)$. Now, let strategy profile σ_N consist of $\sigma_i = (f_i, t_i, h_i)$ where everybody assumes choosing the whole state space in all circumstances (i.e., $f_i(C) = S$ for all i and C) and applies the same selector h_i such that $h_i(S) = x$. Now we get that $o(\sigma_N) = x$, so $\{x\} \in E^{\alpha}_{\mathcal{G}}(N)$, and hence $E(N) \neq E^{\alpha}_{\mathcal{G}}(N)$.

Step 6: choices are not added by the construction. The proof of the other direction ($E^{\alpha}_{\mathcal{G}}(C) \subseteq E(C)$) fails too, because in order to establish the inclusion for $C = N$, it is reduced to inclusion in step 5 for $C = \emptyset$, and we have just shown that it does not necessarily hold.

This concludes our analysis of the proof of Pauly's representation theorem in [11]. We consider it important for two reasons. First, we have identified precisely what was wrong with the construction of the proof. Second, we will reuse the sound parts of the original construction when proving a revised version of the correspondence in Section 4.2 and to obtain some additional results in Section 4.4.

4. TRULY PLAYABLE EFFECTIVITY FUNCTIONS

In this section we introduce an additional constraint on playable effectivity functions, that will enable us to restore the correspondence with strategic games in in Section 4.2.

4.1 Characterizing True Playability

DEFINITION 7. *An effectivity function E is* truly playable *iff it is playable and $E(\emptyset)$ has a complete nonmonotonic core.*

Several equivalent characterizations of truly playable effectivity functions are given in Proposition 5. For one of them, we will need the additional notion of a *crown*. Intuitively, an effectivity function is a crown if every choice of the agents in the grand coalition includes at least one state that the grand coalition can enforce precisely.

DEFINITION 8. *An effectivity function E is a* crown *iff $X \in E(N)$ implies $\{x\} \in E(N)$ for some $x \in X$.*

PROPOSITION 5. *The following are equivalent for every playable effectivity function $E : 2^N \to 2^{2^S}$.*

1. *E is truly playable.*
2. *$E(\emptyset)$ has a non-empty nonmonotonic core.*
3. *$E^{nc}(\emptyset)$ is a singleton and $E(\emptyset)$ is a principal filter, generated by $E^{nc}(\emptyset)$.*
4. *E is a crown.*

PROOF. (1) $\Rightarrow$ (2): immediate, by safety.

(2) $\Rightarrow$ (3): Let $Z \in E^{nc}(\emptyset)$ and let $X \in E(\emptyset)$. Then, by superadditivity, $Z \cap X \in E(\emptyset)$, and $Z \cap X \subseteq Z$, hence $Z \cap X = Z$ by definition of $E^{nc}(\emptyset)$. Thus, $Z \subseteq X$. So, $E(\emptyset)$ is the principal filter generated by Z, hence $E^{nc}(\emptyset) = \{Z\}$.

(3) $\Rightarrow$ (1): immediate from the definitions.

(3) $\Rightarrow$ (4): Let $E^{nc}(\emptyset) = \{Z\}$ and suppose $\{x\} \notin E(N)$ for all $x \in X$ for some $X \in E(\emptyset)$. Then, by N-maximality, $S \setminus \{x\} \in E(\emptyset)$, i.e. $Z \subseteq S \setminus \{x\}$ for every $x \in X$. Then $Z \subseteq S \setminus X$, hence $S \setminus X \in E(\emptyset)$. Therefore, $X \notin E(N)$ by superaditivity and liveness. By contraposition, E is a crown.

(4) $\Rightarrow$ (3): Let $Z = \{z \mid \{z\} \in E(N)\}$ and let $X \in E(\emptyset)$. Take any $z \in Z$, which is nonempty by liveness and the fact that E is a crown. By superadditivity we obtain that $\{z\} \cap X \in E(\emptyset)$, hence $z \in X$ by liveness. Thus, $Z \subseteq X$. Moreover, $Z \in E(\emptyset)$, for else $S \setminus Z \in E(N)$ by N-maximality, hence $\{x\} \in E(N)$ for some $x \in S \setminus Z$, which contradicts the definition of Z. Therefore, $E(\emptyset)$ is the principal filter generated by Z, hence $E^{nc}(\emptyset) = \{Z\}$. □

COROLLARY 6. *Every playable effectivity function $E : 2^N \to 2^{2^S}$ on a finite domain S is truly playable.*

PROOF. Straightforward, by Proposition 5.3 and the fact that every filter on a finite set is principal. □

4.2 Truly Playable Functions Correspond to Strategic Games

The proof of Theorem 2.27 from [11] fails when we consider the effectivity function of the empty coalition or the grand coalition. However the proof *is* correct for the other cases. We will now show that the additional condition of true playability yields correctness of the original construction from [11].

THEOREM 7. *A coalitional effectivity function E α-corresponds to a strategic game if and only if E is truly playable.*

PROOF. By Propositions 2 and 5, for any strategic game $\mathcal{G}$ its α-effectivity function $E^\alpha_\mathcal{G}$ is truly playable.

For the other direction, given a truly playable effectivity function E, we slightly change Pauly's procedure outlined in Section 3.2 (Steps 1–4). We impose an additional constraint on players' strategies $\sigma_i = (f_i, t_i, h_i)$, namely, we require that $h_i(X) = x$ for some $\{x\} \in E(N)$. In other words, the selector functions only select the "jewels" in the crown. Note that for $C \notin \{\emptyset, N\}$ the new procedure yields game $\mathcal{G}'$ with exactly the same $E^\alpha(C)$ as the original construction $\mathcal{G}$ from [11] (we omit the proof due to lack of space). It remains now to show that the procedure constructs a strategic game $\mathcal{G}$ such that $E(C) = E^\alpha_\mathcal{G}(C)$ for all $C \subseteq N$, that is, to show that steps 5 and 6 work well in truly playable structures.

Ad. Step 5. We show that $E(C) \subseteq E^\alpha_\mathcal{G}(C)$ for $C = \emptyset$ and $C = N$, the only cases in which the original proof failed.

Assume that $X \in E(\emptyset)$. We need to prove that $X \in E^\alpha_\mathcal{G}(\emptyset)$. By true playability we know that there exists $Y \in E^{nc}(\emptyset)$ such that $Y \subseteq X$. By Proposition 5, $E^{nc}(\emptyset) = \{Y\}$ and $E(\emptyset) = \{Z \mid Y \subseteq Z\}$. We will show now that $Y = \{x \mid \{x\} \in E(N)\}$ (*). First, suppose that $x \in Y$ and $\{x\} \notin E(N)$, then by N-maximality $S \setminus \{x\} \in E(\emptyset)$, a contradiction. Second, let $\{x\} \in E(N)$ and $x \notin Y$, then by superadditivity $\emptyset \in E(N)$ which contradicts liveness.

Now, consider any strategy profile σ_N. We have $o(\sigma_N) = h_{i_0}\big(\bigcap_{l=1}^k f(C_l)\big) \in Y$ because every h_i returns only elements in Y by construction.

For the case $C = N$, assume that $X \in E(N)$. We need to prove that $X \in E^\alpha_\mathcal{G}(N)$. By true playability, there exists $x \in X$ such that $\{x\} \in E(N)$. Now, let σ_N consist of strategies $\sigma_i = (f_i, t_i, h_i)$ such that $f_i(N) = \{x\}$ for every i. It is easy to see that $o(\sigma_N) = x$, and hence $\{x\} \in E^\alpha_\mathcal{G}(N)$. Thus, $X \in E^\alpha_\mathcal{G}(N)$ because $E^\alpha_\mathcal{G}(N)$ is closed under supersets.

Ad. Step 6. Dually to Step 5, we show that $E^\alpha_\mathcal{G}(C) \subseteq E(C)$. That is, assuming $X \notin E(C)$ we show that $X \notin E^\alpha_\mathcal{G}(N)$. We do it by a slight modification of the original proof from [11].

Suppose first that $C = N$. Then, $\overline{X} \in E(\emptyset)$ by N-maximality, and by Step 5 we have $\overline{X} \in E^\alpha_\mathcal{G}(\emptyset)$. Since $E^\alpha_\mathcal{G}$ is truly playable, we have also that $X \notin E^\alpha_\mathcal{G}(N)$.

Assume now that $C \neq N$, and let $j_0 \in \overline{C}$. Let σ_C be any strategy for coalition C. We must show that there is a strategy $\sigma_{\overline{C}}$ such that $o(\sigma_C, \sigma_{\overline{C}}) \notin X$. To show this, we take $\sigma_{\overline{C}} = (f_i, t_i, h_i)_{i \in \overline{C}}$ such that for all $C' \supseteq \overline{C}$ and for all $i \in \overline{C}$ we have $f_i(C') = S$. We also choose t_{j_0} such that $((t_1 + \ldots + t_n) \bmod n) + 1 = j_0$. Note that $\overline{C}$ must be a subset of one of the partitions C_l in $P_\infty(f)$, say C_{l_0}. Moreover, there must be a partitioning $\langle C_1, \ldots, C_k \rangle$ of $N \setminus C_{l_0}$ such that $\mathcal{G}(f) = f(C_{l_0}) \cap \bigcap_{l=1}^k f(C_l) = \bigcap_{l=1}^k f(C_l)$. Since $f(C_l) \in E(C_l)$ we get that $\mathcal{G}(f) \in E(N) \setminus C_{l_0}$ by superadditivity. By coalition-monotonicity and the fact that $N \setminus C_{l_0} \subseteq C$, we also have $\mathcal{G}(f) \in E(C)$. Finally, by (*) and superadditivity we obtain $\mathcal{G}(f) \cap \{x \mid \{x\} \in E(N)\} \in E(C)$.

Since $X \notin E(C)$ and $E(C)$ is closed under supersets, it must hold that $\mathcal{G}(f) \cap \{x \mid \{x\} \in E(N)\} \not\subseteq X$. Thus, there is some $s_0 \in S$ such that: $s_0 \in \mathcal{G}(f)$, $\{s_0\} \in E(N)$, and $s_0 \notin X$. Now we fix h_{j_0} so that $h_{j_0}(\mathcal{G}(f)) = s_0$. Then, $o(\sigma_C, \sigma_{\overline{C}}) = h_{j_0}(\mathcal{G}(f)) = s_0 \notin X$ which concludes the proof. □

4.3 Non-Truly Playable Structures

In this section we focus on the class of playable but not truly playable effectivity functions, hereafter called "non-truly playable". Non-truly playable effectivity functions have a simple abstract characterization, following from Proposition 5:

PROPOSITION 8. *Effectivity function $E : 2^N \to 2^{2^S}$ is non-truly playable if and only if it is playable and $E(\emptyset)$ is a non-principal filter.*

To see a more generic class of examples, consider an infinite domain S, and let $\mathcal{F}$ be any non-principal filter on S. Then we define an effectivity function $E_{\mathcal{F}}$ on S as follows.

- $E_{\mathcal{F}}(\emptyset) = \mathcal{F}$.
- $E_{\mathcal{F}}(N) = \{X \mid \overline{X} \notin \mathcal{F}\}$
- For each C with $\emptyset \subsetneq C \subsetneq N$ take $E_{\mathcal{F}}(C)$ to be any set of sets such that $E_{\mathcal{F}}(\emptyset) \subseteq E_{\mathcal{F}}(C) \subseteq E_{\mathcal{F}}(N)$ that is closed under outcome monotonicity and that are pairwise closed under regularity and superadditivity.

PROPOSITION 9. *$E_{\mathcal{F}}$ is playable but not truly playable.*

We omit the proof due to lack of space.

4.4 From Playable to Truly Playable Effectivity Functions

In this section we show that one can reconstruct a non-truly playable effectivity function into a truly playable one with "minimal" modifications. To do so, we interpret choices of the grand coalition containing multiple outcome states as ones that involve inherent nondeterminism. That is, we interpret $\{x_1, x_2, \dots\} \in E(N)$ as a choice where no agent has control over which state out of $x_1, x_2, \dots$ will become the outcome; as a consequence any of these states can possibly be encountered in the next moment. Under such assumption, it is possible to recover true playability by a simple extension of Pauly's procedure. The extension consists in adding an extra player $\mathbf{d}$ (the "decider") who settles the nondeterminism and decides which of $x_1, x_2, \dots$ is going to become the next state.

PROPOSITION 10. *Let $E : 2^N \to 2^{2^S}$ be a playable effectivity function. There exists a truly playable effectivity function $E' : 2^{N\cup\{\mathbf{d}\}} \to 2^{2^S}$ with additional player $\mathbf{d} \notin N$, such that:*

- *$E'(C) = E(C)$ for every $C \subseteq N, C \neq \emptyset$,*
- *$E'(\emptyset) = \{S\}$, and*
- *$E'(N \cup \{\mathbf{d}\}) = 2^S \setminus \{\emptyset\}$.*

PROOF. Given a playable E, we construct a strategic game whose α-effectivity function satisfies the properties above. Then, existence of a truly playable effectivity function follows immediately. The idea is to take the construction from the proof of Theorem 2.27 in [11] and reassign selection of the outcome state to the additional player $\mathbf{d}$.

Let $h : 2^S \setminus \{\emptyset\} \to S$ be any selector function that selects an arbitrary element from the argument set. In our case, h will designate the "default" outcome for each subset of S. Now, the game $\mathcal{G}$ is constructed as follows:

- $N' = N \cup \{\mathbf{d}\}$;
- The strategies of each player $i \neq \mathbf{d}$ are simply the player's assignments of coalitional choice, i.e., $\Sigma_i = F_i$, as in section 3.2;
- The strategies of $\mathbf{d}$ are state selections: $\Sigma_{\mathbf{d}} = S$;
- The transition function is based on the same partitioning of N as before, that yields $\langle C_1, \dots, C_k\rangle$. Then, the game proceeds to the state selected by the decider if his choice is consistent with the choices of the others, otherwise it proceeds to the appropriate "default" outcome:

$$o(\sigma_N, s) = \begin{cases} s & \text{if } s \in \bigcap_{i=1}^k f(C_l) \\ h(\bigcap_{i=1}^k f(C_l)) & \text{else.} \end{cases}$$

Now, it is easy to see that for every $\emptyset \subsetneq C \subsetneq N$ indeed $E^\alpha_{\mathcal{G}}(C) = E(C)$ because that was the case in the original construction, and the only difference now is that $\mathbf{d}$ "took over" the selection of a state in $\bigcap_{i=1}^k f(C_l)$ from a collective choice of N. For $C = N$, we also have $E^\alpha_{\mathcal{G}}(N) = E(N)$ since for every σ_N we get by superadditivity that $\bigcap_{i=1}^k f(C_l) \in E(N)$, and every state from the intersection can be potentially selected by $\mathbf{d}$. Moreover, $\{s\} \in E^\alpha_{\mathcal{G}}(N \cup \{\mathbf{d}\})$ for every $s \in S$ because $\{s\}$ is enforced by $\sigma_{N\cup\{\mathbf{d}\}} = \langle f_1, \dots, f_{|N|}, s\rangle$ such that $f_i = S$ for all $i \in N$. Thus, by outcome monotonicity, $E^\alpha_{\mathcal{G}}(N \cup \{\mathbf{d}\}) = 2^S \setminus \{\emptyset\}$. Finally, by true playability of $E^\alpha_{\mathcal{G}}$, we have $E^\alpha_{\mathcal{G}}(\emptyset) = \{\{s \mid \{s\} \in E^\alpha_{\mathcal{G}}(N \cup \{\mathbf{d}\})\}\} = \{S\}$. We observe additionally that $E^\alpha_{\mathcal{G}}(\mathbf{d}) = \{\{s\} \cup \{h(X) \mid X \in E^\alpha_G(\mathbf{d}) \text{ and } s \notin X\} \mid s \in S\}$. □

5. LOGICS AND TRUE PLAYABILITY

In this section, we investigate the impact of true playability on logics of coalitional ability. We begin by indicating that the validities of Coalition Logic do not change if we restrict models to truly playable. As a consequence, CL (and even ATL) cannot distinguish between playable and truly playable models. Then, we discuss two extensions of CL that can discern the two classes of structures.

For preliminaries on modal logic see e.g. [5, 3].

5.1 Ramifications for CL

We recall from [11] that the models of Coalition Logic (also called *coalition models*) are neighborhood models of the type $M = (W, E, V)$ consisting of a set of states W, a dynamic effectivity function $E : W \to \left(2^N \to 2^{2^W}\right)$ and a valuation function $V : W \to 2^P$ ranging over a countable set of atomic propositions P. A coalitional *frame* is a coalition model minus the valuation. A model (resp. frame) is *playable* iff it includes only playable effectivity functions at each $w \in W$, and *truly playable* iff it includes only truly playable functions at each $w \in W$. The operator $[C]$ is interpreted as follows:

$$M, w \models [C]\phi \text{ if and only if } \phi^M \in E(w)(C),$$

where ϕ^M is the set $\{v \in W | M, v \models \phi\}$. Formula φ is *valid in model M* ($M \models \varphi$) if and only if it holds in every state in M; φ is *valid in frame F* ($F \models \varphi$) if and only if it is valid in every model based on F. We extend these notions to classes of models and frames in the obvious way.

We note that the problem with Pauly's Representation Theorem has no repercussions on the semantics of CL and the soundness/completeness results for that logic. Let us formally define Play to be the class of playable coalition models, and TrulyPlay as the class of truly playable models. Since TrulyPlay $\subsetneq$ Play, every CL formula valid in Play is valid in TrulyPlay, too. The converse follows from the finite model property of CL with respect to Play [11] and the fact that it coincides with TrulyPlay on finite models.

COROLLARY 11. *The axiomatization of CL from [11] is sound and complete wrt truly playable coalition models, and hence, also with respect to strategic game models.*

Furthermore, the semantics based on effectivity functions can be extended to ATL (see. [6]; also, cf. [11] for the fragment of ATL without "until", called *Extended CL*). Again, it can be shown that Play and TrulyPlay determine the same

sets of validities for ATL, by checking the soundness of the axiomatization for ATL given in [7] for Play, and using the completeness result for ATL with respect to strategic game models (equivalently, TrulyPlay) proved in the same paper.

5.2 CL with Infinite Disjunctions

One possible extension of CL that can tell apart the classes Play and TrulyPlay involves infinite disjunctions of formulas. The idea is that in truly playable models, every choice of the grand coalition can be narrowed down to a singleton. The infinitary disjunction $\bigvee_{i\in\mathcal{I}}$ for a set of indices $\mathcal{I}$ has the natural interpretation:

$$M, w \models \bigvee_{i\in\mathcal{I}} \phi_i \text{ if and only if } M, w \models \phi_i \text{ for some } i \in \mathcal{I}.$$

PROPOSITION 12. *For any cardinal number*[2] κ*, let* Play_κ *(resp.* $\mathsf{TrulyPlay}_\kappa$*) denote the class of playable (resp. truly playable) coalition models with the domain of outcomes* W *of cardinality at most* κ *and let* $\{p_\iota\}_{\iota\in\kappa}$ *be a set of different propositional letters.*

1. $\mathsf{Play}_\kappa \not\models [N]\bigvee_{\iota\in\kappa} p_\iota \leftrightarrow \bigvee_{\iota\in\kappa}[N]p_\iota$;
2. $\mathsf{TrulyPlay}_\kappa \models [N]\bigvee_{\iota\in\kappa} p_\iota \leftrightarrow \bigvee_{\iota\in\kappa}[N]p_\iota$.

PROOF. For (1) simply check the example in Section 3.1 with the set S being κ and every state ι associated with a designating atomic proposition p_ι. Claim (2) follows from Proposition 5. □

5.3 CL with "Outcome Selector" Modality

Adding infinitary operators to a logical language makes its practical applicability problematic. Here we propose another (in fact, simpler) extension of CL, by adding a new *normal* modality $\langle\mathcal{O}\rangle$, with a dual $[\mathcal{O}]$, called "outcome selector". The informal reading of $\langle\mathcal{O}\rangle\phi$ should be "there is an outcome state, enforceable by the grand coalition and satisfying ϕ". In order to define the semantics of $\langle\mathcal{O}\rangle$ in the usual semantic way, we first expand coalition models to what we call *extended coalition models* with an additional "outcome enforceability" relation R. Later we will use axioms to impose the right behavior of R.

DEFINITION 9 (EXTENDED COALITION FRAMES). *An* extended (playable) coalition frame *is a neighbourhood frame* $F = (W, E, R)$ *where* W *is a set of outcomes,* E *a playable effectivity function,* R *a binary relation on* W*.*

An extended coalition model is an extended coalition frame endowed with a valuation function. Given an extended coalition model $M = (W, E, R, V)$, the modality $\langle\mathcal{O}\rangle$ is interpreted as follows.

$M, w \models \langle\mathcal{O}\rangle\phi$ if and only if wRs and $M, s \models \phi$.

That is, $\langle\mathcal{O}\rangle$ has standard Kripke semantics with respect to the outcome enforceability relation R. Note that extended coalition models do not require any interaction between the effectivity function and the relation R. However, given the intuitive reading of the relation R, the interaction suggests itself, and the following definition account for that.

DEFINITION 10 (STANDARD COALITION FRAMES). *A* standard coalition frame *is an extended coalition frame such that, for all* $w, v \in W$*, we have* wRv *if and only if* $\{v\} \in E(w)(N)$*.*

[2]We regard cardinals as (special) ordinals in von Neumann sense: any ordinal is the set of all smaller ordinals.

A standard coalition model is a standard coalition frame with a valuation function. Depending on the properties of the underlying effectivity functions we call extended coalition frames and models playable or truly playable.

5.4 Characterizing Standard Truly Playable Coalition Frames

PROPOSITION 13. *An extended coalition frame* F *is standard and truly playable if and only if* $F \models [N]q \leftrightarrow \langle\mathcal{O}\rangle q$*, for any atomic proposition* q*.*

PROOF. Left to right: Assume that F is standard and truly playable. Assume first that $(F, V), w \models [N]q$ for any V and $w \in W$. By definition of E we have that $q^M \in E(w)(N)$. As F is truly playable there is $v \in q^M$ with $\{v\} \in E(w)(N)$. However F is also standard so wRv. But this means that $(F, V), w \models \langle\mathcal{O}\rangle q$. Conversely, if $(F, V), w \models \langle\mathcal{O}\rangle q$ then wRv for some $v \in q^M$. F being standard we have that $\{v\} \in E(w)(N)$. By outcome monotonicity $q^M \in E(w)(N)$, i.e. $(F, V), w \models [N]q$.

Right to left: Assume that $F \models [N]q \leftrightarrow \langle\mathcal{O}\rangle q$. Let us first prove that F is standard. Suppose wRv for some $w, v \in W$. Let V be a valuation that assigns the atom q only to v. We have that $M, w \models \langle\mathcal{O}\rangle q$. Then, by the assumptions we also have $M, w \models [N]q$, which means that $\{v\} \in E(w)(N)$. Conversely, suppose now that $\{v\} \in E(w)(N)$. For the same valuation V we must have that $(F, V), w \models [N]q$ and by assumption that $\langle\mathcal{O}\rangle q$, which means that wRv. Thus, F is standard. To prove that F is truly playable, assume that for some $X \subseteq W$, $X \in E(w)(N)$ and let now V be a valuation function such that $V(q) = X$. By definition of E we have that $(F, V), w \models [N]q$, hence by assumption, that $(F, V), w \models \langle\mathcal{O}\rangle q$, which means that wRv for some $v \in V(q)$. Then, F being standard, $\{v\} \in E(w)(N)$. □

5.5 Standard Truly Playable Models: Axioms

We propose the following axiomatic system TPCL for the class of standard truly playable coalition models TrulyPlay, extending Pauly's axiomatization of CL. The axioms include propositional tautologies plus the following schemes:

1. $[N]\top$
2. $\neg[C]\bot$
3. $\neg[\emptyset]\phi \to [N]\neg\phi$
4. $[C]\phi \wedge [D]\psi \to [C\cup D](\phi\wedge\psi)$ for any disjoint $C, D \subseteq N$
5. $[N]\phi \leftrightarrow \langle\mathcal{O}\rangle\phi$
6. $[\mathcal{O}](\phi \to \psi) \to ([\mathcal{O}]\phi \to [\mathcal{O}]\psi)$.

The inference rules are: Modus Ponens, plus:

$$\frac{\phi \to \psi}{[C]\phi \to [C]\psi}, \quad \text{and} \quad \frac{\phi}{[\mathcal{O}]\phi}.$$

REMARK. Axiom 5 seems to render the outcome modality $[\mathcal{O}]$ redundant. This, however, is not so, because the semantics of the modality $[N]$ is (monotonic) neighbourhood semantics, while the semantics of $[\mathcal{O}]$ is *by default* Kripke semantics. Relating these by Axiom 5 suffices to enforce the true playability of the underlying frames, as shown in Proposition 13. On the other hand, it is easy to show that the normality Axiom 6, as well as the necessitation rule for $[\mathcal{O}]$, are derivable from the rest. We have only added them to emphasize the fact that $[\mathcal{O}]$ is a normal modality.

The proof of the following claim is routine.

PROPOSITION 14. *TPCL is sound for the class TrulyPlay: every formula derivable in TPCL is valid in TrulyPlay.*

5.6 Completeness for TPCL

THEOREM 15 (COMPLETENESS THEOREM). *Every formula consistent in TPCL is satisfiable in TrulyPlay. Consequently, the logic TPCL is complete for the class TrulyPlay.*

We will prove the completeness, using canonical model construction followed by filtration for monotonic modal logics, partly using constructions from [5] and [11]. Thus, we will also obtain finite model property for TPCL. Here we only sketch the standard canonical model construction and refer the reader for further details to [5] and [11].

We start with a formula δ which is consistent in TPCL. By a well-known argument, it is contained in some maximal TPCL-consistent set. We take the set $W^{\mathcal{L}}$ of maximally consistent sets and define for every formula ϕ the *proof set* of ϕ as $\phi^* = \{s \in W^{\mathcal{L}} \mid \phi \in s\}$.

To shorten the notation we hereafter denote the logic TPCL by $\mathcal{L}$.

DEFINITION 11 (CANONICAL MODEL). *The canonical model for TPCL is $M^{\mathcal{L}} = (W^{\mathcal{L}}, E^{\mathcal{L}}, R^{\mathcal{L}}, V^{\mathcal{L}})$ where:*
- $w \in V^{\mathcal{L}}(p)$ if and only if $p \in w$;
- $X \in E^{\mathcal{L}}(w)(C)$ iff $\exists \psi^* \subseteq X : [C]\psi \in w$, for $C \neq N$
- $X \in E^{\mathcal{L}}(w)(C)$ iff $\forall \psi^*$ if $X \subseteq \psi^*$ then $[C]\psi \in w$, for $C = N$
- $wR^{\mathcal{L}}v$ iff $\forall \psi$, if $\psi \in v$ then $\langle\mathcal{O}\rangle\psi \in w$.

Some remarks:

- That $E^{\mathcal{L}}$ is playable and well-defined is proved in [11].
- The canonical relation for N is defined in [11] in the following equivalent way: $X \in E^{\mathcal{L}}(w)(N)$ if and only if $[\emptyset]\psi \notin w$ for all ψ^* such that $\psi^* \subseteq \overline{X}$. The equivalence follows easily from the fact that $\vdash_{\mathcal{L}} [N]\phi \leftrightarrow \neg[\emptyset]\neg\phi$.

PROPOSITION 16 (TRUTH LEMMA). *For any $w \in W^{\mathcal{L}}$ we have that $M^{\mathcal{L}}, w \models \phi$ if and only if $\phi \in w$.*

PROOF. By induction on the length of ϕ: standard for atomic propositions, boolean formulas, and formulas of the form $\langle\mathcal{O}\rangle\psi$; proved in [11] for formulas of the form $[C]\psi$. □

The canonical model is an extended coalition model, however it is neither standard nor truly playable. The reason for that is the fact that for all $\psi \in \mathcal{L}, \psi \in v$ implies that $[N]\psi \in w$ is not sufficient to conclude that $\{v\} \in E^{\mathcal{L}}(w)(N)$ as states are not characterized by unique formulas of the language of $\mathcal{L}$. In order to obtain a standard and truly playable model satisfying the given $\mathcal{L}$-consistent formula δ we are going to filter the canonical model with the set $\Sigma(\delta)$, obtained by taking all subformulae of δ and closing under boolean operators. That set is finite *up to propositional equivalence.*

Filtrations.

First, we define a general notion of filtration for extended coalition models and then a special filtration construction that preserves playability. Filtrations of coalition models are introduced, e.g., in [8] for the purpose of axiomatizing Nash-consistent Coalition Logic. Here we only add the filtration for the relation corresponding to the modality $\langle\mathcal{O}\rangle$.

Let $M = (W, E, R, V)$ be an extended coalition model and Σ a subformula-closed set of formulas over $\mathcal{L}$. The equivalence classes induced by Σ on M are defined as follows:

$v \equiv_\Sigma w \Leftrightarrow$ for all $\phi \in \Sigma : M, v \models \phi$ if and only if $M, w \models \phi$.

We denote the equivalence class to which v belongs by $|v|$ and the set $\{|v| \mid v \in X\}$ by $|X|$ for any $v \in W$ and $X \subseteq W$.

DEFINITION 12 (FILTRATION). *Let $M = (W, E, R, V)$ be an extended coalition model and Σ a subformula closed set of formulas over $\mathcal{L}$. A coalition model $M^f_\Sigma = (W^f_\Sigma, E^f_\Sigma, R^f_\Sigma, V^f_\Sigma)$ is a filtration of M through Σ whenever the following conditions are satisfied:*

- $W^f_\Sigma = |W|$.
- *For all $C \subseteq N$ and $\phi \in \Sigma$, $\phi^M \in E(w)(C)$ implies $\{|v| \mid M, v \models \phi\} \in E^f_\Sigma(|w|)(C)$.*
- *For all $C \subseteq N$ and $Y \subseteq |W|$: $Y \in E^f_\Sigma(|w|)(C)$ implies that for all $\phi \in \Sigma$ if $\phi^M \subseteq \{v \mid |v| \in Y\}$ then $\phi^M \in E(w)(C)$.*
- *If wRv then $|w|R|v|$.*
- *If $|w|R|v|$ then for all $\langle\mathcal{O}\rangle\phi \in \Sigma$, if $M, v \models \phi$ then $M, w \models \langle\mathcal{O}\rangle\phi$.*
- *$V^f_\Sigma(p) = |V(p)|$ for all atoms $p \in \Sigma$.*

The conditions above are needed to ensure the Filtration Lemma, as showed in [8] for the neighbourhood relations and e.g., in [5] for the binary relation.

PROPOSITION 17 (FILTRATION LEMMA). *If $M^f_\Sigma = (W^f_\Sigma, E^f_\Sigma, R^f_\Sigma, V^f_\Sigma)$ is a filtration of M through Σ then for all $\phi \in \Sigma$ we have that $M, w \models \phi$ if and only if $M^f_\Sigma, |w| \models \phi$.*

DEFINITION 13 (PLAYABLE FILTRATION). *Let $M = (W, E, R, V)$ be an extended coalition model and $\Sigma(\delta)$ the boolean closure of the set of subformulas of δ, such that $\delta \in \mathcal{L}$, the language of TPCL. A coalition model $M^F_{\Sigma(\delta)} = (W^F_{\Sigma(\delta)}, E^F_{\Sigma(\delta)}, R^F_{\Sigma(\delta)}, V^F_{\Sigma(\delta)})$ is a* playable filtration *of M through $\Sigma(\delta)$ whenever the following conditions are satisfied:*

- $W^F_{\Sigma(\delta)} = |W|$.
- *For all $C \subsetneq N, C \neq N$, and $Y \subseteq |W|$: $Y \in E^F_{\Sigma(\delta)}(|w|)(C)$ if and only if there exists $\phi \in \Sigma(\delta)$ such that $\phi^M \subseteq \{v \mid |v| \in Y\}$ and $\phi^M \in E(w)(C)$.*
- *For all $Y \subseteq |W|$: $Y \in E^F_{\Sigma(\delta)}(|w|)(N)$ if and only if $\overline{Y} \notin E^F_{\Sigma(\delta)}(|w|)(\emptyset)$.*
- *$|w|R^F_{\Sigma(\delta)}|v|$ if and only if there exists $w' \in |w|, \exists v' \in |v|$ such that $w'Rv'$.*
- *$V^F_{\Sigma(\delta)}(p) = |V(p)|$ for all atoms $p \in \Sigma(\delta)$.*

That $M^F_{\Sigma(\delta)}$ is a filtration in the sense of Definition 12 is proved in [8] for the coalitional modalities. We have added to that a minimal filtration for modality $\langle\mathcal{O}\rangle$. So $M^F_{\Sigma(\delta)}$ is a filtration in the sense of Definition 12. In [8] it is also shown that playability is preserved by that filtration and that every subset of $W^F_{\Sigma(\delta)}$ is definable by a formula of $\Sigma(\delta)$ as follows. First, for every state $|w| \in |W|$ we define

$$\chi^F_{\Sigma(\delta)}(|w|) := \bigwedge\{\phi \in \Sigma(\delta) \mid M^F_{\Sigma(\delta)}, |w| \models \phi\}.$$

Then, for every $Y \subseteq |W|$ we put

$$\chi^F_{\Sigma(\delta)}(Y) := \bigvee\{\chi^F_{\Sigma(\delta)}(|w|) \mid |w| \in Y\}.$$

It is straightforward to show, using the filtration lemma, that for every $Y \subseteq |W|$:

$$M^F_{\Sigma(\delta)}, |w| \models \chi^F_{\Sigma(\delta)}(Y) \text{ if and only if } |w| \in Y,$$

that is, $\chi^F_{\Sigma(\delta)}(Y)$ indeed characterizes the set y in $M^F_{\Sigma(\delta)}$.

PROPOSITION 18. *$M^{\mathcal{L},F}_{\Sigma(\delta)}$ is standard and truly playable.*

PROOF. To prove that $M^{\mathcal{L},F}_{\Sigma(\delta)}$ is standard we have to show that for each $w, v \in W$, $|v| R^{\mathcal{L},F}_{\Sigma(\delta)} |w|$ if and only if $\{|v|\} \in E^{\mathcal{L},F}_{\Sigma(\delta)}(|w|)(N)$. From right to left it is straightforward. For the other direction, suppose $|v| R^{\mathcal{L},F}_{\Sigma(\delta)} |w|$. Then $M^{\mathcal{L},F}_{\Sigma(\delta)}, |v| \models \langle\mathcal{O}\rangle \chi^F_{\Sigma(\delta)}(|w|)$ by definition of $R^{\mathcal{L},F}_{\Sigma(\delta)}$ and by the properties of filtrations. By the fact that $R^{\mathcal{L},F}_{\Sigma(\delta)}$ is a minimal filtration we have that $\exists w' \in |w|, \exists v' \in |v|$ such that $v' R^{\mathcal{L}} w'$. By definition of $R^{\mathcal{L}}$ and the Truth Lemma we have that $M^{\mathcal{L}}, v' \models \langle\mathcal{O}\rangle \chi^F_{\Sigma(\delta)}(|w|)$. By the axioms of $\mathcal{L}$ and the Truth Lemma we have $M^{\mathcal{L}}, v' \models [N] \chi^F_{\Sigma(\delta)}(|w|)$, hence $M^{\mathcal{L}}, v' \models \neg[\emptyset]\neg \chi^F_{\Sigma(\delta)}(|w|)$. Then $(\neg \chi^F_{\Sigma(\delta)}(|w|))^{M^{\mathcal{L}}} \notin E^{\mathcal{L}}(v')(\emptyset)$ by the definition of $E^{\mathcal{L}}$. But, by Definition 12 $\{(\neg \chi^F_{\Sigma(\delta)}(|w|))^{M^{\mathcal{L},F}_{\Sigma(\delta)}}\} \notin E^{\mathcal{L},F}_{\Sigma(\delta)}(|v|)(\emptyset)$ and in turn $\{(\chi^F_{\Sigma(\delta)}(|w|))^{M^{\mathcal{L},F}_{\Sigma(\delta)}}\} \in E^{\mathcal{L},F}_{\Sigma(\delta)}(|v|)(N)$. Recall now that $(\chi^F_{\Sigma(\delta)}(|w|))^{M^{\mathcal{L},F}_{\Sigma(\delta)}} = |w|$.

Now, to prove that $M^{\mathcal{L},F}_{\Sigma(\delta)}$ is truly playable, assume $Y \in E^{\mathcal{L},F}_{\Sigma(\delta)}(|w|)(N)$. Then, $(\neg \chi^F_{\Sigma(\delta)}(Y))^{M^{\mathcal{L},F}_{\Sigma(\delta)}} \notin E^{\mathcal{L}}(w)(\emptyset)$ by the definition of filtration, which means that for all $\phi \in \Sigma(\delta)$, if $\{v \mid |v| \in (\neg \chi^F_{\Sigma(\delta)}(Y))^{M^{\mathcal{L},F}_{\Sigma(\delta)}}\} \subseteq \phi^M$ then $\phi^M \notin E^{\mathcal{L}}(w)(\emptyset)$. In particular $(\neg \chi^F_{\Sigma(\delta)}(Y))^{M^{\mathcal{L}}} \notin E^{\mathcal{L}}(w)(\emptyset)$. By the definition of $E^{\mathcal{L}}$ we have that $[\emptyset]\neg \chi^F_{\Sigma(\delta)}(Y) \notin w$ and by true playability that $\langle\mathcal{O}\rangle \chi^F_{\Sigma(\delta)}(Y) \in w$. By the definition of canonical relation for $\langle\mathcal{O}\rangle$ we have that there exists v with $w R^{\mathcal{L}} v$ such that $\chi^F_{\Sigma(\delta)}(Y) \in v$. By definition of filtration $|w| R^{\mathcal{L},F}_{\Sigma(\delta)} |v|$ and by the Filtration Lemma $M^{\mathcal{L},F}_{\Sigma(\delta)}, |v| \models \chi^F_{\Sigma(\delta)}(Y)$. Finally, $\{|v|\} \in E^{\mathcal{L},F}_{\Sigma(\delta)}(|w|)(N)$ since $M^{\mathcal{L},F}_{\Sigma(\delta)}$ is standard. □

This completes the proof of the Completeness theorem 15.

COROLLARY 19 (FINITE MODEL PROPERTY). *The logic TPCL has the finite model property with respect to the class of models TrulyPlay.*

6. CONCLUSIONS

In this paper, we have revisited the correspondence between two classes of abstract game forms: strategic games from noncooperative game theory on one hand, and effectivity functions from cooperative game theory on the other. We consider our contribution as threefold. First, we have corrected a well-known and often used result from [11] relating strategic games and playable effectivity functions. We have shown that strategic games do not correspond to all playable functions, but to a strict subset of the class, which we call *truly playable effectivity functions*. Second, we have provided several abstract characterizations of truly playable functions. We have also shown that the remaining playable effectivity functions (that we call non-truly playable) are induced by non-principal filters, and hence only scenarios with infinitely many possible outcomes can fall in that class. Finally, we have pointed out that Coalition Logic and ATL are not expressive enough to characterize true playability. On the other hand, they can be extended in a relatively simple way to obtain such a characterization. To this purpose we have proposed an extension of Coalition Logic with a normal *outcome selector* modality that we have shown sufficient for axiomatic characterization of truly playable structures.

The importance of our work is mainly theoretical. Essentially, it implies that all the claims that have been proved using Pauly's correspondence between playable effectivity functions and games should be revisited and possibly reinterpreted in the light of the results presented here. Example of such issues, already addressed here, include: axiomatization for Coalition Logic in the class of multi-player game models, axiomatization of ATL in coalitional models, and the respective finite model properties. In practical terms, this also means that, whenever a decision procedure is built on those theoretical results, the designer should be aware of the correct correspondence between the two classes of game models, which is especially relevant for satisfiability-checking algorithms. Tableaux for extensions of Coalition Logic, like the one for a combination of CL and description logic $\mathcal{ALC}$ from [12], are examples of such procedures.

Acknowledgements. Wojciech Jamroga acknowledges the support of the FNR (National Research Fund) Luxembourg under project S-GAMES – C08/IS/03. Valentin Goranko acknowledges the support of the *HYLOCORE* project, funded by the Danish Natural Science Research Council.

7. REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In W. P. de Roever, H. Langmaack, and A. Pnueli, editors, *COMPOS*, volume 1536 of *Lecture Notes in Computer Science*, pages 23–60. Springer, 1997.

[2] N. Belnap, M. Perloff, and M. Xu. *Facing the Future: Agents and Choices in Our Indeterminist World*. Oxford University Press, 2001.

[3] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science, 2001.

[4] J. Broersen, A. Herzig, and N. Troquard. A normal simulation of coalition logic and an epistemic extension. In D. Samet, editor, *Proceedings Theoretical Aspects Rationality and Knowledge (TARK XI), Brussels*, pages 92–101. ACM Digital Library, 2007.

[5] B. Chellas. *Modal Logic: an Introduction*. Cambridge University Press, 1980.

[6] V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280, 2004.

[7] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theor. Comput. Sci.*, 353(1-3):93–117, 2006.

[8] H. H. Hansen and M. Pauly. Axiomatising Nash-consistent coalition logic. In *JELIA*, pages 394–406, 2002.

[9] H. Moulin and B. Peleg. Cores of effectivity functions and implementation theory. *Journal of Mathematical Economics*, 10(1):115–145, June 1982.

[10] M. Osborne and A. Rubinstein. *A course in Game Theory*. The MIT Press, 1994.

[11] M. Pauly. *Logic for Social Software*. ILLC Dissertation Series, 2001.

[12] I. Seylan and W. Jamroga. Description logic for coalitions. In *Proceedings of AAMAS'09*, pages 425–432, 2009.

Scientia Potentia Est*

Thomas Ågotnes
Dept of Information Science and Media Studies
University of Bergen
PB. 7802, 5020 Bergen
Norway
thomas.agotnes@uib.no

Wiebe van der Hoek
Dept of Computer Science
University of Liverpool
Liverpool L69 7ZF
United Kingdom
wiebe@csc.liv.ac.uk

Michael Wooldridge
Dept of Computer Science
University of Liverpool
Liverpool L69 7ZF
United Kingdom
mjw@liv.ac.uk

ABSTRACT

In epistemic logic, Kripke structures are used to model the distribution of information in a multi-agent system. In this paper, we present an approach to *quantifying* how much information each particular agent in a system has, or how important the agent is, with respect to some fact represented as a goal formula. It is typically the case that the goal formula is distributed knowledge in the system, but that no individual agent alone knows it. It might be that several different groups of agents can get to know the goal formula together by combining their individual knowledge. By using power indices developed in voting theory, such as the Banzhaf index, we get a measure of how important an agent is in such groups. We analyse the properties of this notion of information-based power in detail, and characterise the corresponding class of voting games. Although we mainly focus on distributed knowledge, we also look at variants of this analysis using other notions of group knowledge. An advantage of our framework is that power indices and other power properties can be expressed in standard epistemic logic. This allows, e.g., standard model checkers to be used to quantitatively analyse the distribution of information in a given Kripke structure.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

Epistemic logic, power indices, model checking

1. INTRODUCTION

Epistemic logic is widely used in the multi-agent systems community to reason about the knowledge and ignorance of agents in terms of the information they possess [5]. In many situations, it would be useful to be able to *quantify* how information is distributed in a system, or to reason about the *relative importance* of the information that different agents have. In general, it is difficult to answer the question of whether an agent has more information than another agent except for in special cases, such as when one agent knows everything another agent knows [15]. In this paper, we quantify the distribution of information in a system in a specific sense satisfying two assumptions. The first is that we are interested in who knows more *about* some given fact. The second is that we are interested in situations where information can be *communicated* between agents, and it is not always possible or desirable to communicate with every other agent in the system.

Consider the following situation. M knows that if sales are up this quarter, the stock price will increase ($p \rightarrow q$). T knows that if the new CEO has signed the contract, the stock price will increase ($r \rightarrow q$). W knows that sales are up this quarter and that the new CEO has signed the contract ($p \wedge r$). Assume that this describes all (relevant) facts that the three agents know. Who knows more? We are here interested in a more specific type of question: who has the most *important* or *valuable* information *about* whether or not the stock price will increase (q), *in* a social setting where communication is possible? None of the agents alone knows q, but they can *combine* their knowledge to find out that q is in fact true. And here the importance of the knowledge of the three agents differ: M and W can together find out q, as can T and W. M and T cannot. It can thus be argued that W knows more about q in this social setting, since he can combine his knowledge in several different ways with others' knowledge – and, indeed, it is not hard to see that W's knowledge is *necessary* for any group to be able to find out q, unlike that of M or T. If it is important for each individual agent to find out q, and since no agent already knows q, the only possibility is to communicate with someone else; in which case clearly W would be considered the most *important* agent.

In this paper we analyse the relative importance of the knowledge each agent has in a system where information about some fact or objective (q in our example above) is distributed throughout the system. To this end, we employ *power indices* such as the Banzhaf index, known from voting theory. The starting point is a pointed Kripke structure. It is typically the case that the objective is distributed knowledge in the system, but that no individual agent knows it. It might be that several different groups of agents can get to know the objective by combining their knowledge. Our approach measures the importance of an agent in an arbitrary group of agents wrt. deriving the objective. We consider an agent to be powerful, or to have important information, if the probability of changing the distributed knowledge in the group from ignorance to knowledge about the objective by joining some arbitrary group, is high. This concept of *information based power* can, e.g., be used to identify agents that are crucial to the functioning of the multi-agent system.

The question of "who knows more" in epistemic logic has re-

*For also Knowledge itself is Power; with apologies to Francis Bacon.

Cite as: Scientia Potentia Est, Thomas Ågotnes and Wiebe van der Hoek and Michael Wooldridge, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 735-742.

cently been studied in [15]. The notion of information based power we introduce in this paper is a more fine-grained generalisation: if an agent knows more in the sense of [15] then she has a higher power index, but not necessarily the other way around. Solution concepts for coalitional games have recently been used to measure the degree of inconsistency in databases [8]. In [2] power indices are used to analyse the relative importance of agents when in terms of complying or not complying with a *normative system* defined over a Kripke-like structure [12, 1]. However, we are not aware of any approaches using power indices to measure relative importance of agents in terms of their knowledge/information as described by a Kripke structure.

The paper is organised as follows. In the two next sections we briefly review some background material about epistemic logic and power indices that we will use. In Section 4 we define power indices for agents, given a pointed Kripke structure and a goal formula. We give a complete characterisation of the power indices that can be obtained in this way, study their properties in detail, and show how standard epistemic logic can be used to express power properties. Since these power properties can be expressed in epistemic logic, we can also use epistemic logic to reason about agents' *knowledge* about such properties. In Section 5 we study what agents know about the distribution of information-based power in the system. In most of the paper we use distributed knowledge to define power, but in Section 6 we discuss other types of group knowledge as well. We conclude in Section 7.

2. EPISTEMIC LOGIC

Assume a finite set of agents $Ag = \{1, \ldots, n\}$ and a countably infinite set of atomic propositions Θ. The language $\mathcal{L}_K$ of the epistemic logic $S5_n$ is defined by the following grammar:

$$\varphi ::= \top \mid p \mid K_i\varphi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$$

where $p \in \Theta$ and $i \in Ag$. An *epistemic (Kripke) structure*, M, (over Ag, Θ) is an $(n+2)$-tuple [5]:

$$M = \langle W, \sim_1, \ldots, \sim_n, \pi \rangle, \quad \text{where}$$

- W is a finite, non-empty set of *states*;
- $\sim_i \subseteq W \times W$ is an *epistemic accessibility relation* for each agent $i \in Ag$, where each $\sim_i$ is an equivalence relation; and
- $\pi : W \to 2^{\Theta}$ is a Kripke valuation function, which gives the set of primitive propositions satisfied in each state.

Formulae are interpreted in a *pointed structure*, a pair M, s, where M is a model and s is a state in M, as follows.

- $M, s \models \top$
- $M, s \models p$ iff $p \in \pi(s)$ (where $p \in \Theta$)
- $M, s \models \neg\varphi$ iff $M, s \not\models \varphi$
- $M, s \models \varphi \wedge \psi$ iff $M, s \models \varphi$ and $M, s \models \psi$
- $M, s \models K_i\varphi$ iff for all t such that $s \sim_i t$, $M, t \models \varphi$.

We will make use of extensions of $S5_n$ with *group knowledge*. To this end, when $G \subseteq Ag$, we denote the union of G's accessibility relations by $\sim_G^E$, so $\sim_G^E = (\bigcup_{i \in G} \sim_i)$. We use $\sim_G^C$ to denote the transitive closure of $\sim_G^E$. Finally, $\sim_G^D$ denotes the intersection of G's accessibility relations (cf. [5, p.66–70]). The logics $S5_n^D$, $S5_n^C$ and $S5_n^{CD}$ are obtained as follows. The respective languages, $\mathcal{L}_D$, $\mathcal{L}_C$, and $\mathcal{L}_{CD}$, are obtained by adding the clause $D_G\varphi$, $C_G\varphi$, and both, respectively, where $G \subseteq Ag$, to the definition of $\mathcal{L}_K$. The interpretation of the two group operators:

- $M, s \models D_G\varphi$ iff for all t such that $s \sim_G^D t$, $M, t \models \varphi$
- $M, s \models C_G\varphi$ iff for all t such that $s \sim_G^C t$, $M, t \models \varphi$

We use the same notation for the satisfaction relation for all these logics; it will be clear from context which logic we are working in. As usual, we write $M \models \varphi$ if $M, s \models \varphi$ for all s in M, and $\models \varphi$ if $M \models \varphi$ for all M; in this latter case, we say that φ is *valid*. A formula is *satisfied* in a pointed model if it is true. When Φ is a set of formulae, $\Phi \models \varphi$, Φ *entails* φ, means that any pointed model that satisfies Φ also satisfies φ. A formula is *satisfiable* if there exists a pointed model that satisfies it. A formula or set of formulae is satisfiable in a *set* of pointed models if it is satisfied by *at least one* pointed model in that set. The usual propositional abbreviations are used, in addition to $E_G\varphi$ ($G \subseteq Ag$) for $\bigwedge_{i \in G} K_i\varphi$; $\hat{K}_i\varphi$ for $\neg K_i\neg\varphi$; $\hat{D}_G\varphi$ for $\neg D_G\neg\varphi$ and $\hat{C}_G\varphi$ for $\neg C_G\neg\varphi$. We will often abuse notation and write singleton sets of agents $\{i\}$ as i.

$E_G\varphi$ means that all individuals in the group G know φ. $D_G\varphi$ means that φ is distributed knowledge among G. Roughly speaking, this knowledge would come about if all members of G were to share their information (but see also Section 4.2). $C_G\varphi$, that φ is common knowledge in G, means that $E_G\varphi \wedge E_GE_G\varphi \wedge E_GE_GE_G\varphi \wedge \ldots$. These concepts of group and individual knowledge are related as follows (with $i \in G$):

$$\models (C_G\varphi \to E_G\varphi) \wedge (E_G\varphi \to K_i\varphi) \wedge (K_i\varphi \to D_G\varphi) \wedge (D_G\varphi \to \varphi)$$

The above implications express that common knowledge is the strongest property, and truth the weakest. However, since $C_G\varphi$ is such a strong notion, this often means it will only be obtained for 'weak' φ. Or [5], common knowledge can be paraphrased as what 'any fool knows', while distributed knowledge corresponds to what 'a wise man knows'.

Finally, the *knowledge set* of $G \subseteq Ag$ in M, s is:

$$\mathcal{K}_G(M, s) = \{\varphi \in \mathcal{L}_K : M, s \models K_i\varphi \text{ for some } i \in G\}$$

3. COALITIONAL GAMES AND POWER

We briefly review some key concepts from the area of cooperative game theory [10] and the theory of voting power [6] that we will use in the following. A *cooperative* (or *coalitional*) *game* is a pair $\Gamma = \langle Ag, \nu \rangle$, where $Ag = \{1, \ldots, n\}$ is a set of *players*, or *agents*, and $\nu : 2^{Ag} \to \mathbb{R}$ is the *characteristic function* of the game, which assigns to every set of agents a numeric value, which is conventionally interpreted as the value that this group of agents could obtain if they chose to cooperate. A cooperative game is said to be *simple* if the range of ν is $\{0, 1\}$; in simple games we say that G are *winning* if $\nu(G) = 1$, while if $\nu(G) = 0$, we say G are *losing*. A simple cooperative game is said to be *monotonic* if $\nu(G) = 1$ implies that $\nu(H) = 1$, whenever $G \subseteq H$. A monotonic simple cooperative game is sometimes called a *simple voting game* [6]. For simple games, a number of *power indices* attempt to characterise in a systematic way the *influence* that a given agent has, by measuring how effective this agent is at turning a losing coalition into a winning coalition [6]. The best-known of these is perhaps the *Banzhaf index* and its relatives, the Banzhaf score and Banzhaf measure [3].

Agent i is said to be a *swing player* for G if G is not winning but $G \cup \{i\}$ is. We define a function $swing(G, i)$ so that this function returns 1 if i is a swing player for G, and 0 otherwise, i.e.,

$$swing(G, i) = \begin{cases} 1 & \text{if } \nu(G) = 0 \text{ and } \nu(G \cup \{i\}) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Now, we define the *Banzhaf score* for agent i, denoted σ_i, to be the

number of coalitions for which i is a swing player:

$$\sigma_i = \sum_{G \subseteq Ag \setminus \{i\}} swing(G, i). \quad (1)$$

The *Banzhaf measure* μ_i, is the probability that i would be a swing player for a coalition chosen at random from $2^{Ag \setminus \{i\}}$:

$$\mu_i = \frac{\sigma_i}{2^{n-1}} \quad (2)$$

The *Banzhaf index* for a player $i \in Ag$, denoted by β_i, is the proportion of coalitions for which i is a swing to the total number of swings in the game – thus the Banzhaf index is a measure of relative power, since it takes into account the Banzhaf score of other agents:

$$\beta_i = \frac{\sigma_i}{\sum_{j \in Ag} \sigma_j} \quad (3)$$

Finally, we define the *Shapley-Shubik index*; here the *order* in which agents join a coalition plays a role. Let $P(Ag)$ denote the set of all permutations of Ag, with typical members ϖ, ϖ', etc. If $\varpi \in P(Ag)$ and $i \in Ag$, then let $prec(i, \varpi)$ denote the members of Ag that precede i in the ordering ϖ. Given this, let ς_i denote the Shapley-Shubik index of i, defined as follows:

$$\varsigma_i = \frac{1}{|Ag|!} \sum_{\varpi \in P(Ag)} swing(prec(i, \varpi), i) \quad (4)$$

Thus the Shapley-Shubik index is essentially the Shapley value [10, p.291] applied to simple ($\{0, 1\}$-valued) cooperative games.

We say that a player is a *veto player* if it is included in all winning coalitions, a *dictator* if $\mu_i = 1$, and a *dummy* if $\mu_i = 0$.

4. POWER OF DISTRIBUTED KNOWLEDGE

We define the power of agents given a pointed Kripke structure, and an objective specified as a *goal formula*. Intuitively, an agent is maximally powerful if she already knows the goal formula, and is completely powerless if she does not know anything needed in combination with others' knowledge to be able to conclude that the goal formula is true. In between these two extremes are potentially many intermediate levels of power: the more sub-groups the agent can join in order for the group to have shared knowledge of the objective, the more powerful the agent is.

In order to formalise the fact that information about the goal formula is shared in a group, we use the concept of distributed knowledge. We define a simple coalitional game where a coalition is winning iff it has distributed knowledge about the goal formula.

Formally, a *goal structure* is a tuple $S = \langle M, s, \chi \rangle$, where M, s is a pointed model over agents Ag and $\chi \in \mathcal{L}_D$ is a goal formula. Given a goal structure we define the simple game $\langle Ag, \nu_S^D \rangle$:

$$\nu_S^D(G) = \begin{cases} 1 & M, s \models D_G \chi \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE 1. *Figure 1 shows a model M_{MTW} of the situation described in the introduction. Observe that $M_{MTW}, s \models K_M(p \rightarrow q) \wedge K_T(r \rightarrow q) \wedge K_W(p \wedge r)$, and also that these formulae represent "private" knowledge of the respective agents; i.e., we have that $M_{MTW}, s \models \neg K_M(r \rightarrow q) \wedge \neg K_M(p \wedge r) \wedge \neg K_T(p \rightarrow q) \wedge \neg K_T(p \wedge r) \wedge \neg K_W(p \rightarrow q) \wedge \neg K_W(r \rightarrow q)$. Furthermore observe that $M_{MTW}, s \models \neg D_x q$ for all $x \in \{M, T, W\}$, and that $M_{MTW}, s \models \neg D_{\{M,T\}} q \wedge D_{\{M,W\}} q \wedge D_{\{T,W\}} q$. We thus get that M is swing for exactly $\{W\}$, that T is swing for exactly $\{W\}$, that W is swing for exactly $\{M\}$, $\{T\}$ and $\{M, T\}$, and thus that:*

$$\sigma_M = \sigma_T = 1, \sigma_W = 3 \quad \mu_M = \mu_T = \tfrac{1}{4}, \mu_W = \tfrac{3}{4}$$
$$\beta_M = \beta_T = \tfrac{1}{5}, \beta_W = \tfrac{3}{5} \quad \varsigma_M = \varsigma_T = \tfrac{1}{6}, \varsigma_W = \tfrac{2}{3}.$$

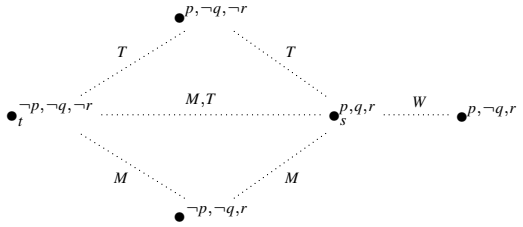


Figure 1: The model M_{MTW}. Reflexive loops are omitted.

What are the properties of ν_S^D? From the fact that $D_G \chi$ implies $D_H \chi$ when $G \subseteq H$ it follows that ν_S^D is always *monotonic*. In fact, monotonicity completely characterise the (simple) games induced in this way: every monotonic (voting) game is induced by some Kripke structure and goal formula via the definition above.

THEOREM 1. *For any simple cooperative game $\Gamma = \langle Ag, \nu \rangle$, there exists a goal structure S such that $\nu_S^D = \nu$ iff Γ is monotonic.*

PROOF. The implication to the right is immediate (as already mentioned), so assume that ν is monotonic. Let $p \in \Theta$. We construct a goal structure $S = \langle M, s, \chi \rangle$ such that $\nu_S^D = \nu$ as follows: $W = \{s_0\} \cup \{s_H : \nu(H) = 0\}$; $s = s_0$; $V(p) = \{s_0\}$; $\chi = p$. $\sim_i$ is defined by the following equivalence classes: $[s_0]_{\sim_i} = \{s_0\} \cup \{s_H : i \in H\}$ and for every H' such that $i \notin H'$, $[s_{H'}]_{\sim_i} = \{s_{H'}\}$. Informally: for each H such that $\nu(H) = 0$ there is a designated state s_H where p is false, which no agent in H can discern from s_0.

Let $\nu(G) = 1$. We must show that $M, s_0 \models D_G p$, so let t be such that $(s_0, t) \in \bigcap_{i \in G} \sim_i$. It suffices to show that $t = s_0$. Assume otherwise: that $t = s_H$ for some H such that $\nu(H) = 0$. For every $i \in G$, $s_0 \sim_i s_H$, and by the definition of $\sim_i$ it follows that $i \in H$. Thus, $G \subseteq H$. But since $\nu(G) = 1$ and $\nu(H) = 0$, that contradicts monotonicity.

Conversely, let $\nu(G) = 0$. We have that $s_0 \sim_i s_G$ for every $i \in G$ and $M, s_G \models \neg p$. Thus $M, s_0 \not\models D_G p$. □

4.1 Expressing Power

Epistemic logic can be used to express and reason about power in Kripke structures. The following expressions can, e.g., be used together with a standard model checker, to determine the power distribution in a given structure.

- i is swing for G when the goal is χ:

$$Swing(G, i, \chi) \equiv \neg D_G \chi \wedge D_{G \cup \{i\}} \chi$$

- The Banzhaf score of i wrt. goal χ is at least k:

$$BAL(i, k, \chi) \equiv \bigvee_{G_1 \neq \cdots \neq G_k \subseteq Ag \setminus \{i\}} \bigwedge_{G \in \{G_1, \ldots, G_k\}} Swing(G, i, \chi)$$

- The Banzhaf score of i wrt. goal χ is k:

$$B(i, k, \chi) \equiv BAL(i, k, \chi) \wedge \neg BAL(i, k+1, \chi)$$

- Of potential interest is checking whether or not one agent has more information/power than another. Note that the maximal Banzhaf score is determined by the maximum number of coalitions not containing the agent; 2^{n-1}. The Banzhaf score of agent i is at least as high as that of agent j:

$$BNoLower(i, j, \chi) \equiv \bigvee_{k \in [0, 2^{n-1}]} BAL(i, k, \chi) \wedge \neg BAL(j, k, \chi)$$

- i is a veto player wrt. goal χ:

$$Veto(i,\chi) \equiv \neg D_{Ag\setminus\{i\}}\chi$$

i is a veto player iff it is included in all winning coalitions, iff all coalitions without i are losing, iff $\neg D_G\chi$ holds for all G without i. By monotonicity this holds iff $Veto(i,\chi)$ holds.

- i is a dictator wrt. goal χ:

$$Dictator(i,\chi) \equiv Veto(i,\chi) \wedge K_i\chi$$

i is a dictator iff all coalitions containing i are winning, and no coalition without i is winning. This holds iff $Dictator(i,\chi)$ holds, by monotonicity.

- i is a dummy wrt. goal χ:

$$Dummy(i,\chi) \equiv \bigwedge_{G\in 2^{Ag}} D_{G\cup\{i\}}\chi \rightarrow D_G\chi$$

i is a dummy iff $\forall G : M,s \models \neg(\neg D_G\chi \wedge D_{G\cup\{i\}}\chi)$ which is equivalent to $\forall G : M,s \models D_{G\cup\{i\}}\chi \rightarrow D_G\chi$.

4.2 Full Communication

Implicit in the idea of information-based power is that groups of agents should somehow be able to *realise* the knowledge distributed among them in order to jointly find out that the goal formula is true. However, while distributed knowledge is the most popular concept in the literature aiming to capture the "sum" of the knowledge in a group, it has the following property, as first pointed out in [13]. It might be that G has distributed knowledge of the goal, but it is still not possible for the group to establish χ through communication in the following sense: it might not be the case that there exists a formula φ_i for each $i \in G$ such that $M,s \models \bigwedge_{i\in G} K_i\varphi_i$ and $\models \bigwedge \varphi_i \rightarrow \chi$. This (possibly lacking) communication property is equivalent [13] to:

$$M,s \models D_G\chi \quad \Rightarrow \quad \bigcup_{i\in G} \mathcal{K}_i(M,s) \models \chi \tag{5}$$

and [13] calls this the *principle of full communication* (the other direction of (5), $\bigcup_{i\in G} \mathcal{K}_i(M,s) \models \chi \Rightarrow M,s \models D_G\chi$, holds on any model). As an example, consider the model M_1 in Figure 2. In this model p is distributed knowledge among agents 1 and 2 in state s, but p is not entailed from the individual knowledge of 1 and 2 in s and the model does not satisfy the principle of full communication.

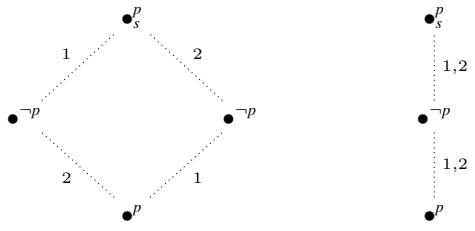


Figure 2: Models M_1 (left) and M_2 (right). Reflexive and transitive edges omitted.

So, if we take the p as the goal formula, agent 1 is swing for $\{2\}$ in state s in the model M_1 above, but it is not possible for agents 1 and 2 to actually infer p together by communicating using the epistemic language. Our information-based power measures make particular sense in models that satisfy the principle of full communication, because in such models whatever is distributed knowledge can be obtained by communication in the sense that it follows from individual knowledge that the involved agents can specify and communicate as logical formulas. So which models satisfy the principle of full communication? There are two particularly relevant model properties here (generalisations of propositions given in [13]). A model $M = \langle W, \sim_1, \ldots, \sim_n, \pi\rangle$ is a:

- *full model* [7] iff for all $s \in W$, $G \subseteq Ag$, and $\Phi \subseteq \mathcal{L}_D$: if $\Phi \cup \mathcal{K}_G(M,s)$ is satisfiable then Φ is satisfiable in $\{t : (s,t) \in \sim_G^D\}$.
- *full communication model* [11] iff for all $s \in W$, $G \subseteq Ag$, and $\varphi \in \mathcal{L}_K$: if $\{\varphi\} \cup \mathcal{K}_G(M,s)$ is satisfiable then φ is satisfiable in $\{t : (s,t) \in \sim_G^D\}$.

Clearly, full models are full communication models. [7] shows that fullness is sufficient for the principle of full communication to hold, while [11] shows that a model satisfies the principle of full communication *if and only if* the model is a full communication model.

While this definition of full communication models may seem somewhat technical, note that the principle of full communication is often violated by the existence of bisimilar states in the model (such as in the model above). Indeed, bisimulation contractions of finite models are full communication models (they are *distinguishing* in the sense of [13], due to the existence of characteristic formulae). Models that are finite and do not contain bisimilar states (and thus are their own bisimulation contractions) are very common.

Thus, on full communication models we get an alternative, equivalent, definition of power. We have that:

$$\nu_S^D(G) = 1 \quad \Leftrightarrow \quad \bigcup_{i\in G} \mathcal{K}_i(M,s) \models \chi \tag{6}$$

when M is a full communication model.

4.3 Properties of Power

The relationship between power properties and epistemic properties is of natural interest, not the least in order to validate that our definition of power is reasonable. The relationship properties in the following lemma are discussed below.

LEMMA 1. *Let the goal structure $S = \langle M,s,\chi\rangle$ be given.*

1. *If $M,s \models \neg D_{Ag}\chi$, then $x_i = 0$ for all i and $x \in \{\sigma,\mu,\beta,\varsigma\}$.*
2. *If $M,s \models \neg\chi$, then $x_i = 0$ for all i and $x \in \{\sigma,\mu,\beta,\varsigma\}$.*
3. *If $M,s \models K_i\chi$, then $x_i \geq x_j$ for all j and $x \in \{\sigma,\mu,\beta,\varsigma\}$.*
4. *If $M,s \models \neg D_{Ag\setminus\{i\}}\chi$, $x_i \geq x_j$ for all j and $x \in \{\sigma,\mu,\beta,\varsigma\}$.*
5. *If $M,s \models K_i\chi \wedge \neg K_j\chi$, then $x_i > x_j$ for all $x \in \{\sigma,\mu,\beta,\varsigma\}$.*
6. *On full communication models, if $\mathcal{K}_i(M,s) \subseteq \mathcal{K}_j(M,s)$ then $x_i \leq x_j$, for any power measure $x \in \{\sigma,\mu,\beta,\varsigma\}$.*

The first property says that if not enough information to infer the goal formula is distributed throughout the complete system, then every agent has *no power*. The second property is a special case of the first – the goal *cannot* be derived because it is not true. The third and fourth properties represent the other extreme: *maximum power*. The agent has maximum power (at least as much power as anyone else) if she already knows the goal, or if the rest of the system does not have enough information to derive the goal (i.e, if the agent is a veto player). The fifth and sixth properties are about *relative power*. The fifth says that an agent who already knows χ is always strictly more powerful than an agent who does not know χ. The sixth property says that if one agent knows at least as much

as another agent, then the first agent is at least as powerful. This relates our definition of power to a more classical notion of "knowing more" in a reasonable way. Our notion is more fine grained; the implication does not hold in the other direction. The sixth property holds for full communication models, which, again, is a natural class of models in which to interpret our power measures since they come with a natural mechanism for distribution of information.

PROOF (OF LEMMA 1). 1. Follows immediately from monotonicity: if i is swing for G, then $M,s \models D_{G\cup\{i\}}\chi$.

2. Immediate from $\models \neg\chi \rightarrow D_{Ag}\neg\chi$ and the first item.

3. If suffices to show that i is swing for any coalition any agent j is swing for. So assume that $M,s \models \neg D_G\chi \wedge D_{G\cup\{j\}}\chi$. From $M,s \models K_i\chi$ it follows that $M,s \models D_{G\cup\{i\}}\chi$, and thus i is also swing for G.

4. Assume that j is swing for G. From $M,s \models D_{G\cup\{j\}}\chi$, the assumption that $M,s \models \neg D_{Ag\setminus\{i\}}\chi$ and monotonicity, it follows that $i \in G$. Thus it also follows that i is swing for $(G\setminus\{i\})\cup\{j\}$. Because $i \in G$ and $j \notin G$ for coalitions G for which j is swing, $(G_1\setminus\{i\})\cup\{j\} \neq (G_2\setminus\{i\})\cup\{j\}$ for any two different coalitions G_1, G_2 for which j is swing, and thus there are at least as many swings for i.

5. If j is swing for G, $M,s \models \neg D_G\varphi$ so G cannot contain i and i is also swing for G. In addition, i is swing for $\emptyset$, unlike j.

6. Let M be a full communication model and assume that i is swing for G, i.e., that $M,s \models D_G\chi \wedge D_{G\cup\{i\}}\chi$. From the fact that M is a full communication model and eq. (6) above, we get that $\bigcup_{l\in G\cup\{i\}} \mathcal{K}_l(M,s) \models \chi$. From $\mathcal{K}_i(M,s) \subset \mathcal{K}_j(M,s)$ it follows that $\bigcup_{l\in G\cup\{j\}} \mathcal{K}_k(M,s) \models \chi$ which again means that $M,s \models D_{G\cup\{j\}}\chi$. Thus, j is swing for G. □

In the following lemma we look at power measures in "similar" models. The proper notion of bisimulation for distributed knowledge, and hence our power measures, is given in the second point.

LEMMA 2.

1. *The power measures are not invariant under (standard) bisimulation. That is, bisimilar pointed models may have different power measures.*
2. *The power measures are invariant under* collective bisimulation *[11].*
3. *On full models, the power measures are invariant under (standard) bisimulation.*

PROOF. 1. A counter-example is found in Figure 2, which contains two bisimilar models with two agents. It is easy to see that by taking $\chi = p$, we get $\sigma_1 = 1$ in M_1 but $\sigma_1 = 0$ in M_2.

2. follows immediately from the fact that satisfaction in $\mathcal{L}_D$ is invariant under collective bisimulation [11, Prop. 19].

3. For full models the notions of collective bisimulation and bisimulation coincide [11, Prop. 20]. □

Finally, let us look at the relationship between power properties and the structure of the goal formula. We will make use of the logical expressions of power properties from Section 4.1.

Starting with tautologies and contradictions:

$$\begin{array}{ll} \models \neg Swing(G,i,\top) & \models \neg Swing(G,i,\bot) \\ \models Veto(i,\bot) & \models \neg Veto(i,\top) \\ \models \neg Dictator(i,\bot) & \models \neg Dictator(i,\top) \end{array}$$

With such goal formulae, no agent can be swing for any coalition. Every agent is a veto player for $\bot$, while no agent is a veto player for $\top$. No agent can be a dictator for $\bot$ nor $\top$.

The case of conjunction:

$$\models (Swing(G,i,\chi_1) \wedge Swing(G,i,\chi_2)) \rightarrow Swing(G,i,\chi_1 \wedge \chi_2)$$

Swings are closed under the operation of taking conjunction of goal formulae. The converse does not hold, but this does:

$$\models Swing(G,i,\chi_1 \wedge \chi_2) \rightarrow (Swing(G,i,\chi_1) \vee Swing(G,i,\chi_2))$$

– if i is swing wrt. a conjunction, she is swing wrt. at least one of the conjuncts (but not necessarily both).

For negation we have that (but not the other way around):

$$\models Swing(G,i,\neg\chi) \rightarrow \neg Swing(G,i,\chi)$$

Moving on to the case that the goal formula is epistemic, first observe the following properties of distributed $S5$ knowledge: $\models D_GD_{G'}\varphi \rightarrow D_G\varphi$ for any G, G', and $\models D_GD_{G'}\varphi \leftrightarrow D_G\varphi$ when $G \subseteq G'$. From these properties it follows that:

$$\begin{array}{ll} \models Swing(H,i,D_G\chi) \rightarrow Swing(H,i,\chi) & \text{when } H \subseteq G \\ \models Swing(H,i,D_G\chi) \leftrightarrow Swing(H,i,\chi) & \text{when } H\cup\{i\} \subseteq G \end{array}$$

In particular, using a goal formula $D_G\varphi$ is equivalent to using φ when it comes to counting swings within G.

If we take $G = \{j\}$ in the expressions above, we get the case where the goal formula describes individual knowledge. It follows that:

$$\begin{array}{ll} \models Swing(\emptyset,i,K_j\chi) \rightarrow Swing(\emptyset,i,\chi) & \text{for any } j \\ \models Swing(\{j\},i,K_j\chi) \rightarrow Swing(\{j\},i,\chi) & \text{for any } j \\ \models Swing(\emptyset,i,K_i\chi) \leftrightarrow Swing(\emptyset,i,\chi) & \end{array}$$

5. KNOWLEDGE OF POWER

We have thus associated power indices with states of Kripke structures, by assuming that they are defined by agents' knowledge. But epistemic logic allows us to reason about agents' knowledge *about* state-properties – so we can go from analysing the power of knowledge to analysing knowledge of power: what do the agents in the system know about the distribution of power?

The formula $K_jSwing(G,i,\chi)$, where $Swing(G,i,\chi) = \neg D_G\chi \wedge D_{G\cup\{i\}}\chi$, denotes the fact that agent j knows that i is swing for G. If we look first at the more general case of *distributed* knowledge of that fact, we have the following (we formally prove this and the following validities in Theorem 2 below):

$$\models Swing(G,i,\chi) \rightarrow D_{G\cup\{i\}}Swing(G,i,\chi) \tag{7}$$

– if i is swing for G, then this is distributed knowledge in $G\cup\{i\}$.

However, this does not carry over to individual knowledge. It turns out that $Swing(G,i,\chi) \wedge \neg K_jSwing(G,i,\chi)$ is satisfiable, for any j including $j = i$. Thus, an agent can be swing for a coalition, without neither the agent nor the agents in the coalition knowing it. When, then, *does* an agent know that she is swing? The answer is: *almost never*. The following holds:

$$\models K_j\neg Dummy(i,\chi) \rightarrow K_j\chi \tag{8}$$

for any i,j (including $i = j$). In other words, an agent can only know that any agent (including herself) is swing for any coalition if she (the first agent) already knows the goal formula! In the typical case that χ is distributed information throughout the system, but no individual agent alone knows χ, *no* agent knows that *any* agent can swing *any* coalition from ignorance to knowledge about χ. It follows that

$$\models K_j\neg Dummy(i,\chi) \rightarrow K_j \bigwedge_{k\in Ag} BNoLower(j,k,\chi) \tag{9}$$

– only agents that are maximally powerful (at least as powerful as any other agent), and know that they are, can know that anyone (including themselves) are not a dummy player.

It also holds that

$$\models K_j Swing(G,i,\chi) \rightarrow K_j Swing(G,j,\chi) \quad (10)$$

– if an agent knows that another agent is swing for some coalition, then the first agent must be swing for the same coalition. In particular: $\models K_j \neg Dummy(i,\chi) \rightarrow K_j \neg Dummy(j,\chi)$.

However, *no* agent *in* a coalition can know that someone is swing for that coalition:

$$\models \bigwedge_{j \in G} \neg K_j Swing(G,i,\chi) \quad (11)$$

For veto players, we have that

$$\models K_i Veto(j,\chi) \rightarrow \neg K_i \neg Dummy(i,\chi) \quad i \neq j \quad (12)$$

– the only agents that can know that someone else is a veto player are agents that consider it possible that they are dummies themselves.

For dictators, we have that

$$\models \neg K_j Dictator(i,\chi) \quad i \neq j \quad (13)$$

– the only agent that can know who the dictator is, is the dictator.

Turning to knowledge about the values of power indices, we have

$$\models K_j B(i,k,\chi) \rightarrow BNoLower(j,i,\chi) \quad (14)$$

– no agent can know the Banzhaf score of any agent with a lower score than herself.

We can conclude that the distribution of power is generally not known *in* the system. We emphasise that this does not pose any problem for our interpretation of the power indices as measures of the distribution of information in the system, as we discuss further in Section 7.

THEOREM 2. *Properties (7)–(14) hold.*

PROOF. We make use of the fact that distributed knowledge satisfies the $S5$ properties [4], which follows from the fact that the intersection of equivalence relations is an equivalence relation, as well as the monotonicity property ($D_G\varphi \rightarrow D_H\varphi$ when $G \subseteq H$).

(7): from $\neg D_G\chi$ it follows that $D_G \neg D_G\chi$ by negative introspection, and $D_{G\cup\{i\}} \neg D_G\chi$ follows by monotonicity. $D_{G\cup\{i\}} D_{G\cup\{i\}}\chi$ follows from $D_{G\cup\{i\}}\chi$ by positive introspection. $D_{G\cup\{i\}} Swing(G,i)$ follows by knowledge distribution.

(8): $K_j \neg Dummy(i,\chi)$ is equal to $K_j \bigvee_G (D_{G\cup\{i\}}\chi \wedge \neg D_G\chi)$. By reflexivity $D_{G\cup\{i\}}\chi$ implies χ, and thus $\bigvee_G (D_{G\cup\{i\}}\chi \wedge \neg D_G\chi)$ implies that χ. By knowledge distribution, $K_j\chi$ holds.

(9): let $K_j \neg Dummy(i,\chi)$ be true. By (8), $K_j\chi$ and from positive introspection $K_jK_j\chi$. From Lemma 1.3 it follows that $K_j BNoLower(j,k,\chi)$ for any k.

(10): from $K_j Swing(G,i,\chi)$ it follows that $K_j \neg D_G\chi$. By (8) it also follows that $K_j\chi$. By knowledge distribution, $K_j(\neg D_G\chi \wedge K_j\chi)$, which by monotonicity implies that $K_j(\neg D_G\chi \wedge D_{G\cup\{j\}}\chi)$.

(11): if $K_j Swing(G,i,\chi)$ is true for some $j \in G$, then $K_j Swing(G,j,\chi)$ by (10), and $Swing(G,j,\chi)$ by reflexivity. But this is a contradiction.

(12): from $K_i Veto(j,\chi)$ it follows that $K_i \neg K_i\chi$ when $i \neq j$, from which it follows that $\neg K_i\chi$. If $K_i \neg Dummy(i,\chi)$ is true, then $K_i\chi$ by (8); a contradiction.

(13): $K_j Dictator(i,\chi)$ is equivalent to $K_j(Veto(i,\chi) \wedge K_i\chi)$, which implies that $K_j\chi$ and $Veto(i,\chi)$. From the latter it follows that $\neg D_{Ag\setminus\{i\}}\chi$, and from monotonicity it follows that $\neg K_j\chi$ – a contradiction.

(14): if $\sigma_i = 0$, the formula holds trivially. If $\sigma_i > 0$, $K_jB(i,k,\chi)$ implies that there is a G such that $K_j(\neg D_G\chi \wedge D_{G\cup\{i\}}\chi)$ is true. It follows that $K_j\chi$, and by Lemma 1.3 that $\sigma_j \geq \sigma_i$. □

6. OTHER TYPES OF GROUP KNOWLEDGE

We have so far used the notion of distributed knowledge to measure power. Can other notions of group knowledge be used? Note that both everybody-knows and common knowledge are anti-monotonic, in the sense that $C_G\varphi$ implies $C_{G'}\varphi$ when $G' \subseteq G$, while distributed knowledge is monotonic ($D_{G'}\varphi$ implies $D_G\varphi$). This means that simply "replacing" distributed knowledge in the definition of the game by any of these notions would not make sense (e.g., $\neg C_G\varphi \wedge C_{G\cup\{i\}}\varphi$ is not satisfiable). However, there is another way in which we can look at an agent's power with respect to common knowledge (and similarly with everybody-knows). An agent has "negative" power if he can swing a coalition from *having* common knowledge of the goal, to *not* having it. In other words, this would correspond to an agent's power to spoil, rather than to achieve, the goal. Using this definition of the power measures, a high value means that the agent has *little* information, and including it in a group is likely to, e.g., break common knowledge needed for coordination.

Let us start with everybody-knows. Given $S = \langle M,s,\chi\rangle$, let:

$$\nu_S^E(G) = \begin{cases} 1 & M,s \models \neg E_G\chi \\ 0 & \text{otherwise} \end{cases}$$

We say that a simple cooperative game is *determined* if there is a set of agents $Winners \subseteq Ag$ such that $\nu(G) = 1$ iff $G \cap Winners \neq \emptyset$. Note that determined games are monotonic.

THEOREM 3. *For any simple cooperative game $\Gamma = \langle Ag, v\rangle$, there exists a goal structure S such that $\nu_S^E = \nu$ iff Γ is determined.*

PROOF. For the implication to the right, given S let $Winners = \{i : M,s \models \neg K_i\chi\}$. It is easy to see that $\nu_S^E(G) = 1$ iff $G \cap Winners \neq \emptyset$. For the implication to the right, we define $S = \langle M,s,\chi\rangle$ as follows. Let $p \in \Theta$. Let $W = \{s,t\}$; $s_0 = s$; $V(p) = \{s\}$, $V(q) = \emptyset$ for $q \neq p$; $s \sim_i t \Leftrightarrow i \in Winners$; $\chi = p$. Let $\nu(G) = 1$. That means that there is an agent i such that $i \in G \cap Winners$. From $i \in Winners$ it follows that $M,s_0 \models \neg K_ip$, and since $i \in G$ we get that $M,s_0 \models \neg E_G\chi$. For the other direction, let $M,s_0 \models \neg E_Gp$. That means that $M,s_0 \models \neg K_ip$ for some $i \in G$. But the only possibility then is that also $i \in Winners$. Thus, $i \in G \cap Winners$, and thus $\nu(G) = 1$. □

It is easy to see that for determined games, the Banzhaf score is the same for all winners, as well as the same (0) for all non-winners:

LEMMA 3. *For any determined game and any agent i,*

$$\sigma_i = \begin{cases} 2^{|Ag\setminus Winners|} & i \in Winners \\ 0 & otherwise \end{cases}$$

It follows that it is easy to compute the power measures:

THEOREM 4. *Given a goal structure $S = \langle M,s,\chi\rangle$ and an agent i in M, the Banzhaf score σ_i for i in the game $\langle Ag, \nu_S^E\rangle$ can be computed in polynomial time.*

PROOF. By Theorem 3 the game is determined. The winners can be computed in polynomial time: for every state t, check whether $M,t \models \neg\chi$, and if it does add i to $Winners$ if there is an i-transition from t to s. σ_i is computed from the size of $Winners$ according to Lemma 3. □

Moving on to common knowledge, given $S = \langle M, s, \chi \rangle$, let:

$$\nu_S^C(G) = \begin{cases} 1 & M, s \models \neg C_G \chi \\ 0 & \text{otherwise} \end{cases}$$

EXAMPLE 2. *The following two examples are inspired by [14, Section 2.3]. In the first setting, the set of agents Ag is the set of participants of a conference, and $a \in Ag$ represents our hero Alco. During one afternoon, while all other participants are attending a joint session, Alco spends his time in the bar of the conference hotel. The session chair announces χ: 'tomorrow, sessions start at 9:00 rather than 10:00'. Everybody (i.e., Ag) at the conference feels very responsible for the well-being of the participants, and only if $C_{Ag}\chi$ holds, people will stop informing each other of χ. If s is the situation immediately after the chair's announcement, we obviously have $M, s \models Swing(Ag \setminus \{a\}, a, \chi)$, where Swing is now defined for common knowledge: $Swing(G, i, \chi) = C_G\chi \wedge \neg C_{G\cup\{i\}}\chi$. Now consider a new state s_1, in which Alco leaves the bar to get some fresh air, and which leads to a state s_2 where at the general session a friend f of Alco makes the chair (publicly) aware that Alco was in the bar during the announcement χ. At this moment it is common knowledge among $Ag \setminus \{a\}$ that $Swing(Ag \setminus \{a\}, a, \chi)$, but then the chair replies to f by saying that there is an intercom in the bar that is directly connected to the conference room. Note that a is now still a veto player wrt. Ag and χ, since Alco does not know about the discussion regarding his absence during the announcement of χ. In other words, although in s_2 we have $E_{Ag}\chi$, we also have $\neg K_a K_f K_a \chi$: Alco knows that his friend f may have concerns about Alco not knowing χ (this concern is justified, since f notified the chair), and Alco does not know that f has been properly informed (that $K_a\chi$) by the chair, so one may expect that a will make at some time an effort to make publicly known that he knows χ, so people can stop worrying about $a's$ time-table tomorrow.*

Swing players for common knowledge in a coalition G often come with delicate protocols for the communication in G. An example here is the celebrations of Santa Claus in certain cultures, where it is common knowledge among those over a certain age that Santa Claus is in fact not responsible for the presents at the evening (this is χ), while χ is not known among the participants under a certain age. Now, even when everybody at the Christmas party knows that χ, there may be several swing players for several coalitions, which explains that conversations have to be participated in carefully. To be more precise, suppose that $E_G E_G \chi \wedge \neg K_i K_j K_i \chi$ (with $i, j \in G$). Since i knows that everybody in G knows χ already, he might chose not to look childish to j and reveal to j that $K_i\chi$, indicating he is not a fool. But i might also chose to exploit $\neg K_i K_j K_i \chi$, and challenge j into a 'dangerous conversation', where j may think he needs to be careful not to reveal χ to i.

These examples also suggest that power is in fact an interesting issue in dynamic contexts, after enough communication has taken place for instance, Alco may seize to be a swing player. Dynamic Epistemic Logic ([14]) paves the right formal framework to study these phenomena, like the fact that some true formulas can never be known no matter how often they are announced: they would always have a veto player (Moore sentences like $(p \wedge \neg K_a p)$ being the most prominent examples).

Like for the case of distributed knowledge, the class of games obtained in this way is exactly the monotonic games.

THEOREM 5. *For any simple cooperative game $\Gamma = \langle Ag, \nu \rangle$, there exists a goal structure S such that $\nu_S^C = \nu$ iff Γ is monotonic.*

PROOF. It is easy to see that ν_S^C is monotonic.

For the other direction, let ν be monotonic. If there is no coalition G with $\nu(G) = 1$, let M consist of only one state s with $V(p) = \{s\}$ and $\sim_a = W \times W$ for every $a \in Ag$. It is easily seen that $\nu^C_{M,s,p}(G) = 0$ for all coalitions G.

Otherwise put first of all $s \in W \cap V(p)$ and add (s, s) to each $\sim_a$. Let $H_1, \ldots H_k$ be the coalitions with the property that $\nu(H_i) = 1$ and for no proper subset of H_i, it holds that $\nu(H) = 1$. For each such H_i, do the following. Let $H_i = \{a^i_1, a^i_2, \ldots a^i_{m(i)}\}$. Add new states $W_i = \{s^i_1, s^i_2, \ldots s^i_{m(i)}\}$ to W in such a way that (s, s^i_1) and (s^i_1, s) become members of $\sim_{a^i_1}$ and furthermore add (s^i_j, s^i_{j+1}), (s^i_{j+1}, s^i_j) to $\sim_{a^i_{j+1}}$ with $1 \leq j < m(i)$. Add (s^i_j, s^i_j) to each $\sim_a$ $(1 \leq m(i))$. Finally, add $W_i \setminus \{s^i_{m(i)}\}$ to $V(p)$. When this process has finished for all H_i, take the transitive symmetric reflexive closure of every $\sim_a$ so far defined. The effect of this last step is that for every agent a and every two states s^i_1 and s^j_1 with (s, s^i_1) and $(s, s^j_1) \in \sim_a$, we also add (s^i_1, s^j_1) and (s^j_1, s^i_1) to $\sim_a$.

A straight path π in the model is a sequence of state-agent alterations $\langle x_1, a_1, x_2, a_2 \ldots x_n \rangle$, with each $x_i \in W, a_i \in Ag$, and $(x_i, x_{i+1}) \in \sim_{a_i}$ such that $x_i \neq x_j$ if $i \neq j$. It is a straight s-path if $x_1 = s$. Let $Ag(\pi)$ be the set of agents occurring in π. Note that a straight s-path that ends in state s_n denotes a 'shortest' route in the model from s to s_n, since the states in a straight path are different. A straight path $x_1, a_1, x_2, a_2 \ldots x_n$ leads to φ if x_n is the only-φ world in it. The following is an important property of our model: there is a straight path π leading to $\neg p$ iff for some H_i, we have $\nu(H_i) = 1$ and $Ag(\pi) = H_i$.

We now prove that $\forall G \subseteq Ag$ $(\nu(G) = 1$ iff $M, s \models \neg C_G p)$. First, if $\nu(G) = 1$, there is a smallest set $H_i = \{a^i_1, \ldots, a^i_{m(i)}\} \subseteq G$ such that $\nu(H_i) = 1$. For this H_i, we have constructed a straight s-path π leading to $\neg p$ and for which $Ag(\pi) = H_i$. So, we have $M, s \models \neg C_{H_i} p$, and hence $M, s \models \neg C_G p$, i.e., $\nu_S^C(G) = 1$. Secondly, suppose $M, s \models \neg C_G p$, it means for our model that there is a straight s-path π leading to $\neg p$ for which $Ag(\pi) \subseteq G$ (indeed, there may be agents $a \in G \setminus Ag(\pi)$). But the only such paths we have in M are paths that use a minimal set of agents H_i for which $\nu(H_i) = 1$, so $\nu(Ag(\pi)) = 1$. By monotonicity, $\nu(G) = 1$. □

7. DISCUSSION

We have shown that our information-based notion of power has reasonable properties, at least on full communication models – which come with a natural mechanism for distribution of information. We have also shown that it is easy to compute such power indices using a standard model checker for epistemic logic.

It is natural to define swings using distributed knowledge. A high power index here means that the agent's knowledge is important for an arbitrary group jointly getting to know the goal formula by sharing their information. We also gave alternative definitions of "negative" power in terms of swinging a group from a situation where every member knows the goal, or the goal is common knowledge. Here, a high power index means that the agent knows little: if it is important to have common knowledge in a group (e.g., for coordination), then it is likely that including a high-power agent will lead to failure. The everybody-knows case is computationally tractable, but the price is a lower "resolution": the agents divide into only two classes, with agents in the same class having the same power. It is interesting that the common knowledge case and the distributed knowledge correspond to the same class of voting games (Theorems 1 and 5). If this seems counter-intuitive, keep in mind that the two theorems express that there is a connection between distributed knowledge and *lack* of common knowledge: conceiving distributed knowledge as a game where a coalition wins if it implicitly knows the goal formula, is structurally similar to con-

ceiving common knowledge as a game where a coalition wins if it does *not* commonly know the goal.

[15] studies a particular notion of "knowing more". Their concept "i knows at least what j knows" is defined by $R_i(s) \subseteq R_j(s)$ where s is a state and $R_x(s) = \{t : (s,t) \in R_x\}$ and R_x is an indistinguishability relation for agent x. Our power measures for distributed knowledge agree: if $R_i(s) \subseteq R_j(s)$ then $Swing(G,j,\chi)$ implies that $Swing(G,i,\chi)$ for any χ, and thus $\sigma_i \geq \sigma_j$. The implication does not hold in the other direction; our notion of "knowing more" is more fine grained. [15] also introduces a modal operator $\succeq$ where, for agents i and j, the formula $i \succeq j$ expresses that whatever state is an alternative for j, is also an alternative for i. This provides a way to locally express that $K_i\varphi \rightarrow K_j\varphi$ for all φ. There is one sense in which such an operator allows one also to express properties of the power of knowledge in a compact way. For distributed knowledge for instance, the formula $i \succeq j$ implies that $(Swing(G,i,\chi) \rightarrow Swing(G,j,\chi))$ and $\neg Swing(G \cup \{i\},j,\chi)$ – for *any* χ. When reasoning about the power in the context of everybody knows, "opposite" properties derive: $\models (i \succeq j) \rightarrow (Swing(G,j,\chi) \rightarrow Swing(G,i,\chi))$ and $\models (i \succeq j) \rightarrow \neg Swing(G \cup \{j\},i,\chi)$. Note that such properties cannot be expressed in modal logic without such an operator: for instance in $\models (K_i\varphi \rightarrow K_j\varphi) \rightarrow (Swing(G,i,\chi) \rightarrow Swing(G,j,\chi))$ the formula φ is a specific formula (not a scheme), and $\models (K_i\varphi \rightarrow K_j\varphi) \Rightarrow \models (Swing(G,i,\chi) \rightarrow Swing(G,j,\chi))$ is obviously true, but much weaker: the antecedent is false (if $i \neq j$).

In Section 5 we saw that agents *in* the system generally know very little about the distribution of information-based power in the system. For example, an agent with a high power index typically does not know which coalitions she needs to join in order to derive the goal formula (or indeed *that* she is a high-power agent). We emphasise that this is not in any way a problem for the interpretation of our power indices. A high Banzhaf index means, in our setting, that the probability of changing some arbitrary coalition from ignorance to knowledge about the goal is high – in the same way that it is interpreted as the probability of changing an outcome in voting theory. In fact, that an agent does not know which coalitions it is swing for makes the probability of being swing for an *arbitrary* coalition more interesting. Furthermore, in many distributed and multi-agent systems, such as sensor networks, agents are restricted to communication with some arbitrary sub-group of all agents at any given time. We think of these power measures as a tool for external analysis of the information distribution in a system, to find out, e.g., whether information is evenly distributed or whether there are some agents that are particularly crucial to the functioning of the system in the sense that the information they have is difficult to obtain elsewhere in the system. The negative results about knowledge of power properties can also be seen as a *barrier against strategic behaviour*: it is almost never possible for an agent to know that it suffices to share information with only some particular subgroup of the grand coalition.

An interesting direction for future work is to associate formulae of the form $D_G D_H \varphi$ with *composite* voting games [6, p. 27]. In this paper we have studied a semantic notion of power, associated with a point in a Kripke structure. Another direction for future work is to develop a *syntactic* notion of power, based on a set of epistemic formulae. For such an approach it would be necessary to syntactically describe that agents know "this and nothing more", and extensions of epistemic logic with *only knowing* [9] seem like a promising starting point.

Acknowledgments

We thank the AAMAS program committee and Pål Grønås Drange for comments that helped us improve the paper.

8. REFERENCES

[1] T. Ågotnes, W. van der Hoek, J. A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In M. M. Veloso, editor, *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 1175–1180, California, 2007. AAAI Press.

[2] T. Ågotnes, W. van der Hoek, M. Tennenholtz, and M. Wooldridge. Power in normative systems. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 145–152, Budapest, Hungary, May 2009. IFAMAAS.

[3] J. F. Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press: Cambridge, England, 2001.

[5] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. The MIT Press: Cambridge, MA, 1995.

[6] D. S. Felsenthal and M. Machover. *The Measurement of Voting Power*. Edward Elgar: Cheltenham, UK, 1998.

[7] J. Gerbrandy. *Bisimulations on Planet Kripke*. Ph.D. thesis, University of Amsterdam, 1999.

[8] A. Hunter and S. Konieczny. On the measure of conflicts: Shapley inconsistency values. *Artificial Intelligence*, 174(14):1007 – 1026, 2010.

[9] H. J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42:263–309, 1990.

[10] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press: Cambridge, MA, 1994.

[11] F. Roelofsen. Distributed knowledge. *Journal of Applied Non-Classical Logics*, 17(2):255–273, 2007.

[12] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Diego, CA, 1992.

[13] W. van der Hoek, B. van Linder, and J.-J. Meyer. Group knowledge is not always distributed (neither is it always implicit). *Mathematical Social Sciences*, 38:215–240, 1999.

[14] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, Berlin, 2007.

[15] H. Van Ditmarsch, W. Van Der Hoek, and B. Kooi. Knowing more: from global to local correspondence. In *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pages 955–960, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

Tractable Model Checking for Fragments of Higher-Order Coalition Logic*

Patrick Doherty
Dept. of Computer and Information Science, Linköping University, Sweden
patrick.doherty@liu.se

Barbara Dunin-Kęplicz
Institute of Informatics, Warsaw University, Poland
and Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
keplicz@mimuw.edu.pl

Andrzej Szałas
Institute of Informatics, Warsaw University Warsaw, Poland
and Dept. of Computer and Information Science, Linköping University, Sweden
andrzej.szalas@{mimuw.edu.pl,liu.se}

ABSTRACT

A number of popular logical formalisms for representing and reasoning about the abilities of teams or coalitions of agents have been proposed beginning with the Coalition Logic (CL) of Pauly. Ågotnes et al introduced a means of succinctly expressing quantification over coalitions without compromising the computational complexity of model checking in CL by introducing Quantified Coalition Logic (QCL). QCL introduces a separate logical language for characterizing coalitions in the modal operators used in QCL. Boella et al, increased the representational expressibility of such formalisms by introducing Higher-Order Coalition Logic (HCL), a monadic second-order logic with special set grouping operators. Tractable fragments of HCL suitable for efficient model checking have yet to be identified. In this paper, we relax the monadic restriction used in HCL and restrict ourselves to the diamond operator. We show how formulas using the diamond operator are logically equivalent to second-order formulas. This permits us to isolate and define well-behaved expressive fragments of second-order logic amenable to model-checking in PTIME. To do this, we appeal to techniques used in deductive databases and quantifier elimination. In addition, we take advantage of the monotonicity of the effectivity function resulting in exponentially more succinct representation of models. The net result is identification of highly expressible fragments of a generalized HCL where model checking can be done efficiently in PTIME.

Categories and Subject Descriptors

F.4 [**Mathematical Logic And Formal Languages**]: Miscellaneous

General Terms

Theory, Verification

*This work is partially supported by grants from the ELLIIT Excellence Center at Linköping-Lund in Information Technology, the Swedish Research Council (VR) Linnaeus Center CADICS, VR grant 90385701, NFFP5-The Swedish National Aviation Engineering Research Programme and grant N N206 399334 from the Polish MNiSW.

Cite as: Tractable Model Checking for Fragments of Higher-Order Coalition Logic, Patrick Doherty, Barbara Dunin-Kęplicz and Andrzej Szałas, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 743-750.

Keywords

Coalition Formation, Coalition Logic, Model Checking, Complexity

1. INTRODUCTION

In recent years, developing formal techniques for representing and reasoning about the abilities of teams or coalitions has become a major focus of research in the areas of artificial intelligence and multiagent systems [2–10, 12, 16, 17, 19, 21–29]. In particular, combining ideas and techniques from game theory, logic and social theory has become highly relevant due to the widespread use of social software and trends in robotics and agent systems where cooperation among such agents is becoming increasingly important. A number of popular logical formalisms for representing and reasoning about the abilities of teams or coalitions of agents have been proposed beginning with the Coalition Logic (CL) of Pauly [19] which is a propositional multimodal logic. Recent trends in development of logical formalisms for reasoning about coalitions have tried to increase expressivity of such formalisms while retaining tractability in the reasoning components associated with such formalisms. For instance, Ågotnes et al [3] introduce a means of succinctly expressing quantification over coalitions without compromising the computational complexity of model checking in CL by introducing Quantified Coalition Logic (QCL). More recently, Boella et al [7], increased the representational expressibility of such formalisms by introducing Higher-Order Coalition Logic (HCL), a monadic second-order logic with special set grouping operators. HCL subsumes both CL and QCL representationally, and includes a sound and complete axiomatization for weakly playable frames, but currently lacks a tractable reasoning component.

Due to the modal nature of many of these formalisms which are based on the use of effectivity functions as part of coalition frames, the major computational problem has been that of model checking.

> Given a *succinct* representation of a model $\mathcal{M}$, a state s, and a formula φ of your favorite coalition logic $\mathcal{L}$, is it the case that $\mathcal{M}, s \models_{\mathcal{L}} \varphi$?

In addition to the representational problem, research focus associated with the reasoning problem has been placed on finding succinct representations of models in $\mathcal{L}$, extending the expressivity of formulas φ and trying to guarantee tractability of the model checking problem for the full language or its fragments used in $\mathcal{L}$.

Higher-order logic is particularly suited as a representation language for modeling the abilities and interactions between coalitions. It is representationally expedient in the sense that coalitions are in fact sets of agents and one wants to represent such sets and their properties in as direct a way as possible. This of course can be done directly and succinctly in higher-order logic. Boella et al [7] provide very convincing arguments in this respect. On the other hand, reasoning in higher-order logic is more problematic. Yet there are well-behaved fragments of such logics that deserve investigation and surprisingly, one can isolate fragments which are representationally expressive and also computationally tractable.

This is both the focus and contribution of this paper. Our contribution is to propose a higher-order logic HCL* which essentially subsumes HCL representationally and semantically. Additionally, we isolate a number of interesting fragments of HCL* which are amenable to tractable model checking in PTIME using succinct implicit representations of model frames. The techniques used to do this involve quantifier elimination [13] and the use of standard techniques from deductive database theory [1, 18].

In Sections 2 and 3 we give an overview of coalition logic [19], quantified coalition logic [3] and higher-order coalition logic [7] to set the context and provide scientific continuity. In Section 4 we propose the higher-order logic HCL* which is a generalization of HCL. Section 5 discusses various succinct representations of models using deductive database techniques. In Section 6 we define several fragments of HCL* and provide lemmas showing that formulas from these fragments are amenable to model-checking in PTIME. Section 7 summarizes assertion types that can be model checked in PTIME. We then conclude with some comments and future work in Section 8.

2. COALITION LOGIC

Coalition Logic (CL) [19] is a propositional modal logic, with modalities indexed by coalitions. The semantics of CL is based on the concept of an *effectivity function* developed in social choice theory to model the ability of a group of individuals. In CL, it is relativized to state and has the form

$$\mathcal{E} : \mathcal{P}(Ag) \times S \longrightarrow \mathcal{P}(\mathcal{P}(S)) \tag{1}$$

where Ag is a set of agents, S is a set of states and $\mathcal{P}(.)$ denotes the powerset of a given set.

For a given coalition $C \subseteq Ag$ and a state $s \in S$, C can cooperate to ensure that for any $T \in \mathcal{E}(C, s)$, the next state will be in T regardless of the actions of other agents outside C. A variety of possible strategies can lead to a set of possible outcomes. To gain some intuitions concerning such functions consider the following example based on one considered in [19].[1]

EXAMPLE 1. *Angelina has to decide whether she wants to marry Edwin, the Judge, or stay single. Edwin and the Judge each can similarly decide whether they want to stay single or marry Angelina. This situation can be modeled using a function $\mathcal{E}$ of the form (1) as follows. The set of agents is $Ag = \{a, e, j\}$ and the set of states is $S = \{s_0, s_s, s_e, s_j\}$, where s_0 is an initial state, where Angelina, Edwin and Judge are singles, s_s is a state where Angelina remains single, s_e where she marries Edwin, and s_j where she marries the Judge.*

Angelina has the right to remain single, so $\{s_s\} \in \mathcal{E}(\{a\}, s_0)$. Edwin can only guarantee that he does not marry Angelina, so we have $\{s_s, s_j\} \in \mathcal{E}(\{e\}, s_0)$. Analogously, for the Judge, we have $\{s_s, s_e\} \in \mathcal{E}(\{j\}, s_0)$. Angelina and Edwin together can achieve any situation except the one where Angelina marries the Judge, and hence $\{s_s\}, \{s_e\} \in \mathcal{E}(\{a, e\}, s_0)$. Again, the situation is analogous for the Judge: $\{s_s\}, \{s_j\} \in \mathcal{E}(\{a, j\}, s_0)$. Edwin and the Judge can together guarantee that Angelina remains single, so $\{s_s\} \in \mathcal{E}(\{e, j\}, s_0)$.

Note that Angelina can act as a dictator forcing everybody to stay single. On the other hand, neither Edwin nor the Judge have such a strong strategy. □

Coalition Logic is a propositional multimodal logic, where formulas are defined by the following grammar:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid [C]\varphi \tag{2}$$

where Ag is fixed, $C \subseteq Ag$ and p ranges over the set of Boolean variables Φ_0. The intended meaning of $[C]\varphi$ is that coalition C has the ability to achieve φ.

Given Ag, a *model* $\mathcal{M}$ is a triple $\langle S, \mathcal{E}, \pi\rangle$, where

- $S = \{s_1, \ldots, s_n\}$ is a finite non-empty set of *states*
- $\mathcal{E}$ is an effectivity function
- $\pi : S \longrightarrow \mathcal{P}(\Phi_0)$ is a *valuation function*, which for every state $s \in S$ gives the set of Boolean variables satisfied at s.

The satisfaction relation is defined as usual for $\top$, atomic variables and connectives. For the modal case we have:

$$\mathcal{M}, s \models_{CL} [C]\varphi \quad \text{iff there is } T \in \mathcal{E}(C, s) \text{ such that for all } t \in T \text{ we have } \mathcal{M}, t \models_{CL} \varphi.$$

Table 1: Properties of effectivity function (1).

For every $C \subseteq Ag$, $s \in S$ and $X \subseteq Y \subseteq S$, if $X \in \mathcal{E}(C, s)$ then $Y \in \mathcal{E}(C, s)$	*outcome monotonicity*
For every $C \subseteq D \subseteq Ag$ and $s \in S$, $\mathcal{E}(C, s) \subseteq \mathcal{E}(D, s)$	*coalition monotonicity*
For all $X \subseteq S$ and $s \in S$, if $X \in \mathcal{E}(C, s)$ then $\bar{X} \notin \mathcal{E}(\bar{C}, s)$	*C-regularity*
For all $C \subseteq Ag$, $\mathcal{E}$ is C-regular	*regularity*
For all $X \subseteq S$ and $s \in S$, if $\bar{X} \notin \mathcal{E}(\bar{C}, s)$ then $X \in \mathcal{E}(C, s)$	*C-maximality*
For all $C \subseteq Ag$, $\mathcal{E}$ is C-maximal	*maximality*
For all $X, Y \subseteq S$, $C \subseteq D \subseteq Ag$ and $s \in S$, if $C \cap D = \emptyset$, $X \in \mathcal{E}(C, s)$ and $Y \in \mathcal{E}(D, s)$ then $X \cap Y \in \mathcal{E}(C \cup D, s)$	*superadditivity*

In [20] the some important properties of effectivity functions are studied. These properties are shown in Table 1, where $\bar{C} \stackrel{\text{def}}{=} Ag \backslash C$ and $\bar{X} \stackrel{\text{def}}{=} S \setminus X$. Restricting effectivity functions with certain properties determines particular classes of models.

An effectivity function $\mathcal{E}$ is *playable* provided that for all $C \subseteq Ag$ and $s \in S$, (i) $\emptyset \notin \mathcal{E}(C, s)$, (ii) $S \in \mathcal{E}(C, s)$, (iii) $\mathcal{E}$ is Ag-maximal, (iv) $\mathcal{E}$ is outcome monotonic and (v) $\mathcal{E}$ is superadditive. The term *playability* is justified by the fact that an effectivity function is playable iff it is an effectivity function of a strategic game (see Theorem 3.2 in [20]).

Coalition monotonicity will play an important role in the context of our model checking results. The following lemma (see, e.g., [19, 20]) guarantees this property for playable effectivity functions.

LEMMA 2. *Every playable effectivity function is regular and coalition monotonic.* □

[1]Characters are actually taken from the comic opera *Trial by Jury* and the example originates from [15].

3. EXTENSIONS OF COALITION LOGIC

Recent work with logical formalisms for representing and reasoning about the abilities of coalitions from a game-theoretic perspective have focused jointly on issues of expressivity and tractability, balancing the two against each other. The major computational problem in this respect is model checking. In this section, we briefly describe two prominent logical formalisms, Quantified Coalition Logic [3] and Higher-Order Coalition logic [7] which attempt to generalize Coalition Logic in several respects. This summary is intended to provide a context for the higher-order logic HCL* and model checking results for fragments of this logic that we introduce in Section 4.

3.1 Quantified Coalition Logic

Quantified Coalition Logic (QCL) [3] is an extension of CL that permits a limited form of quantification over coalitions. Although it provides no increase in expressivity, it is exponentially more succinct than CL and computationally no worse with respect to model checking. Rather than providing C directly in formulas of the form $[C]\varphi$, it allows C to be specified in a special language for coalition predicates given by the grammar:

$$P ::= subseteq(C) \mid supseteq(C) \mid \neg P \mid P \vee P \qquad (3)$$

This allows one to express coalitions based on being a subset (a superset) of a coalition. There are two modalities in QCL:[2]

- $\langle\psi\rangle\varphi$ – there is a coalition satisfying ψ which can achieve φ
- $[\psi]\varphi$ – every coalition satisfying ψ can achieve φ.

3.2 Higher-Order Coalition Logic

Recently, Boella et al [7] introduced Higher-Order Coalition Logic (HCL) as a more general and expressive way to quantify over coalitions. HCL is a monadic second-order logic with special set grouping operators which can be used to characterize different coalitions. Both CL and QCL can be effectively embedded into HCL and there is no need for separate languages to represent coalitions and the effect coalitions have, as is the case with QCL. An axiomatization is provided for HCL and it is shown to be sound and complete for weakly playable semantic structures. Tractable fragments of HCL suitable for efficient model checking have yet to be identified, although this paper will identify a number of such fragments for a related logic HCL*.

We write $\sigma[x := u]$ (respectively, $\sigma[X := U]$) to denote the assignment which differs from σ only in assigning u to x (respectively, U to X).

HCL is a well-behaved fragment of second-order logic, where second-order quantifiers are restricted to binding unary relation variables. Free and bound variables (V_I), relation symbols, connectives $\wedge, \neg$, quantifiers $\forall, \exists$ are defined as in classical first-order logic. To obtain the monadic second-order language, the first-order language is extended by a countable set V_S of set variables (one-argument relation variables) and formulas of HCL are defined using the following grammar:

$$\begin{aligned}\varphi ::= \quad & F(x_1, \ldots, x_k) \mid X(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid \\ & \forall X\varphi \mid \forall x\varphi \mid [\{x\}\varphi]\varphi \mid \langle\{x\}\varphi\rangle\varphi\end{aligned} \qquad (4)$$

where

- $F(x_1, \ldots, x_k)$ is a first-order atomic formula
- $x \in V_I$ and $X \in V_S$
- $\{x\}\varphi$ is a grouping operator which denotes the set of all elements d such that $\varphi[x := d]$ holds.

[2]Note that these modalities are not dual to each other (see [3]).

The intended meaning of modalities in HCL is the same as in the case of QCL. However, HCL offers a much richer language to express properties of coalitions. Consider the following examples, mainly from [7], illustrating the expressiveness of HCL:

- $\forall x(super\_user(x) \rightarrow user(s))$ – any super user is a user
- $\forall X(\forall x(X(x) \rightarrow user(x)) \rightarrow \langle\{y\}X(y)\rangle\varphi)$ – there is a coalition, where all $users$ can achieve φ
- $\langle\{x\}\psi(x)\rangle\varphi \rightarrow \langle\{y\}\exists x(\psi(x) \wedge collaborates(y, x))\rangle\varphi$ – whenever there is a coalition, say C, satisfying ψ which can achieve φ, there is also a coalition consisting of collaborators of at least one member of C which can achieve φ.

Semantic structures for HCL correspond to weak playability. An effectivity function $\mathcal{E}$ is *weakly playable* if for all $C \subseteq D \subseteq Ag$ and $s \in S$, (i) $\emptyset \notin \mathcal{E}(Ag, s)$, (ii) $\emptyset \in \mathcal{E}(D, s)$ implies $\emptyset \in \mathcal{E}(C, s)$, (iii) $\emptyset \notin \mathcal{E}(\emptyset, S)$ implies $S \in \mathcal{E}(C, s)$, (iv) $\mathcal{E}$ is Ag-maximal, (v) $\mathcal{E}$ is outcome monotonic and (vi) $\mathcal{E}$ is superadditive.

HCL uses a *general* or *Henkin* semantics. A more detailed discussion about the semantic basis for HCL is provided in Section 4.

4. COALITION LOGIC HCL*

The goal of this paper is to provide a logical formalism for reasoning about the abilities of coalitions that has high expressiveness, yet is still amenable to tractable model-checking. HCL certainly has high expressiveness and is more general than both CL and QCL. On the other hand, it currently lacks nice computational properties for different fragments of the language. Our results are intended to provide both well-behaved fragments and tractable model checking techniques for higher-order logic using HCL*

In this section, we introduce the higher-order logic HCL*. For the purposes of continuity and context, before providing formal definitions, we list the difference between HCL and HCL* and then remark on some of these differences.

1. HCL restricts quantification over relations to unary (monadic) predicates. HCL* relaxes this restriction and permits quantification over relations of arbitrary arity.

2. HCL includes both the diamond and box operators in the language. HCL* excludes the box operator from the language. Since box can be defined in terms of diamond, this is done for technical reasons pertaining to model-checking and does not limit expressivity of the language in general.

3. HCL uses a *general* or *Henkin* semantics which approximates the standard semantics for higher-order logic. HCL* uses the standard semantics for higher-order logic.

4. HCL restricts frames to those whose effectivity function is weakly playable. HCL* only requires frames to be monotonic (both outcome and coalition monotonic) for our model checking results to apply.

Before commenting on these differences, we provide the syntax and semantics of HCL*.

The syntax of HCL* is given by the following grammar:

$$\begin{aligned}\varphi ::= \quad & \top \mid F(x_1, \ldots, x_k) \mid X(x_1, \ldots, x_k) \mid \\ & \neg\varphi \mid \varphi \vee \varphi \mid \forall X\varphi \mid \forall x\varphi \mid \langle\{x\}\varphi\rangle\varphi\end{aligned} \qquad (5)$$

A *coalition frame* is a tuple $\langle Ag, S, \mathcal{E}\rangle$, where Ag is a finite, nonempty set (of agents), S is a finite set of states and $\mathcal{E}$ is an effectivity function.

A coalition frame is *monotonic* if its effectivity function is both outcome monotonic and coalition monotonic. Monotonicity

is the weakest requirement necessary for our model-checking results. In the rest of the paper we assume that the frames considered are monotonic.

A *coalition model based on a frame* $\langle Ag, S, \mathcal{E}\rangle$ is a tuple $\mathcal{M} = \langle Ag, S, \mathcal{E}, \mathcal{I}, \sigma\rangle$, where:

- $\mathcal{I}$ is a first-order interpretation, for any first order formula α and $s \in S$ it assigns a set of tuples $\alpha^{\mathcal{I}}(s)$ satisfying α in state s
- σ assigns in states: (i) values in Ag to individual variables, (ii) sets of tuples of respective arity to relation variables.[3]

Let $\mathcal{M} = \langle Ag, S, \mathcal{E}, \mathcal{I}, \sigma\rangle$ be a coalition model and $s \in S$. We define the *satisfaction relation* as follows, where $\mathcal{M}'{=}\langle Ag, S, \mathcal{E}, \mathcal{I}, \sigma[x := d]\rangle$ and $\mathcal{M}''{=}\langle Ag, S, \mathcal{E}, \mathcal{I}, \sigma[X := D]\rangle$:

- $\mathcal{M}, s \models \top$
- $\mathcal{M}, s \models F(x_1, \ldots, x_k)$ iff $\langle \sigma(x_1), \ldots, \sigma(x_k)\rangle \in F^{\mathcal{I}}(s)$
- $\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$
- $\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$
- $\mathcal{M}, s \models X(x_1, \ldots, x_k)$ iff $\langle \sigma(x_1), \ldots, \sigma(x_k)\rangle \in \sigma(X, s)$
- $\mathcal{M}, s \models \forall x\varphi$ iff for all $d \in Ag$, $\mathcal{M}', s \models \varphi$
- $\mathcal{M}, s \models \forall X\varphi$, for a k-argument relation variable X, iff for all $D \in \mathcal{P}(Ag^k)$, $\mathcal{M}'', s \models \varphi$
- $\mathcal{M}, s \models \langle\{x\}\psi\rangle\varphi$ iff there is $C{=}\{d \mid \mathcal{M}', s \models \psi\}$ and $T \in \mathcal{E}(C, s)$ such that for all $t{\in}T$, $\mathcal{M}, t{\models}\varphi$.

HCL uses a *general* or *Henkin* semantics which approximates the standard semantics used by HCL* for higher-order logic. Henkin semantics is weaker than the standard semantics. In general $\models_H \varphi$ implies $\models \varphi$. This is in large part due to restriction of second-order quantification solely to definable sets which is a prerequisite for showing completeness of the proof system associated with Henkin semantics. The standard semantics is not restricted to definable sets and in HCL*, second-order quantifiers range over *all* relations of respective arity, which directly reflects intuitions behind them. On the other hand, HCL* is undecidable. However, in the context of model-checking, when a given finite structure is fixed and the language includes equality ($=$) as well as constants denoting domain elements, then every set becomes definable and both semantics become compatible in the sense that for any finite model M, $M \models_H \varphi$ iff $M \models \varphi$. Note that the required constants and equality is always available given the unique names and closed world assumptions.

HCL* permits quantification over relation variables of any arity, not only monadic ones, as required in HCL. Due to this, HCL* provides increased expressivity. For example, the following HCL* formula is outside of the HCL syntax:

$$\begin{aligned}&\forall u\forall X\big((\forall x\forall y(X(x,y) \rightarrow Cn(x,y))\wedge\\ &\forall x\forall z((Cp(x,z) \vee \exists y(X(x,y) \wedge X(y,z))) \rightarrow X(x,z)))\\ &\rightarrow \langle\{y\}\exists x(X(x,y) \wedge S(x))\rangle W(u)\big).\end{aligned} \quad (6)$$

If, for example, Cn stands for "controls", Cp for "being able to cooperate", S for "smart" and W for "wins" then (6) states that

> for every agent u, there is a coalition formed from agents that are able to cooperate with one another and are controlled transitively by smart agents, that can make u a winner,

where "controlled transitively" is formalized by a transitive X containing relation Cp and contained in Cn.

On the other hand, the box operator, $[\{x\}\psi]$, while included in HCL syntax, is not part of HCL* syntax. The box operator, $[\{x\}\psi]$, is definable by means of the diamond $\langle\{x\}\psi\rangle$ operator (see [3]). However, using such definitions results in an exponential blow up in the length of formulas. In the context of model-checking and quantifier elimination, dealing with formulas which include the box operator directly is problematic, as they are defined by a formula using the sequence of quantifiers $\forall\exists\forall$. The first two alternating quantifiers binding relational variables cause substantial technical problems.

HCL* restricts frames to those whose effectivity function has the property of monotonicity (both outcome and coalition monotonic). Observe that playability implies both conditions: outcome monotonicity (by definition of playability) and coalition monotonicity (by Lemma 2). HCL considers frames whose effectivity functions have the property of being weakly playable. If one considers weak playability only, outcome monotonicity is assumed by definition, however one has to additionally assume the coalition monotonicity property.

5. REPRESENTATION OF MODELS

Querying deductive databases and model-checking are very similar. When querying a deductive database, we can view the database as a model and the query as a formula which is being checked for satisfaction relative to the database. We will in fact take advantage of this analogy when doing model-checking in HCL*. Since functions are typically not allowed in deductive databases, we will equivalently replace effectivity functions $\mathcal{E}$ by *effectivity relations*:

$$E \subseteq \mathcal{P}(Ag) \times S \times \mathcal{P}(S) \quad (7)$$

such that $E(C, s, T) \stackrel{\text{def}}{\equiv} T \in \mathcal{E}(C, s)$. This representation then views a model frame in HCL* as a deductive database containing the relation E and model checking as satisfying a query relative to that database. One can then study the complexity of model checking relative to the language fragments of HCL* used in the query language by using results from deductive database theory and descriptive complexity. Observe that one can identify any set with its characteristic relation, i.e., rather than using set X, we may use the unary relation $X(x) \stackrel{\text{def}}{\equiv} x \in X$.

To simplify the presentation, coalition models will be represented in a deductive database using the relation $E()$ defined above, but more succinct representations are also possible, as discussed in the end of this section. The extensional part will contain facts represented as $E()$ atoms and the intensional part will contain a rule encapsulating monotonicity assumptions:

$$(E(X, x, Y) \wedge X \subseteq X' \wedge Y \subseteq Y') \rightarrow E(X', x, Y'). \quad (8)$$

This, in fact, ensures a succinct representation of coalition models when doing model checking. Since we assume that coalition models are monotonic, we do not have to include information that follows from monotonicity. The same representation is used in [19] for outcome monotonicity only. Our representation of the effectivity relation is more succinct, since we also use coalition monotonicity. This may result in exponentially smaller representations. For example, if $E(\{a\}, s, \{s\})$ holds, we do not have to include an exponential number of facts of the form $E(C, s, \{s\})$ for all C with $a \in C$ in the model. Consequently, we avoid the problem of explicit model checking criticized elsewhere in the literature (e.g., [3]). It is important to note that rule (8) is not intended to generate a possibly exponential number of facts. It is only used to

[3]In this definition, we restrict models to a single sort for agents, but our results are also valid for many-sorted structures.

reduce the size of models and to model check facts involving $E()$ literals as is demonstrated in the proof of Lemma 6.

We can also increase succinctness of the model representation by applying the Closed World Assumption, allowing one not to list negative facts. Using this approach, model checking then becomes similar to querying deductive databases (see, e.g., [1]). The following example illustrates the representation used for coalition models.

EXAMPLE 3 (EXAMPLE 1 CONTINUED). *The model considered in the introductory example consists of the following facts:*

$$E\big(\{a\}, s_0, \{s_s\}\big), E\big(\{e\}, s_0, \{s_s, s_j\}\big), E\big(\{j\}, s_0, \{s_s, s_e\}\big)$$
$$E\big(\{a, e\}, s_0, \{s_s\}\big), E\big(\{a, e\}, s_0, \{s_e\}\big)$$
$$E\big(\{a, j\}, s_0, \{s_s\}\big), E\big(\{a, j\}, s_0, \{s_j\}\big), E\big(\{e, j\}, s_0, \{s_s\}\big).$$

Recall that the rule (8) reflecting the monotonicity of E is in the intensional part of the database. Observe that due to (8), one can remove facts $E\big(\{a, e\}, s_0, \{s_s\}\big)$ and $E\big(\{a, j\}, s_0, \{s_s\}\big)$.

Also, no matter how many other agents and states are involved, the above model of this particular scenario does not need to be extended. □

Using our representation, the *size of a coalition frame* $\mathcal{F} = \langle Ag, S, \mathcal{E}\rangle$ is given by:

$$|\mathcal{F}| \stackrel{\text{def}}{=} \max\Big\{|Ag|, |S|, \sum_{E(C,s,X)\in\mathcal{E}} \big(|C| + |X| + 1\big)\Big\}. \quad (9)$$

Complexity results will be provided w.r.t. size of models. The input or query formula is considered to be fixed, thus has a constant length. This is standard practice. Observe that the size of models can, in the worst case, be exponential w.r.t. both $|Ag|$ and $|S|$.

In this context, what do we mean by *succinct* representation of models? Our claim is that a large class of models used in practical applications can be succinctly represented by leveraging formal results from deductive database theory. Recall that we represent an effectivity function as a relation $E()$ and then represent that relation (usually defined in terms of atoms) in a deductive database with an intensional rule for monotonicity. In fact, one can use any equivalent formula in first-order fixpoint logic to represent the effectivity relation. This formula may include fixpoints, quantifiers and relations other than $E()$. Additionally, we can use other intensional rules in addition to the monotonicity rule.

Why is this fundamentally important? Well, any model that is polynomial in the size of agents and states can be equivalently represented as a fixpoint formula and this fixpoint formula can be polynomially compiled into a deductive database. The tractability of model checking obviously applies to this class of models, too.[4] Let's illustrate this idea with the following Example 4.

EXAMPLE 4. *Consider n sax players, m bass players and k drummers ($n, m, k \geq 1$). To organize a concert, one needs at least a trio consisting of a sax player, a bass player and a drummer. Let s_0 be the initial state, s_c be the state where a concert is possible and s_n where it is not. Let $S(x), B(x)$ and $D(x)$ stand for "x is a sax player", "x is a bass player" "x is a drummer", respectively. Then in this model we need $n + m + k$ facts:*[5]

$$\{S(s) \mid s \text{ is a sax player}\} \cup \{B(b) \mid b \text{ is a sax player}\} \cup$$
$$\{D(d) \mid d \text{ is a drummer}\},$$

in addition to facts reflecting that coalitions consisting of all sax players (of all bass players or of all drummers) have the power to block the concert:

$$E(S, s_0, \{s_n\}), E(B, s_0, \{s_n\}), E(D, s_0, \{s_n\})$$

as well as rule (8) and the following intensional rule expressing that any suitable trio makes the concert possible:

$$(S(x)\wedge C(x)\wedge B(y)\wedge C(y)\wedge D(z)\wedge C(z)) \rightarrow E(C, s_0, \{s_c\}). \quad (10)$$

Note that the size of the model is $O(n+m+k)$ (and after unwinding rule (10), it is $O(n+m+k+n*m*k)$) rather than $O(2^{n+m+k})$, when rules (8) and (10) are not used. □

To our knowledge, these techniques for reducing the size of models resulting in succinct representations is novel and quite powerful. It also shows how the integration of the model-checking techniques developed in this paper together with deductive database techniques results in an expressive and efficient representational technique. Additionally, one has a more formal characterization of what is meant by *succinct* representation.

6. MODEL CHECKING

When checking satisfiability of a formula from HCL*, we do this relative to a model and a state. Given an arbitrary formula in HCL*, we will introduce a translation operator Tr which maps each formula into another second-order formula in HCL*. This operator has two purposes.

1. It parameterizes all relational predicates in the formula with an additional state argument.
2. It translates any instance of the diamond modality into a second-order formula which is equivalent.

The net result is that a query is now reduced to an arbitrary second-order formula without modalities in HCL* whose satisfiability we would like to check relative to a coalition model. Transforming the model checking problem into the problem of a 2nd-order query to a deductive database representing a coalition model has great advantages. We can now isolate fragments of second-order logic which, through the use of quantifier elimination reduce the problem to a 1st-order or fixpoint query on a relational database. Results from deductive database theory ensure us that this can be done efficiently relative to the size of the database which we know contains a succinct representation of a coalition model due to the advantageous use of the monotonicity constraint.

We now provide the translation operator *Tr*. The translation *Tr*(φ, s) results in a formula expressing the fact that formula φ is satisfied in state s. To define *Tr*, with every k-argument symbol like F, X of the HCL* language we associate respectively a fresh $(k+1)$-argument symbol F', X' not appearing in the original HCL* language:

- *Tr*$(F(x_1, \ldots, x_k), s) \stackrel{\text{def}}{=} F'(s, x_1, \ldots, x_k)$
- *Tr*$(\neg\varphi, s) \stackrel{\text{def}}{=} \neg$*Tr*$(\varphi, s)$
- *Tr*$(\varphi \vee \psi, s) \stackrel{\text{def}}{=}$ *Tr*$(\varphi, s) \vee$ *Tr*(ψ, s)
- *Tr*$(X(x_1, \ldots, x_k), s) \stackrel{\text{def}}{=} X'(s, x_1, \ldots, x_k)$
- *Tr*$(\forall x\varphi, s) \stackrel{\text{def}}{=} \forall x$*Tr*$(\varphi, s)$
- *Tr*$(\forall X\varphi, s) \stackrel{\text{def}}{=} \forall X$*Tr*$(\varphi, s)$
- *Tr*$(\langle\{x\}\psi\rangle\varphi, s) \stackrel{\text{def}}{=} \exists X\big(\forall x(X(x) \equiv$ *Tr*$(\psi, s))\wedge$
  $\exists Y(E(X, s, Y)) \wedge \forall y(Y(y) \rightarrow$ *Tr*$(\varphi, y))\big)$.

[4] In fact, any equivalent representation of the class of fixpoint formulas such as stratified Datalog would also do as a representational mechanism.

[5] We implicitly assume that all players are different. For example, no sax player is at the same time a bass player, etc.

Observe that the last clause of the *Tr* operator above translates any instance of the diamond operator into an equivalent second-order formula. The following important lemma allows us to replace these translations of the diamond operator in a query formula with a more efficient but equivalent query about $E()$ without second-order quantifiers.

LEMMA 5 (DIAMOND ELIMINATION LEMMA). *For every coalition model $\mathcal{M} = \langle Ag, S, \mathcal{E}, \mathcal{I}, \sigma\rangle$ and $s \in S$,*

$$\mathcal{M}, s \models Tr(\langle\{x\}\psi\rangle\varphi, s) \equiv E(\{x\}Tr(\psi, s), s, \{y\}Tr(\varphi, y)).$$

PROOF.
($\rightarrow$) Assume that $\mathcal{M}, s \models Tr(\langle\{x\}\psi\rangle\varphi, s)$. By definition,

$$\mathcal{M}, s \models \exists X\big(\forall x(X(x) \equiv Tr(\psi, s)) \wedge \exists Y(E(X, s, Y)) \wedge \forall y(Y(y) \rightarrow Tr(\varphi, y))\big),$$

In particular, $\mathcal{M}, s \models \exists X \forall x(X(x) \rightarrow Tr(\psi, s))$. By monotonicity of E we have that $\mathcal{M}, s \models E(\{x\}Tr(\psi, s), s, \{y\}Tr(\varphi, y))$.

($\leftarrow$) Assume that $\mathcal{M}, s \models E(\{x\}Tr(\psi, s), s, \{y\}Tr(\varphi, y))$. Let $X(x) \stackrel{\text{def}}{\equiv} Tr(\psi, s)$ and $Y(y) \stackrel{\text{def}}{\equiv} Tr(\varphi, y)$. Such X and Y obviously satisfy

$$\mathcal{M}, s \models \forall x(X(x) \equiv Tr(\psi, s)) \wedge E(X, s, Y) \wedge \forall y(Y(y) \rightarrow Tr(\varphi, y)).$$

Therefore,

$$\mathcal{M}, s \models \exists X\big(\forall x(X(x) \equiv Tr(\psi, s)) \wedge \exists Y(E(X, s, Y)) \wedge \forall y(Y(y) \rightarrow Tr(\varphi, y))\big),$$

so, by definition of *Tr*, $\mathcal{M}, s \models Tr(\langle\{x\}\psi\rangle\varphi, s)$, which completes the proof. □

Given this lemma and the following lemma, we can already show that formulas in the fragment of HCL* containing arbitrary instances of the diamond operator, but no other 2nd-order quantifiers can be model-checked for satisfiability in PTIME.

LEMMA 6. *Model checking for formulas without second-order quantifiers is in* PTIME.

PROOF. Let $\mathcal{M}$ be a coalition model and s a state in $\mathcal{M}$. We first eliminate diamonds from the input formula.

Checking whether $\mathcal{M}$ is a model for a formula without occurrences of effectivity relation can be done in polynomial time in the standard way (see, e.g., [1, 18]).

Checking the truth value of a given expression of the form $E\big(\{x\}Tr(\psi, s), s, \{y\}\varphi(y)\big)$ can be done by traversing facts in the model and checking whether there is a fact $E\big(C, s, T\big)$ such that $\mathcal{M}, s \models E\big(C, s, T\big) \rightarrow E\big(\{x\}Tr(\psi, s), s, \{y\}\varphi(y)\big)$. Such a fact exists iff $\mathcal{M}, s \models E\big(\{x\}Tr(\psi, s), s, \{y\}\varphi(y)\big)$. To check the required implication we use monotonicity: we simply check whether:

- the set C is included in the set being the value of $\{x\}Tr(\psi, s)$ in $\mathcal{M}$ and s
- the set T is included in the set being the value of $\{y\}\varphi(x)$ in $\mathcal{M}$ and s.

Computing the sets $\{x\}Tr(\psi, s)$ and $\{y\}\varphi(x)$ can be done in polynomial time by an obvious extension of the technology of querying databases in logic (see, [1]). □

Let us now assume that our queries use both the diamond operator and additional 2nd-order quantifiers. Our next task is to identify additional fragments of HCL* where these additional quantifiers can be eliminated. Any formulas in such fragments are then guaranteed to be amenable to model-checking in PTIME based on the results which follow.

Since universal quantifiers are definable by existential ones (using the standard definition $\forall = \neg\exists\neg$), in what follows we will focus on the existential fragment of HCL* without any loss in expressivity. The *existential fragment* of HCL* is the smallest set containing arbitrary HCL* formulas without any universal quantifiers, formulas of the form

$$\exists X_1 \ldots \exists X_r \varphi, \tag{11}$$

where φ contains no second-order quantifiers, and which is closed under Boolean connectives, first-order quantifiers and modalities.

The first fragment of well-behaved formulas will be those that are positive. By the *positive fragment* of HCL*, we mean formulas in the existential fragment with the additional restriction that for formulas of the form (11), φ is positive w.r.t. all relation variables $X_1 \ldots X_r$. The standard definition of positive formulas [13] will have to be slightly modified since relations such as the effectivity relation E have arguments that might be formulas.

An occurrence of a relation variable X is *positive* (*negative*) in φ, if it appears under an even (respectively, odd) number of negation signs.[6] A formula φ is *positive* (*negative*) w.r.t. X if all occurrences of X in φ are positive (respectively, negative).

For example, formula

$$\neg X(a) \vee Y(b) \vee \neg E\big(\{x\}X(x), s, \{y\}\neg Y(y)\big)$$

is negative w.r.t. X and positive w.r.t. Y.

We could deal with the monotonic fragment of HCL* instead. The reason we do not is that in general, checking monotonicity or down-monotonicity is not decidable, while checking positivity and negativity can be done in time linear in the length of the considered formula. Since positivity implies monotonicity and negativity implies down-monotonicity, it is algorithmically convenient to use positivity and negativity rather than monotonicity.

We now have the following lemma.

LEMMA 7. *Model checking for the positive fragment of* HCL* *is in* PTIME.

PROOF. In light of Lemma 6 it suffices prove the claim for formulas of the form (11), where φ is positive w.r.t. $X_1 \ldots X_r$. Consider φ' obtained from φ by replacing all occurrences of $X_1 \ldots X_r$ by $\top$. It appears that φ' is equivalent to formula $\exists X_1 \ldots \exists X_r \varphi$. To show this, consider a coalition model $\mathcal{M}$ and its state s.

($\rightarrow$) If $\mathcal{M}, s \models \varphi'$ then there are $X_1 \ldots X_r$ such that $\mathcal{M}, s \models \exists X_1 \ldots \exists X_r \varphi$ (it suffices to define all of them to be $\top$).

($\leftarrow$) Assume that $\mathcal{M}, s \models \exists X_1 \ldots \exists X_r \varphi$. Formula φ is positive w.r.t. $X_1 \ldots \exists X_r$, so monotone w.r.t. $X_1 \ldots \exists X_r$.[7] Since for each $1 \leq i \leq r$, formula $X_i(\ldots) \rightarrow \top$ is a tautology, by monotonicity we get that $\mathcal{M}, s \models \varphi'$. □

The final fragment of well-behaved formulas we will focus on are the semi-Horn formulas. By the *semi-Horn fragment* of HCL* we mean formulas in the existential fragment of HCL* which have the following form:

$$\exists \bar{X}\big\{\forall \bar{x}[\alpha(\bar{X}, \bar{x}, \bar{z}) \rightarrow X_i(\bar{x})] \wedge \beta(\bar{X})\big\}$$

where $\bar{X}$ stands for $X_1, \ldots, X_r$, $1 \leq i \leq r$, α is positive w.r.t. each of $X_1, \ldots, X_r$ and β is negative w.r.t. each of $X_1, \ldots, X_r$.

We will now show that semi-Horn formulas in the existential fragment can be reduced to logically equivalent fixpoint formulas without higher-order quantifiers and that such formulas can be

[6] Here, as usual, each implication $\varphi \rightarrow \psi$ is replaced by $\neg\varphi \vee \psi$ and each equivalence $\varphi \equiv \psi$ is replaced by $(\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$.
[7] Monotonicity of effectivity relations is used here, too.

model-checked in PTIME. To show tractability of model checking for the semi-Horn fragment of HCL*, we will use the following theorem from [14] (see also [13]), where by LFP $X(\bar{x})\ [\alpha(X,\bar{x},\bar{z})]$ and GFP $X(\bar{x})\ [\alpha(X,\bar{x},\bar{z})]$ we denote the least and the greatest fixpoint of $\alpha(X,\bar{x},\bar{z})$, i.e., the least and the greatest (w.r.t. inclusion) relation satisfying $X(\bar{x}) \equiv \alpha(X,\bar{x},\bar{z})$.[8] For a detailed discussion of fixpoint calculus and their use in databases see, e.g., [1, 18]. For our purposes it is important to show that fixpoint queries are computable in PTIME w.r.t. size of the database (in our case w.r.t. size of the model).

The following additional notation will be used in the theorem and proof. Let $\alpha(x,\bar{y})$ be a higher-order formula, $X(\bar{x})$ be a (higher-order) relation, $\gamma(\bar{x})$ be a (higher-order) formula with all free variables being $\bar{x}$. Then $\alpha^{X(\bar{x})}_{\gamma(\bar{x})}$ denotes the formula obtained from α by substituting all subformulas of the form $X(\bar{t})$ by $\gamma(\bar{t})$.

THEOREM 8. *Let X be a relation variable and $\alpha(X,\bar{x},\bar{z})$, $\beta(X)$ be formulas with relations of arbitrary order, where the number of distinct variables in $\bar{x}$ is equal to the arity of X. Let α be monotone w.r.t. X.*

If $\beta(X)$ is down-monotone w.r.t. X then

$$\exists X\big\{\forall\bar{x}[\alpha(X,\bar{x},\bar{z}) \rightarrow X(\bar{x})] \wedge \beta(X)\big\} \equiv \beta(X)^{X(\bar{x})}_{\text{LFP}\ X(\bar{x})\ [\alpha(X,\bar{x},\bar{z})](\bar{x})}. \tag{12}$$

If $\beta(X)$ is monotone w.r.t. X then

$$\exists X\big\{\forall\bar{x}[X(\bar{x}) \rightarrow \alpha(X,\bar{x},\bar{z})] \wedge \beta(X)\big\} \equiv \beta(X)^{X(\bar{x})}_{\text{GFP}\ X(\bar{x})\ [\alpha(X,\bar{x},\bar{z})](\bar{x})}. \tag{13}$$

The following example illustrates the use of Theorem 8

EXAMPLE 9. *Consider formula (6). It is universally quantified, so we first negate it to replace universal quantifiers by existential quantifiers:*

$$\begin{array}{l}\neg\exists u\exists X\big(\forall x\forall y(X(x,y) \rightarrow Cn(x,y)) \wedge \\ \forall x\forall z((Cp(x,z) \vee \exists y(X(x,y) \wedge X(y,z))) \rightarrow X(x,z)) \wedge \\ \neg\langle\{y\}\exists x(X(x,y) \wedge S(x))\rangle\, W(u)\big).\end{array} \tag{14}$$

Formula under $\exists u$ is semi-Horn. To apply our method we first have to translate the formula using translation Tr and applying Lemma 5. The result is:

$$\begin{array}{l}\neg\exists u\exists X'\big(\forall x\forall y(X'(s,x,y) \rightarrow Cn'(s,x,y)) \wedge \\ \forall x\forall z((Cp'(s,x,y) \vee \exists y(X'(s,x,y) \wedge X'(s,y,z))) \rightarrow X'(s,x,z)) \\ \wedge\ \neg E(\{y\}\exists x(X'(s,x,y) \wedge S'(s,x)), s, W'(s,u))\big).\end{array}$$

To apply the equivalence (12) we formally need a small trick,[9] *namely the second line of the above formula is equivalent to*

$$\forall t\forall x\forall z\big((t = s \wedge (Cp'(s,x,y) \vee \exists y(X'(s,x,y) \wedge X'(s,y,z)))) \rightarrow X'(t,x,z)\big)$$

Now we apply equivalence (12) of Theorem 8 and obtain the following equivalent formula:

$$\begin{array}{l}\neg\exists u\big(\forall x\forall y(X'(s,x,y) \rightarrow Cn'(s,x,y)) \wedge \\ \neg E(\{y\}\exists x(X'(s,x,y) \wedge S'(s,x)), s, W'(s,u))\big),\end{array} \tag{15}$$

where X' should be respectively replaced by

$$\text{LFP}\ X'(t,x,z)\ [t = s \wedge (Cp'(s,x,z) \vee \exists y(X'(s,x,y) \wedge X'(s,y,z)))]. \tag{16}$$

[8] We shall only use this notation in contexts where the least and the greatest relation exist.

[9] Which later will appear reversible.

Using the fact that $t = s$, we get the following equivalent of (16):[10]

$$\text{LFP}\ X'(s,x,z)\ [Cp'(s,x,z) \vee \exists y(X'(s,x,y) \wedge X'(s,y,z))]. \tag{17}$$

Formula (16), in which X's are respectively replaced by the least fixpoint formula (17), is the input to the model checking method. □

Since positivity implies monotonicity and negativity implies down-monotonicity, we have the following theorem as a consequence of equivalence (12) from Theorem 8.

THEOREM 10. *Model checking for the semi-Horn fragment of* HCL* *is in* PTIME.

PROOF. Observe that second-order quantifiers can be eliminated from semi-Horn formulas using (12). The resulting formula is a fixpoint formula. Checking whether it holds in a given model $\mathcal{M}$ can be done in time polynomial w.r.t. the size of $\mathcal{M}$.

Note that in Theorem 8 second-order quantification binds a single relation variable, while in semi-Horn formula there might be a longer tuple of existential quantifiers. However, such a tuple can be encoded by a single relation variable by adding a special argument (or a number of arguments) identifying "original" relations. For example, to encode $X_1,\ldots,X_r$, we can consider a relation variable $X(\hat{i},\bar{x})$, where $\hat{i}$ is the special argument, $\bar{x}$ is the list of arguments of the length being the maximum of lengths of arguments of $X_1,\ldots,X_r$. Now, rather than writing $X_i(\bar{x}_i)$ one can write $X(\hat{i},\bar{x}_i,\bar{y})$, where $\bar{y}$ is a tuple of dummy arguments, needed when the number of arguments of X_i is smaller than the number of arguments in $\bar{x}$. □

One can also define the dual form of semi-Horn formulas. By the *dual semi-Horn fragment* of HCL* we mean formulas in the existential fragment of HCL*, which have the following form:

$$\exists\bar{X}\big\{\forall\bar{x}[X_i(\bar{x}) \rightarrow \alpha(\bar{X},\bar{x},\bar{z})] \wedge \beta(\bar{X})\big\}$$

where $\bar{X}$ stands for $X_1,\ldots,X_r$, $1 \le i \le r$ and α,β are both positive w.r.t. each of $X_1,\ldots,X_r$.

By applying the equivalence (13) from Theorem 8, we have the following theorem.

THEOREM 11. *Model checking for the dual semi-Horn fragment of* HCL* *is in* PTIME. □

7. ASSERTION TYPES EXPRESSIBLE IN TRACTABLE FRAGMENTS OF HCL*

Let us summarize a number of useful types of assertions which can be represented in those fragments of HCL* which admit tractable model checking.

The first, obvious class of expressible formulas is provided by Lemma 6. This is quite a rich class of formulas. Probably the most interesting among them are *existence assertions* allowing one to express that, in a given circumstance $C(\bar{x})$, there is a coalition satisfying a certain condition A which can lead to a set of states guaranteeing that a given goal $G(\bar{z})$ is achieved:

$$C(\bar{x}) \rightarrow \langle\{y\}A(y)\rangle\ G(\bar{z}). \tag{18}$$

Note that both C and G can still contain diamonds. In applications, one can frequently expect queries of the form (18), often simplified to the case where $C(\bar{x})$ is true, i.e., when one is interested whether in a given situation there is a coalition able to achieve a given goal (i.e., $\langle\{y\}A(y)\rangle\ G(\bar{z})$).

[10] Explaining what we have meant by "reversibility" of the trick applied earlier.

Another significant class of formulas is provided by Lemma 7 together with Theorems 10 and 11. A very important subclass of such formulas is the one, where using existential second-order quantifiers, one can "transfer" coalitions among diamonds. We call such assertions *transfer assertions*, which typically can take the form

$$\exists X\big(\forall\bar{x}(A(X,\bar{x}) \rightarrow \langle\{y\}B(X,\bar{x},y)\rangle) \wedge C(X)\big). \qquad (19)$$

Of course, tractable model checking is possible when such a formula translates into a positive or (dual) semi-Horn formula, like in the following example,

$$\exists X\big(\forall\bar{x}(X(\bar{x}) \rightarrow (\langle\{y\}(X(y) \vee large(y))\rangle goal \wedge \\ \forall\bar{x}(strong(x) \rightarrow X(\bar{x}))\big),$$

expressing that there is a coalition consisting of all $strong$ agents in addition to possibly some $large$ agents, capable of achieving the $goal$.

Observe that the class of formulas which admit tractable model checking using the methods provided in this paper is not limited to the above types of assertions, but these assertion types do show the practical use of the fragments we deal with.

8. CONCLUSIONS

We have introduced the higher-order logic HCL* which can be used for reasoning about the abilities of coalitions and interactions between them. HCL* is a generalization of HCL which subsumes both CL and QCL. Additionally, we have isolated a number of expressive fragments of HCL* and shown that the model-checking problem for these fragments can be solved in PTIME by appealing to use of quantifier elimination, results from deductive database theory and descriptive complexity. Additionally, through advantageous use of monotonicity constraints on coalition frames and use of deductive database techniques one can often get exponentially more succinct representations of coalition models in the model-checking process.

For formulas outside of this fragment one could use extensions of the second-order quantifier elimination algorithm of [11] which, although often finding reductions outside these fragments, does not guarantee such reductions. For a presentation of this algorithm as well as other relevant techniques see also [13].

9. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley Pub. Co., 1996.

[2] T. Ågnotes, W. van der Hoek, and M. Wooldridge. On the logic of coalitional games. In *Proc. AAMAS'06*, pages 153–160. AAAI, 2006.

[3] T. Ågnotes, W. van der Hoek, and M. Wooldridge. Quantified Coalition Logic. In *Proc. IJCAI'07*, pages 1181–1186. AAAI, 2007.

[4] T. Ågotnes and H. van Ditmarsch. Coalitions and announcements. In *Proc. AAMAS '08*, pages 673–680, 2008.

[5] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[6] P. Balbiani, O. Gasquet, A. Herzig, F. Schwarzentruber, and N. Troquard. Coalition games over Kripke semantics. In C. Dégremont, L. Keiff, and H. Rückert, editors, *Dialogues, Logics and Other Strange Things – Essays in Honour of Shahid Rahman*, pages 11–32. College Publications, 2008.

[7] G. Boella, D.M. Gabbay, V. Genovese, and L. van der Torre. Higher-Order Coalition Logic. In *Proc. ECAI'10*, pages 555–560, 2010.

[8] J. Broersen, A. Herzig, and N. Troquard. From coalition logic to STIT. *Electron. Notes Theor. Comput. Sci.*, 157(4):23–35, 2006.

[9] N. Bulling, J. Dix, and C.I. Chesnevar. Modelling coalitions: ATL + argumentation. In *AAMAS '08*, pages 681–688, 2008.

[10] V. Dignum, editor. *Handbook of Research on Multi-Agent Systems*. Information Science Reference, 2009.

[11] P. Doherty, W. Łukaszewicz, and A. Szałas. Computing circumscription revisited. *Journal of Automated Reasoning*, 18(3):297–336, 1997.

[12] B. Dunin-Keplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach*. John Wiley & Sons, Ltd., 2010.

[13] D.M. Gabbay, R. Schmidt, and A. Szałas. *Second-Order Quantifier Elimination. Foundations, Computational Aspects and Applications*, volume 12 of *Studies in Logic*. College Publications, 2008.

[14] D.M. Gabbay and A. Szałas. Second-order quantifier elimination in higher-order contexts with applications to the semantical analysis of conditionals. *Studia Logica*, 87:37–50, 2007.

[15] A. Gibbard. A Pareto-consistent libertarian claim. *Journal of EconomicTheory*, 7(4):388–410, 1974.

[16] V. Goranko. Coalition games and alternating temporal logics. In *Proc. 8th Conf. on Theoretical Aspects of Rationality and Knowledge TARK*, pages 259–272. Morgan Kaufmann, 2001.

[17] S. Ieong and Y. Shoham. Marginal contribution nets: A complact representation scheme for coalitional games. In *Proc. ACM EC*, pages 170–179, 2006.

[18] N. Immerman. *Descriptive Complexity*. Springer, 1998.

[19] M. Pauly. *Logic for Social Software*. Ph.D., ILLC Dissertation Series. University of Amsterdam, 2001.

[20] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[21] T. Rahwan and N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *Proc. AAMAS'08*, pages 1417–1420, 2008.

[22] T. Rahwan, T. Michalak, N. R. Jennings, M. Wooldridge, and P. McBurney. Coalition structure generation in multi-agent systems with positive and negative externalities. In *Proc. IJCAI*, 2009.

[23] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *AIJ*, 1-2(111):209–238, 1999.

[24] T. Sandholm and V.R. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94:99–137, 1997.

[25] İ. Seylan and W. Jamroga. Description logic for coalitions. In *Proc. AAMAS'09*, pages 425–432. AAAI, 2009.

[26] Y. Shoham and K. Leyton-Brown. *Multi Agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, 2009.

[27] F. Tohmé and T. Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic and Computation*, 9:793–815, 1999.

[28] W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artif. Intell.*, 164(1-2):81–119, 2005.

[29] J. Wu, C. Wang, L. Zhang, and J. Xie. Coalitional planning in game-like domains via ATL model checking. In *ICTAI '09: Proc. 21st Int. Conf. on Tools with AI*, pages 645–652, 2009.

Robotics and Learning

Active Markov Information-Theoretic Path Planning for Robotic Environmental Sensing

Kian Hsiang Low
Department of Computer Science
National University of Singapore
Republic of Singapore
lowkh@comp.nus.edu.sg

John M. Dolan and Pradeep Khosla
Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213 USA
jmd@cs.cmu.edu, pkk@ece.cmu.edu

ABSTRACT

Recent research in multi-robot exploration and mapping has focused on sampling environmental fields, which are typically modeled using the Gaussian process (GP). Existing information-theoretic exploration strategies for learning GP-based environmental field maps adopt the non-Markovian problem structure and consequently scale poorly with the length of history of observations. Hence, it becomes computationally impractical to use these strategies for *in situ*, real-time active sampling. To ease this computational burden, this paper presents a Markov-based approach to efficient information-theoretic path planning for active sampling of GP-based fields. We analyze the time complexity of solving the Markov-based path planning problem, and demonstrate analytically that it scales better than that of deriving the non-Markovian strategies with increasing length of planning horizon. For a class of exploration tasks called the transect sampling task, we provide theoretical guarantees on the active sampling performance of our Markov-based policy, from which ideal environmental field conditions and sampling task settings can be established to limit its performance degradation due to violation of the Markov assumption. Empirical evaluation on real-world temperature and plankton density field data shows that our Markov-based policy can generally achieve active sampling performance comparable to that of the widely-used non-Markovian greedy policies under less favorable realistic field conditions and task settings while enjoying significant computational gain over them.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Markov processes, stochastic processes; I.2.8 [**Problem Solving, Control Methods, and Search**]: Dynamic programming; I.2.9 [**Robotics**]: Autonomous vehicles

General Terms

Algorithms, Performance, Experimentation, Theory

Keywords

Multi-robot exploration and mapping, adaptive sampling, active learning, Gaussian process, non-myopic path planning

Cite as: Active Markov Information-Theoretic Path Planning for Robotic Environmental Sensing, Kian Hsiang Low, John M. Dolan, and Pradeep Khosla, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 753-760.

1. INTRODUCTION

Research in multi-robot exploration and mapping has recently progressed from building occupancy grids [14] to sampling spatially varying environmental phenomena [5, 6, 8], in particular, environmental fields (e.g., plankton density, pollutant concentration, temperature fields) that are characterized by *continuous-valued, spatially correlated* measurements (see Fig. 1). Exploration strategies for building occupancy grid maps usually operate under the assumptions of (a) *discrete*, (b) *independent* cell occupancies, which impose, respectively, the following limitations for learning environmental field maps: these strategies (a) cannot be fully informed by the continuous field measurements and (b) cannot exploit the spatial correlation structure of an environmental field for selecting observation paths. As a result, occupancy grid mapping strategies are not capable of selecting the most informative observation paths for learning an environmental field map.

Furthermore, occupancy grid mapping strategies typically assume that range sensing is available. In contrast, many *in situ* environmental and ecological sensing applications (e.g., monitoring of ocean phenomena, forest ecosystems, or pollution) permit only point-based sensing, thus making a high-resolution sampling of the entire field impractical in terms of resource costs (e.g., energy consumption, mission time). In practice, the resource cost constraints restrict the spatial coverage of the observation paths. Fortunately, the spatial correlation structure of an environmental field enables a map of the field (in particular, its unobserved areas) to be learned using the point-based observations taken along the resource-constrained paths. To learn this map, a commonly-used approach in spatial statistics [15] is to assume that the environmental field is realized from a probabilistic model called the *Gaussian process* (GP) (Section 3.2). More importantly, the GP model allows an environmental field to be formally characterized and consequently provides formal measures of mapping uncertainty (e.g., based on mean-squared error [5] or entropy criterion [6]) for directing a robot team to explore highly uncertain areas of the field. In this paper, we focus on using the entropy criterion to measure mapping uncertainty.

How then does a robot team plan the most informative resource-constrained observation paths to minimize the mapping uncertainty of an environmental field? To address this, the work of [6] has proposed an information-theoretic multi-robot exploration strategy that selects non-myopic observation paths with maximum entropy. Interestingly, this work has established an equivalence result that the maximum-entropy paths selected by such a strategy can achieve the

dual objective of minimizing the mapping uncertainty defined using the entropy criterion. When this strategy is applied to sampling a GP-based environmental field, it can be reduced to solving a non-Markovian, deterministic planning problem called the *information-theoretic multi-robot adaptive sampling problem* (iMASP) (Section 3). Due to the non-Markovian problem structure of iMASP, its state size grows exponentially with the length of planning horizon. To alleviate this computational difficulty, an anytime heuristic search algorithm called Learning Real-Time A* [1] is used to solve iMASP approximately. However, this algorithm does not guarantee the performance of its induced exploration policy. We have also observed through experiments that when the joint action space of the robot team is large or the planning horizon is long, it no longer produces a good policy fast enough. Even after incurring a huge amount of time and space to improve the search, its resulting policy still performs worse than the widely-used non-Markovian greedy policy, the latter of which can be derived efficiently by solving the myopic formulation of iMASP (Section 3.3).

Though the anytime and greedy algorithms provide some computational relief to solving iMASP (albeit approximately), they inherit iMASP's non-Markovian problem structure and consequently scale poorly with the length of history of observations. Hence, it becomes computationally impractical to use these non-Markovian path planning algorithms for *in situ*, real-time active sampling performed (a) at high resolution (e.g., due to high sensor sampling rate or large sampling region), (b) over dynamic features of interest (e.g., algal blooms, oil spills), (c) with resource cost constraints (e.g., energy consumption, mission time), or (d) in the presence of dynamically changing external forces translating the robots (e.g., ocean drift on autonomous boats), thus requiring fast replanning. For example, the deployment of autonomous underwater vehicles (AUVs) and boats for ocean sampling poses the above challenges/issues among others [3].

To ease this computational burden, this paper proposes a principled Markov-based approach to efficient information-theoretic path planning for active sampling of GP-based environmental fields (Section 4), which we develop by assuming the Markov property in iMASP planning. To the probabilistic robotics community, such a move to achieve time efficiency is probably anticipated. However, the Markov property is often imposed without carefully considering or formally analyzing its consequence on the performance degradation while operating in non-Markovian environments. In particular, to what extent does the environmental structure affect the performance degradation due to violation of the Markov assumption? Motivated by this lack of treatment, our work in this paper is novel in demonstrating both theoretically and empirically the extent of which the degradation of active sampling performance depends on the spatial correlation structure of an environmental field. An important practical consequence is that of establishing environmental field conditions under which the Markov-based approach performs well relative to the non-Markovian iMASP-based policy while enjoying significant computational gain over it. The specific contributions of our work include:

- analyzing the time complexity of solving the Markov-based information-theoretic path planning problem, and showing analytically that it scales better than that of deriving the non-Markovian strategies with increasing length of planning horizon (Section 4.1);
- providing theoretical guarantees on the active sampling performance of our Markov-based policy (Section 4.2) for a class of exploration tasks called the *transect sampling task* (Section 2), from which various ideal environmental field conditions and sampling task settings can be established to limit its performance degradation;
- empirically evaluating the active sampling performance and time efficiency of our Markov-based policy on real-world temperature and plankton density field data under less favorable realistic environmental field conditions and sampling task settings (Section 5).

2. TRANSECT SAMPLING TASK

Fig. 1 illustrates the transect sampling task introduced in [12, 13] previously. A temperature field is spatially distributed over a 25 m $\times$ 150 m transect that is discretized into a 5×30 grid of sampling locations comprising 30 columns, each of which has 5 sampling locations. It can be observed that the number of columns is much greater than the number of sampling locations in each column; this observed property is assumed to be consistent with every other transect. The robots are constrained to simultaneously explore forward one column at a time from the leftmost to the rightmost column of the transect such that each robot samples one location per column for a total of 30 locations. So, each robot's action space given its current location consists of moving to any of the 5 locations in the adjacent column on its right. The number of robots is assumed not to be larger than the number of sampling locations per column. We assume that an adversary chooses the starting robot locations in the leftmost column and the robots will only know them at the time of deployment; such an adversary can be the dynamically changing external forces translating the robots (e.g., ocean drift on autonomous boats) or the unknown obstacles occupying potential starting locations. The robots are allowed to end at any location in the rightmost column.

In practice, the constraint on exploring forward in a transect sampling task permits the planning of less complex observation paths that can be achieved more reliably, using less sophisticated control algorithms, and by robots with limited maneuverability (e.g., unmanned aerial vehicles, autonomous boats and AUVs [10]). For practical applications, while the robot is in transit from its current location to a distant planned waypoint [3, 13], this task can be performed to collect the most informative observations during transit. In monitoring of ocean phenomena and freshwater quality along rivers, the transect can span a plankton density or temperature field drifting at a constant rate from right to left and the autonomous boats are tasked to explore within a line perpendicular to the drift. As another example, the transect can be the bottom surface of ship hull or other maritime structure to be inspected and mapped by AUVs.

3. NON-MARKOVIAN PATH PLANNING

3.1 Notations and Preliminaries

Let $\mathcal{U}$ be the domain of the environmental field representing a set of sampling locations in the transect such that each location $u \in \mathcal{U}$ yields a measurement z_u. The columns of the transect are indexed in an increasing order from left to right with the leftmost column being indexed '0'. Each planning stage is associated with a column from which every robot in

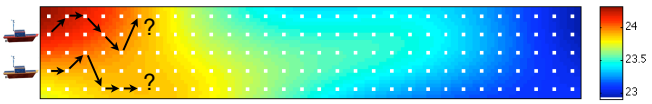


Figure 1: Transect sampling task on a temperature field (measured in °C) spatially distributed over a 25 m × 150 m transect that is discretized into a 5×30 grid of sampling locations (white dots).

the team selects and takes an observation (i.e., comprising a pair of location and its measurement). Let k denote the number of robots in the team. In each stage i, the team of k robots then collects from column i a total of k observations, which are denoted by a pair of vectors x_i of k locations and z_{x_i} of the corresponding measurements. Let $x_{0:i}$ and $z_{x_{0:i}}$ denote vectors comprising the histories of robots' sampling locations and corresponding measurements over stages 0 to i (i.e., concatenations of $x_0, x_1, \ldots, x_i$ and $z_{x_0}, z_{x_1}, \ldots, z_{x_i}$), respectively. Let Z_u, Z_{x_i}, and $Z_{x_{0:i}}$ be random measurements that are associated with the realizations z_u, z_{x_i}, and $z_{x_{0:i}}$, respectively.

3.2 GP-Based Environmental Field

The GP model can be used to formally characterize an environmental field as follows: the environmental field is defined to vary as a realization of a GP. Let $\{Z_u\}_{u \in \mathcal{U}}$ denote a GP, i.e., every finite subset of $\{Z_u\}_{u \in \mathcal{U}}$ has a multivariate Gaussian distribution [9]. The GP is fully specified by its mean $\mu_u \triangleq \mathbb{E}[Z_u]$ and covariance $\sigma_{uv} \triangleq \mathrm{cov}[Z_u, Z_v]$ for all $u, v \in \mathcal{U}$. We assume that the GP is second-order stationary, i.e., it has a constant mean and a stationary covariance structure (i.e., σ_{uv} is a function of $u - v$ for all $u, v \in \mathcal{U}$). In particular, its covariance structure is defined by the widely-used squared exponential covariance function [9]

$$\sigma_{uv} \triangleq \sigma_s^2 \exp\left\{-\frac{1}{2}(u-v)^\top M^{-2}(u-v)\right\} + \sigma_n^2 \delta_{uv} \qquad (1)$$

where σ_s^2 is the signal variance, σ_n^2 is the noise variance, M is a diagonal matrix with length-scale components ℓ_1 and ℓ_2 in the horizontal and vertical directions of a transect, respectively, and δ_{uv} is a Kronecker delta of value 1 if $u = v$, and 0 otherwise. Intuitively, the signal and noise variances describe, respectively, the intensity and noise of the field measurements while the length-scale can be interpreted as the approximate distance to be traversed in a transect for the field measurement to change considerably [9]; it therefore controls the degree of spatial correlation or "similarity" between field measurements. In this paper, the mean and covariance structure of the GP are assumed to be known. Given that the robot team has collected observations $x_0, z_{x_0}, x_1, z_{x_1}, \ldots, x_i, z_{x_i}$ over stages 0 to i, the distribution of Z_u remains Gaussian with the following posterior mean and covariance

$$\mu_{u|x_{0:i}} = \mu_u + \Sigma_{ux_{0:i}} \Sigma_{x_{0:i}x_{0:i}}^{-1} \{z_{x_{0:i}} - \mu_{x_{0:i}}\}^\top \qquad (2)$$

$$\sigma_{uv|x_{0:i}} = \sigma_{uv} - \Sigma_{ux_{0:i}} \Sigma_{x_{0:i}x_{0:i}}^{-1} \Sigma_{x_{0:i}v} \qquad (3)$$

where $\mu_{x_{0:i}}$ is a row vector with mean components μ_w for every location w of $x_{0:i}$, $\Sigma_{ux_{0:i}}$ is a row vector with covariance components σ_{uw} for every location w of $x_{0:i}$, $\Sigma_{x_{0:i}v}$ is a column vector with covariance components σ_{wv} for every location w of $x_{0:i}$, and $\Sigma_{x_{0:i}x_{0:i}}$ is a covariance matrix with components σ_{wy} for every pair of locations w, y of $x_{0:i}$. Note that the posterior mean $\mu_{u|x_{0:i}}$ (2) is the best unbiased predictor of the measurement z_u at unobserved location u. An important property of GP is that the posterior covariance $\sigma_{uv|x_{0:i}}$ (3) is independent of the observed measurements $z_{x_{0:i}}$; this property is used to reduce iMASP to a deterministic planning problem as shown later.

3.3 Deterministic iMASP Planning

Supposing the robot team starts in locations x_0 of leftmost column 0, an exploration policy is responsible for directing it to sample locations $x_1, x_2, \ldots, x_{t+1}$ of the respective columns $1, 2, \ldots, t+1$ to form the observation paths. Formally, a non-Markovian policy is denoted by $\pi \triangleq \langle \pi_0(x_{0:0} = x_0), \pi_1(x_{0:1}), \ldots, \pi_t(x_{0:t}) \rangle$ where $\pi_i(x_{0:i})$ maps the history $x_{0:i}$ of robots' sampling locations to a vector $a_i \in \mathcal{A}(x_i)$ of robots' actions in stage i (i.e., $a_i \leftarrow \pi_i(x_{0:i})$), and $\mathcal{A}(x_i)$ is the joint action space of the robots given their current locations x_i. We assume that the transition function $\tau(x_i, a_i)$ *deterministically* (i.e., no localization uncertainty) moves the robots to their next locations x_{i+1} in stage $i+1$ (i.e., $x_{i+1} \leftarrow \tau(x_i, a_i)$). Putting π_i and τ together yields the assignment $x_{i+1} \leftarrow \tau(x_i, \pi_i(x_{0:i}))$.

The work of [6] has proposed a non-Markovian policy π^* that selects non-myopic observation paths with maximum entropy for sampling a GP-based field. To know how π^* is derived, we first define the value under a policy π to be the entropy of observation paths when starting in x_0 and following π thereafter:

$$\begin{aligned} V_0^\pi(x_0) &\triangleq \mathbb{H}[Z_{x_{1:t+1}} | Z_{x_0}, \pi] \\ &= -\int f(z_{x_{0:t+1}} | \pi) \log f(z_{x_{1:t+1}} | z_{x_0}, \pi) \, dz_{x_{0:t+1}} \end{aligned} \qquad (4)$$

where f denotes a Gaussian probability density function. When a non-Markovian policy π is plugged into (4), the following $(t+1)$-stage recursive formulation results from the chain rule for entropy and $x_{i+1} \leftarrow \tau(x_i, \pi_i(x_{0:i}))$:

$$\begin{aligned} V_i^\pi(x_{0:i}) &= \mathbb{H}[Z_{x_{i+1}} | Z_{x_{0:i}}, \pi_i] + V_{i+1}^\pi(x_{0:i+1}) \\ &= \mathbb{H}[Z_{\tau(x_i, \pi_i(x_{0:i}))} | Z_{x_{0:i}}] + V_{i+1}^\pi((x_{0:i}, \tau(x_i, \pi_i(x_{0:i})))) \\ V_t^\pi(x_{0:t}) &= \mathbb{H}[Z_{x_{t+1}} | Z_{x_{0:t}}, \pi_t] \\ &= \mathbb{H}[Z_{\tau(x_t, \pi_t(x_{0:t}))} | Z_{x_{0:t}}] \end{aligned} \qquad (5)$$

for stage $i = 0, \ldots, t-1$ such that each stagewise posterior entropy (i.e., of the measurements $Z_{x_{i+1}}$ to be observed in stage $i+1$ given the history of measurements $Z_{x_{0:i}}$ observed from stages 0 to i) reduces to

$$\mathbb{H}[Z_{x_{i+1}} | Z_{x_{0:i}}] = \frac{1}{2} \log (2\pi e)^k |\Sigma_{x_{i+1}|x_{0:i}}| \qquad (6)$$

where $\Sigma_{x_{i+1}|x_{0:i}}$ is a covariance matrix with components $\sigma_{uv|x_{0:i}}$ for every pair of locations u, v of x_{i+1}, each of which is independent of observed measurements $z_{x_{0:i}}$ by (3), as discussed above. So, $\mathbb{H}[Z_{x_{i+1}} | Z_{x_{0:i}}]$ can be evaluated in closed form, and the value functions (5) only require the history of robots' sampling locations $x_{0:i}$ as inputs but not that of corresponding measurements $z_{x_{0:i}}$.

Solving iMASP involves choosing π to maximize $V_0^\pi(x_0)$ (5), which yields the optimal policy π^*. Plugging π^* into (5) gives the $(t+1)$-stage dynamic programming equations:

$$\begin{aligned} V_i^{\pi^*}(x_{0:i}) &= \max_{a_i \in \mathcal{A}(x_i)} \mathbb{H}[Z_{\tau(x_i, a_i)} | Z_{x_{0:i}}] + V_{i+1}^{\pi^*}((x_{0:i}, \tau(x_i, a_i))) \\ V_t^{\pi^*}(x_{0:t}) &= \max_{a_t \in \mathcal{A}(x_t)} \mathbb{H}[Z_{\tau(x_t, a_t)} | Z_{x_{0:t}}] \end{aligned} \qquad (7)$$

for stage $i = 0, \ldots, t-1$. Since each stagewise posterior entropy $\mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{x_{0:i}}]$ (6) can be evaluated in closed form as explained above, iMASP for sampling the GP-based field (7) reduces to a deterministic planning problem. Furthermore, it turns out to be the well-known maximum entropy sampling problem [11] as demonstrated in [6]. Policy $\pi^* = \langle \pi_0^*(x_{0:0}), \ldots, \pi_t^*(x_{0:t})\rangle$ can be determined by

$$\begin{aligned}\pi_i^*(x_{0:i}) &= \arg\max_{a_i \in \mathcal{A}(x_i)} \mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{x_{0:i}}] + V_{i+1}^{\pi^*}((x_{0:i}, \tau(x_i,a_i)))\\ \pi_t^*(x_{0:t}) &= \arg\max_{a_t \in \mathcal{A}(x_t)} \mathbb{H}[Z_{\tau(x_t,a_t)}|Z_{x_{0:t}}]\end{aligned} \tag{8}$$

for stage $i = 0, \ldots, t-1$. Similar to the optimal value functions (7), π^* only requires the history of robots' sampling locations as inputs. So, π^* can generate the maximum-entropy paths prior to exploration.

Solving the myopic formulation of iMASP (7) is often considered to ease computation (Section 4.1), which entails deriving the non-Markovian greedy policy $\pi^G = \langle \pi_0^G(x_{0:0}), \ldots, \pi_t^G(x_{0:t})\rangle$ where, for stage $i = 0, \ldots, t$,

$$\pi_i^G(x_{0:i}) = \arg\max_{a_i \in \mathcal{A}(x_i)} \mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{x_{0:i}}] \ . \tag{9}$$

The work of [2] has proposed a non-Markovian greedy policy $\pi^M = \langle \pi_0^M(x_{0:0}), \ldots, \pi_t^M(x_{0:t})\rangle$ to approximately maximize the closely related mutual information criterion:

$$\pi_i^M(x_{0:i}) = \arg\max_{a_i \in \mathcal{A}(x_i)} \mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{x_{0:i}}] - \mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{\overline{x}_{0:i+1}}] \tag{10}$$

for stage $i = 0, \ldots, t$ where $\overline{x}_{0:i+1}$ denotes the vector comprising locations of domain $\mathcal{U}$ not found in $(x_{0:i}, \tau(x_i, a_i))$. It is shown in [2] that π^M greedily selects new sampling locations that maximize the increase in mutual information. As noted in [6], this strategy is deficient in that it may not necessarily minimize the mapping uncertainty defined using the entropy criterion. More importantly, it suffers a huge computational drawback: the time needed to derive π^M depends on the map resolution (i.e., $|\mathcal{U}|$) (Section 4.1).

4. MARKOV-BASED PATH PLANNING

The Markov property assumes that the measurements $Z_{x_{i+1}}$ to be observed next in stage $i+1$ depends only on the current measurements Z_{x_i} observed in stage i and is conditionally independent of the past measurements $Z_{x_{0:i-1}}$ observed from stages 0 to $i-1$. That is, $f(z_{x_{i+1}}|z_{x_{0:i}}) = f(z_{x_{i+1}}|z_{x_i})$ for all $z_{x_0}, z_{x_1}, \ldots, z_{x_{i+1}}$. As a result, $\mathbb{H}[Z_{x_{i+1}}|Z_{x_{0:i}}]$ (6) can be approximated by $\mathbb{H}[Z_{x_{i+1}}|Z_{x_i}]$. It is therefore straightforward to impose the Markov assumption on iMASP (7), which yields the following dynamic programming equations for the Markov-based path planning problem:

$$\begin{aligned}\widetilde{V}_i(x_i) &= \max_{a_i \in \mathcal{A}(x_i)} \mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{x_i}] + \widetilde{V}_{i+1}(\tau(x_i,a_i))\\ \widetilde{V}_t(x_t) &= \max_{a_t \in \mathcal{A}(x_t)} \mathbb{H}[Z_{\tau(x_t,a_t)}|Z_{x_t}] \ .\end{aligned} \tag{11}$$

for stage $i = 0, \ldots, t-1$. Consequently, the Markov-based policy $\widetilde{\pi} = \langle \widetilde{\pi}_0(x_0), \ldots, \widetilde{\pi}_t(x_t)\rangle$ can be determined by

$$\begin{aligned}\widetilde{\pi}_i(x_i) &= \arg\max_{a_i \in \mathcal{A}(x_i)} \mathbb{H}[Z_{\tau(x_i,a_i)}|Z_{x_i}] + \widetilde{V}_{i+1}(\tau(x_i,a_i))\\ \widetilde{\pi}_i(x_t) &= \arg\max_{a_t \in \mathcal{A}(x_t)} \mathbb{H}[Z_{\tau(x_t,a_t)}|Z_{x_t}] \ .\end{aligned} \tag{12}$$

4.1 Time Complexity: Analysis & Comparison

THEOREM 1. *Let $\mathcal{A} \triangleq \mathcal{A}(x_0) = \ldots = \mathcal{A}(x_t)$. Deriving the Markov-based policy $\widetilde{\pi}$ (12) for the transect sampling task requires $\mathcal{O}(|\mathcal{A}|^2(t+k^4))$ time.*

Note that $|\mathcal{A}| = {}^r\mathrm{C}_k = \mathcal{O}(r^k)$ where r is the number of sampling locations per column and $k \leq r$ as assumed in Section 2. Though $|\mathcal{A}|$ is exponential in the number k of robots, r is expected to be small in a transect, which prevents $|\mathcal{A}|$ from growing too large.

In contrast, deriving iMASP-based policy π^* (8) requires $\mathcal{O}(|\mathcal{A}|^t t^2 k^4)$ time. Deriving greedy policies π^G (9) and π^M (10) incur, respectively, $\mathcal{O}(|\mathcal{A}|t^4k^3 + |\mathcal{A}|^2tk^4)$ and $\mathcal{O}(|\mathcal{A}|t|\mathcal{U}|^3 + |\mathcal{A}|^2tk^4) = \mathcal{O}(|\mathcal{A}|t^4r^3 + |\mathcal{A}|^2tk^4)$ time to compute the observation paths over all $|\mathcal{A}|$ possible choices of starting robot locations. Clearly, all the non-Markovian strategies do not scale as well as our Markov-based approach with increasing length $t+1$ of planning horizon or number $t+2$ of columns, which is expected to be large. As demonstrated empirically (Section 5), the Markov-based policy $\widetilde{\pi}$ can be derived faster than π^G and π^M by more than an order of magnitude; this computational advantage is boosted further for transect sampling tasks with unknown starting robot locations.

4.2 Performance Guarantees

We will first provide a theoretical guarantee on how the Markov-based policy $\widetilde{\pi}$ (12) performs relative to the non-Markovian iMASP-based policy π^* (8) for the case of 1 robot. This key result follows from our intuition that when the horizontal spatial correlation becomes small, exploiting the past measurements for path planning should hardly improve the active sampling performance in a transect sampling task, thus favoring the Markov-based policy. Though this intuition is simple, supporting it with formal theoretical results (and their corresponding proofs reported elsewhere [7]) turns out to be non-trivial as shown below.

Recall the Markov assumption that $\mathbb{H}[Z_{x_{i+1}}|Z_{x_{0:i}}]$ (6) is to be approximated by $\mathbb{H}[Z_{x_{i+1}}|Z_{x_i}]$. This prompts us to first consider bounding the difference of these posterior entropies that ensues from the Markov property:

$$\begin{aligned}\mathbb{H}[Z_{x_{i+1}}|Z_{x_i}] - \mathbb{H}[Z_{x_{i+1}}|Z_{x_{0:i}}] &= \frac{1}{2}\log\frac{\sigma^2_{x_{i+1}|x_i}}{\sigma^2_{x_{i+1}|x_{0:i}}}\\ &= \frac{1}{2}\log\left(1 - \frac{\sigma^2_{x_{i+1}|x_i} - \sigma^2_{x_{i+1}|x_{0:i}}}{\sigma^2_{x_{i+1}|x_i}}\right)^{-1} \geq 0 \ .\end{aligned} \tag{13}$$

This difference can be interpreted as the reduction in uncertainty of the measurements $Z_{x_{i+1}}$ to be observed next in stage $i+1$ by observing the past measurements $Z_{x_{0:i-1}}$ from stages 0 to $i-1$ given the current measurements Z_{x_i} observed in stage i. This difference is small if $Z_{x_{0:i-1}}$ does not contribute much to the reduction in uncertainty of $Z_{x_{i+1}}$ given Z_{x_i}. It (13) is often known as the conditional mutual information of $Z_{x_{i+1}}$ and $Z_{x_{0:i-1}}$ given Z_{x_i} denoted by

$$\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}] \triangleq \mathbb{H}[Z_{x_{i+1}}|Z_{x_i}] - \mathbb{H}[Z_{x_{i+1}}|Z_{x_{0:i}}] \ ,$$

which is of value 0 if the Markov property holds.

The results to follow assume that the transect is discretized into a grid of sampling locations. Let ω_1 and ω_2 denote the horizontal and vertical grid discretization widths (i.e., separations between adjacent sampling locations), respectively. Let $\ell_1' \triangleq \ell_1/\omega_1$ and $\ell_2' \triangleq \ell_2/\omega_2$ represent the normalized horizontal and vertical length-scale components, respectively.

The following lemma bounds the variance reduction term $\sigma^2_{x_{i+1}|x_i} - \sigma^2_{x_{i+1}|x_{0:i}}$ in (13):

LEMMA 2. *Let* $\xi \triangleq \exp\left\{-\frac{1}{2\ell_1'^2}\right\}$ *and* $\rho \triangleq 1 + \frac{\sigma_n^2}{\sigma_s^2}$. *If* $\xi < \frac{\rho}{i}$, *then* $0 \leq \sigma^2_{x_{i+1}|x_i} - \sigma^2_{x_{i+1}|x_{0:i}} \leq \frac{\sigma_s^2\xi^4}{\frac{\rho}{i} - \xi}$.

The next lemma is fundamental to the subsequent results on the active sampling performance of Markov-based policy $\widetilde{\pi}$. It provides bounds on $\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}]$, which follow immediately from (13), Lemma 2, and the lower bound $\sigma^2_{x_{i+1}|x_i} = \sigma^2_{x_{i+1}} - (\sigma_{x_{i+1}x_i})^2/\sigma^2_{x_i} \geq \sigma_s^2 + \sigma_n^2 - \sigma_s^2\xi^2$:

LEMMA 3. *If* $\xi < \frac{\rho}{i}$, *then* $0 \leq \mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}] \leq \Delta(i)$ *where* $\Delta(i) \triangleq \frac{1}{2}\log\left(1 - \frac{\xi^4}{(\frac{\rho}{i} - \xi)(\rho - \xi^2)}\right)^{-1}$.

REMARK. *If* $j \leq s$, *then* $\Delta(j) \leq \Delta(s)$ *for* $j, s = 0, \ldots, t$.

From Lemma 3, since $\Delta(i)$ bounds $\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}]$ from above, a small $\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}]$ can be guaranteed by making $\Delta(i)$ small. From the definition of $\Delta(i)$, there are a few ways to achieve a small $\Delta(i)$: (a) $\Delta(i)$ depends on ℓ_1' through ξ. As $\ell_1' \to 0^+$, $\xi \to 0^+$, by definition. Consequently, $\Delta(i) \to 0^+$. A small ℓ_1' can be obtained using a small ℓ_1 and/or a large ω_1, by definition; (b) $\Delta(i)$ also depends on the noise-to-signal ratio σ_n^2/σ_s^2 through ρ. Raising σ_n^2 or lowering σ_s^2 increases ρ, by definition. This, in turn, decreases $\Delta(i)$; (c) Since i indicates the length of history of observations, the remark after Lemma 3 tells us that a shorter length produces a smaller $\Delta(i)$. To summarize, (a) environmental field conditions such as smaller horizontal spatial correlation and noisy, less intense fields, and (b) sampling task settings such as larger horizontal grid discretization width and shorter length of history of observations all contribute to smaller $\Delta(i)$, and hence smaller $\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}]$. This analysis is important for understanding the practical implication of our theoretical results later. A limitation with using Lemma 3 is that of the sufficient condition $\xi < \rho/i$, which will hold if the field conditions and task settings realized above to make $\Delta(i)$ small are adequately satisfied.

The following theorem uses the induced optimal value $\widetilde{V}_0(x_0)$ from solving the Markov-based path planning problem (11) to bound the maximum entropy $V_0^{\pi^*}(x_0)$ of observation paths achieved by π^* from solving iMASP (7):

THEOREM 4. *Let* $\epsilon_i \triangleq \sum_{s=i}^{t}\Delta(s) \leq (t-i+1)\Delta(t)$. *If* $\xi < \frac{\rho}{t}$, *then* $\widetilde{V}_i(x_i) - \epsilon_i \leq V_i^{\pi^*}(x_{0:i}) \leq \widetilde{V}_i(x_i)$ *for* $i = 0, \ldots, t$.

The above result is useful in providing an efficient way of knowing the maximum entropy $V_0^{\pi^*}(x_0)$, albeit approximately: the time needed to derive the two-sided bounds on $V_0^{\pi^*}(x_0)$ is linear in the length of planning horizon (Theorem 1) as opposed to exponential time required to compute the exact value of $V_0^{\pi^*}(x_0)$. Since the error bound ϵ_i is defined as a sum of $\Delta(s)$'s, we can rely on the above analysis of $\Delta(s)$ (see paragraph after Lemma 3) to improve this error bound: (a) environmental field conditions such as smaller horizontal spatial correlation and noisy, less intense fields, and (b) sampling task settings such as larger horizontal grid discretization width and shorter planning horizon (i.e., fewer transect columns) all improve this error bound.

In the main result below, the Markov-based policy $\widetilde{\pi}$ is guaranteed to achieve an entropy $V_0^{\widetilde{\pi}}(x_0)$ of observation paths (i.e., by plugging $\widetilde{\pi}$ into (5)) that is not more than ϵ_0 from the maximum entropy $V_0^{\pi^*}(x_0)$ of observation paths achieved by policy π^*:

THEOREM 5. *If* $\xi < \frac{\rho}{t}$, *then policy* $\widetilde{\pi}$ *is* ϵ_0*-optimal in achieving the maximum-entropy criterion, i.e.,* $V_0^{\pi^*}(x_0) - V_0^{\widetilde{\pi}}(x_0) \leq \epsilon_0$.

Again, since the error bound ϵ_0 is defined as a sum of $\Delta(s)$'s, we can use the above analysis of $\Delta(s)$ to improve this bound: (a) environmental field conditions such as smaller horizontal spatial correlation and noisy, less intense fields, and (b) sampling task settings such as larger horizontal grid discretization width and shorter planning horizon (i.e., fewer transect columns) all result in smaller ϵ_0, and hence improve the active sampling performance of Markov-based policy $\widetilde{\pi}$ relative to that of non-Markovian iMASP-based policy π^*. This not only supports our prior intuition (see first paragraph of this section) but also identifies other means of limiting the performance degradation of the Markov-based policy.

For the multi-robot case, a condition has to be imposed on the covariance structure of GP to obtain a similar guarantee:

$$|\sigma_{uv|x_{0:i}}| \leq |\sigma_{uv|x_m}| \tag{14}$$

for $m = 0, \ldots, i$ and any $u, v, x_0, x_1, \ldots, x_i \in \mathcal{U}$. Intuitively, (14) says that further conditioning does not make Z_u and Z_v more correlated. Note that (14) is satisfied if $u = v$.

Similar to Lemma 3 for the 1-robot case, we can bound $\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}]$ for the multi-robot case but tighter conditions have to be satisfied:

LEMMA 6. *Let* $\ell_1' = \ell_2'$. *If* $\xi < \min(\frac{\rho}{ik}, \frac{\rho}{4k})$ *and* (14) *is satisfied, then* $0 \leq \mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}] \leq \Delta_k(i)$ *where* $\Delta_k(i) \triangleq \frac{k}{2}\log\left(1 - \frac{\xi^4}{(\frac{\rho}{ik} - \xi)(\rho - \frac{4k}{\rho}\xi^2)}\right)^{-1}$.

To improve the upper bound $\Delta_k(i)$, the above analysis of $\Delta(i)$ can be applied here as these two upper bounds are largely similar: (a) environmental field conditions such as smaller spatial correlation and noisy, less intense fields, and (b) sampling task settings such as larger grid discretization width and shorter planning horizon (i.e., fewer transect columns) all entail smaller $\Delta_k(i)$. Decreasing the number k of robots also reduces $\Delta_k(i)$, thus yielding tighter bounds on $\mathbb{I}[Z_{x_{i+1}}; Z_{x_{0:i-1}}|Z_{x_i}]$. Using Lemma 6, we can derive guarantees similar to that of Theorems 4 and 5 on the performance of Markov-based policy $\widetilde{\pi}$ for the multi-robot case.

5. EXPERIMENTS AND DISCUSSION

In Section 4.2, we have highlighted the practical implication of our main theoretical result (i.e., Theorem 5), which establishes various environmental field conditions and sampling task settings to limit the performance degradation of Markov-based policy $\widetilde{\pi}$. This result, however, does not reveal whether $\widetilde{\pi}$ performs well (or not) under "seemingly" less favorable field conditions and task settings that do not jointly satisfy its sufficient condition $\xi < \rho/(tk)$. These include large spatial correlation, less noisy, highly intense fields, small grid discretization width, long planning horizon (i.e., many transect columns), and large number of robots. So, this section evaluates the active sampling performance

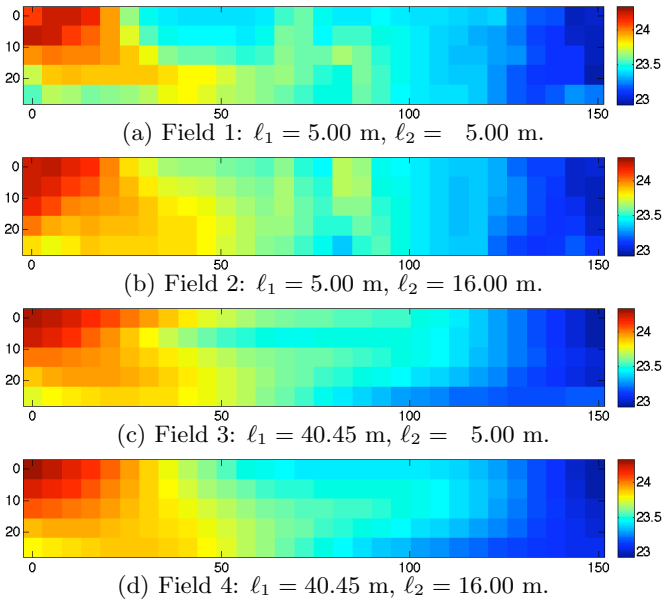
and time efficiency of $\widetilde{\pi}$ empirically on two real-world datasets under such field conditions and task settings as detailed below: (a) May 2009 temperature field data of Panther Hollow Lake in Pittsburgh, PA spanning 25 m by 150 m, and (b) June 2009 plankton density field data of Chesapeake Bay spanning 314 m by 1765 m.

Using maximum likelihood estimation (MLE) [9], the learned hyperparameters (i.e., horizontal and vertical length-scales, signal and noise variances) are, respectively, $\ell_1 = 40.45$ m, $\ell_2 = 16.00$ m, $\sigma_s^2 = 0.1542$, and $\sigma_n^2 = 0.0036$ for the temperature field, and $\ell_1 = 27.53$ m, $\ell_2 = 134.64$ m, $\sigma_s^2 = 2.152$, and $\sigma_n^2 = 0.041$ for the plankton density field. It can be observed that the temperature and plankton density fields have low noise-to-signal ratios σ_n^2/σ_s^2 of 0.023 and 0.019, respectively. Relative to the size of transect, both fields have large vertical spatial correlations, but only the temperature field has large horizontal spatial correlation.

The performance of Markov-based policy $\widetilde{\pi}$ is compared to non-Markovian policies produced by two state-of-the-art information-theoretic exploration strategies: greedy policies π^G (9) and π^M (10) proposed by [6] and [2], respectively. The non-Markovian policy π^* that has to be derived approximately using Learning Real-Time A* is excluded from comparison due to the reason provided in Section 1.

5.1 Performance Metrics

The tested policies are evaluated using the two metrics proposed in [6], which quantify the mapping uncertainty of the unobserved areas of the field differently: (a) The ENT(π) metric measures the posterior joint entropy $\mathbb{H}[Z_{\overline{x}_{0:t+1}}|Z_{x_{0:t+1}}]$ of field measurements $Z_{\overline{x}_{0:t+1}}$ at unobserved locations $\overline{x}_{0:t+1}$ where $\overline{x}_{0:t+1}$ denotes the vector comprising locations of domain $\mathcal{U}$ not found in the sampled locations $x_{0:t+1}$ selected by policy π. Smaller ENT(π) implies lower mapping uncertainty; (b) The ERR(π) metric measures the mean-squared relative error $|\mathcal{U}|^{-1}\sum_{u\in\mathcal{U}}\{(z_u-\mu_{u|x_{0:t+1}})/\bar{\mu}\}^2$ resulting from using the observations (i.e., sampled locations $x_{0:t+1}$ and corresponding measurements $z_{x_{0:t+1}}$) selected by policy π and the posterior mean $\mu_{u|x_{0:t+1}}$ (2) to predict the field where $\bar{\mu} = |\mathcal{U}|^{-1}\sum_{u\in\mathcal{U}} z_u$. Smaller ERR($\pi$) implies higher prediction accuracy. Two noteworthy differences distinguish these metrics: (a) The ENT(π) metric exploits the spatial correlation between field measurements in the unobserved areas whereas the ERR(π) metric implicitly assumes independence between them. As a result, unlike the ERR(π) metric, the ENT(π) metric does not overestimate the mapping uncertainty. To illustrate this, suppose the unknown field measurements are restricted to only two unobserved locations u and v residing in a highly uncertain area and they are highly correlated due to spatial proximity. The behavior of the ENT(π) metric can be understood upon applying the chain rule for entropy (i.e., ENT(π) $= \mathbb{H}[Z_u, Z_v|Z_{x_{0:t+1}}] = \mathbb{H}[Z_u|Z_{x_{0:t+1}}] + \mathbb{H}[Z_v|Z_{x_{0:t+1}}, Z_u]$); the latter uncertainty term (i.e., posterior entropy of Z_v) is significantly reduced or "discounted" due to the high spatial correlation between Z_u and Z_v. Hence, the mapping uncertainty of these two unobserved locations is not overestimated. A practical advantage of this metric is that it does not overcommit sensing resources; in the simple illustration above, a single observation at either location u or v suffices to learn both field measurements well. On the other hand, the ERR(π) metric considers each location to be of high uncertainty due to the independence assumption; (b) In contrast to the ENT(π) metric, the ERR(π) metric can use ground truth measurements to evaluate if the field is being mapped accurately. Let ENTD(π) $\triangleq$ ENT($\widetilde{\pi}$)$-$ENT(π) and ERRD(π) $\triangleq$ ERR($\widetilde{\pi}$)$-$ERR(π). Decreasing ENTD(π) improves the ENT($\widetilde{\pi}$) performance of $\widetilde{\pi}$ relative to that of π. Small |ENTD(π)| implies that $\widetilde{\pi}$ achieves ENT($\widetilde{\pi}$) performance comparable to that of π. ERRD(π) can be interpreted likewise. Additionally, we will consider the time taken to derive each policy as the third metric.



(a) Field 1: $\ell_1 = 5.00$ m, $\ell_2 = 5.00$ m.

(b) Field 2: $\ell_1 = 5.00$ m, $\ell_2 = 16.00$ m.

(c) Field 3: $\ell_1 = 40.45$ m, $\ell_2 = 5.00$ m.

(d) Field 4: $\ell_1 = 40.45$ m, $\ell_2 = 16.00$ m.

Figure 2: Temperature fields (measured in °C) with varying horizontal length-scale ℓ_1 and vertical length-scale ℓ_2.

5.2 Temperature Field Data

We will first investigate how varying spatial correlations (i.e., varying length-scales) of the temperature field affect the ENT(π) and ERR(π) performance of evaluated policies. The temperature field is discretized into a 5×30 grid of sampling locations as shown in Figs. 1 and 2d. The horizontal and/or vertical length-scales of the original field (i.e., field 4 in Fig. 2d) are reduced to produce modified fields 1, 2, and 3 (respectively, Figs. 2a, 2b, and 2c); we fix these reduced length-scales while learning the remaining hyperparameters (i.e., signal and noise variances) through MLE.

Table 1 shows the results of mean ENT(π) and ERR(π) performance of tested policies (i.e., averaged over all possible starting robot locations) with varying length-scales and number of robots. The ENT(π) and ERR(π) for all policies generally decrease with increasing length-scales (except ERR($\widetilde{\pi}$) for 1 robot from field 2 to 4) due to increasing spatial correlation between measurements, thus resulting in lower mapping uncertainty.

For the case of 1 robot, the observations are as follows: (a) When ℓ_2 is kept constant (i.e., at 5 m or 16 m), reducing ℓ_1 from 40.45 m to 5 m (i.e., from field 3 to 1 or field 4 to 2) decreases ENTD(π^G), ERRD(π^G), ENTD(π^M), and ERRD(π^M): when the horizontal correlation becomes small, it can no longer be exploited by the non-Markovian policies π^G and π^M; (b) For field 3 with large ℓ_1 and small ℓ_2, ENTD(π^G) and ENTD(π^M) are large as the Markov property of $\widetilde{\pi}$ prevents it from exploiting the large horizontal

Table 1: Comparison of ENT(π) (left) and ERR(π) ($\times 10^{-5}$) (right) performance for temperature fields that are discretized into 5×30 grids (Fig. 2).

	ENT(π)				ERR(π)			
1 robot	Field				Field			
Policy	1	2	3	4	1	2	3	4
$\widetilde{\pi}$	-83	-246	-543	-597	3.7040	0.5713	2.3680	0.5754
π^G	-82	-246	-554	-598	1.8680	0.5713	0.0801	0.0252
π^M	-80	-211	-554	-596	1.8433	0.5212	0.0701	0.0421
2 robots	Field				Field			
Policy	1	2	3	4	1	2	3	4
$\widetilde{\pi}$	-71	-190	-380	-422	0.3797	0.2101	0.1171	0.0095
π^G	-72	-190	-382	-425	0.3526	0.2101	0.0150	0.0087
π^M	-68	-131	-382	-421	0.6714	0.1632	0.0148	0.0086
3 robots	Field				Field			
Policy	1	2	3	4	1	2	3	4
$\widetilde{\pi}$	-53	-109	-232	-297	0.1328	0.0068	0.0063	0.0031
π^G	-53	-109	-215	-297	0.1312	0.0068	0.0059	0.0031
π^M	-53	-73	-214	-255	0.1080	0.1397	0.0055	0.0030

Table 2: Comparison of ENT(π) (left) and ERR(π) ($\times 10^{-5}$) (right) performance for temperature field that is discretized into 13×75 grid.

ENT(π)	Number k of robots		
Policy	1	2	3
$\widetilde{\pi}$	-4813	-4284	-3828
π^G	-4813	-4286	-3841
π^M	-4808	-4277	-3825

ERR(π)	Number k of robots		
Policy	1	2	3
$\widetilde{\pi}$	1.0287	0.0032	0.0015
π^G	0.0082	0.0030	0.0024
π^M	0.0087	0.0034	0.0019

correlation; (c) When ℓ_1 is kept constant (i.e., at 5 m or 40.45 m), reducing ℓ_2 from 16 m to 5 m (i.e., from field 2 to 1 or field 4 to 3) increases ERRD(π^G) and ERRD(π^M): when vertical correlation becomes small, it can no longer be exploited by $\widetilde{\pi}$, thus incurring larger ERR($\widetilde{\pi}$).

For the case of 2 robots, the observations are as follows: (a) |ENTD(π^G)| and |ENTD(π^M)| are small for all fields except for field 2 where $\widetilde{\pi}$ significantly outperforms π^M. In particular, when ℓ_2 is kept constant (i.e., at 5 m or 16 m), reducing ℓ_1 from 40.45 m to 5 m (i.e., from field 3 to 1 or field 4 to 2) decreases ENTD(π^G), ENTD(π^M), and ERRD(π^G): this is explained in the first observation of 1-robot case; (b) For field 3 with large ℓ_1 and small ℓ_2, ERRD(π^G) and ERRD(π^M) are large: this is explained in the second and third observations of 1-robot case; (c) When ℓ_1 is kept constant (i.e., at 5 m or 40.45 m), reducing ℓ_2 from 16 m to 5 m (i.e., from field 2 to 1 or field 4 to 3) increases ERRD(π^G): this is explained in the third observation of 1-robot case. This also holds for ERRD(π^M) when ℓ_1 is large.

For the case of 3 robots, it can be observed that $\widetilde{\pi}$ can achieve ENT($\widetilde{\pi}$) and ERR($\widetilde{\pi}$) performance comparable to (if not, better than) that of π^G and π^M for all fields.

To summarize the above observations on spatial correlation conditions favoring $\widetilde{\pi}$ over π^G and π^M, $\widetilde{\pi}$ can achieve ENT($\widetilde{\pi}$) performance comparable to (if not, better than) that of π^G and π^M for all fields with any number of robots except for field 3 (i.e., of large ℓ_1 and small ℓ_2) with 1 robot as explained previously. Policy $\widetilde{\pi}$ can achieve comparable ERR($\widetilde{\pi}$) performance for field 2 (i.e., of small ℓ_1 and large ℓ_2) with 1 robot because $\widetilde{\pi}$ is capable of exploiting the large vertical correlation, and the small horizontal correlation cannot be exploited by π^G and π^M. Policy $\widetilde{\pi}$ can also achieve comparable ERR($\widetilde{\pi}$) performance for all fields with 2 and 3 robots except for field 3 (i.e., of large ℓ_1 and small ℓ_2) with 2 robots. These observations reveal that (a) small horizontal and large vertical correlations are favorable to $\widetilde{\pi}$; (b) though large horizontal and small vertical correlations are not favorable to $\widetilde{\pi}$, this problem can be mitigated by increasing the number of robots. For more detailed analysis (e.g., visualization of planned observation paths and their corresponding error maps), the interested reader is referred to [4].

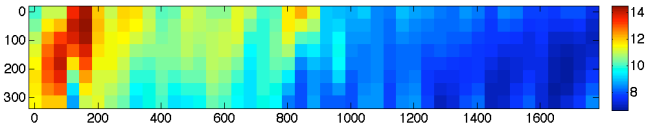


Figure 3: Plankton density (chl-a) field (measured in $mg\ m^{-3}$) spatially distributed over a 314 m $\times$ 1765 m transect that is discretized into a 8×45 grid with $\ell_1 = 27.53$ m and $\ell_2 = 134.64$ m.

We will now examine how the increase in resolution to 13×75 grid affects the ENT(π) and ERR(π) performance of evaluated policies; the resulting grid discretization width and planning horizon are about $0.4\times$ smaller and $2.5\times$ longer, respectively. Table 2 shows the results of mean ENT(π) and ERR(π) performance of tested policies with varying number of robots, from which we can derive observations similar to that for temperature field 4 discretized into 5×30 grid: $\widetilde{\pi}$ can achieve ENT($\widetilde{\pi}$) and ERR($\widetilde{\pi}$) performance comparable to (if not, better than) that of π^G and π^M except for ERR($\widetilde{\pi}$) performance with 1 robot. So, increasing the grid resolution does not seem to noticeably degrade the active sampling performance of $\widetilde{\pi}$ relative to that of π^G and π^M.

5.3 Plankton Density Field Data

Fig. 3 illustrates the plankton density field that is discretized into a 8×45 grid. Table 3 shows the results of mean ENT(π) and ERR(π) performance of tested policies with varying number of robots. The observations are as follows: $\widetilde{\pi}$ can achieve the same ENT($\widetilde{\pi}$) and ERR($\widetilde{\pi}$) performance as that of π^G and superior ENT($\widetilde{\pi}$) performance over that of π^M because small horizontal and large vertical correlations favor $\widetilde{\pi}$ as explained in Section 5.2. By increasing the number of robots (i.e., $k > 2$), $\widetilde{\pi}$ can achieve ERR($\widetilde{\pi}$) performance comparable to (if not, better than) that of π^M.

Table 4 shows the results of mean ENT(π) and ERR(π) performance of tested policies after increasing the resolution to 16×89 grid; the resulting grid discretization width and planning horizon are about $0.5\times$ smaller and $2\times$ longer, respectively. Similar observations can be obtained: $\widetilde{\pi}$ can achieve ENT($\widetilde{\pi}$) performance comparable to that of π^G and superior ENT($\widetilde{\pi}$) performance over that of π^M. By deploying more than 1 robot, $\widetilde{\pi}$ can achieve ERR($\widetilde{\pi}$) performance comparable to (if not, better than) that of π^G and π^M. Again, we can observe that increasing the grid resolution does not seem to noticeably degrade the active sampling performance of $\widetilde{\pi}$ relative to that of π^G and π^M.

5.4 Incurred Policy Time

Fig. 4 shows the time taken to derive the tested policies for sampling the temperature and plankton density fields with varying number of robots and grid resolutions. It can be observed that the time taken to derive $\widetilde{\pi}$ is shorter than that needed to derive π^G and π^M by more than 1 and 4 orders of magnitude, respectively. It is important to point out that Fig. 4 reports the average time taken to derive π^G and π^M over all possible starting robot locations. So, if the starting robot locations are unknown, the incurred time to derive π^G and π^M have to be increased by ${}^r\mathrm{C}_k$-fold. In contrast, $\widetilde{\pi}$ caters to all possible starting robot locations. So, the incurred time to derive $\widetilde{\pi}$ is unaffected. These observations show a considerable computational gain of $\widetilde{\pi}$ over π^G and π^M, which supports our time complexity analysis and comparison (Section 4). So, our Markov-based path planner

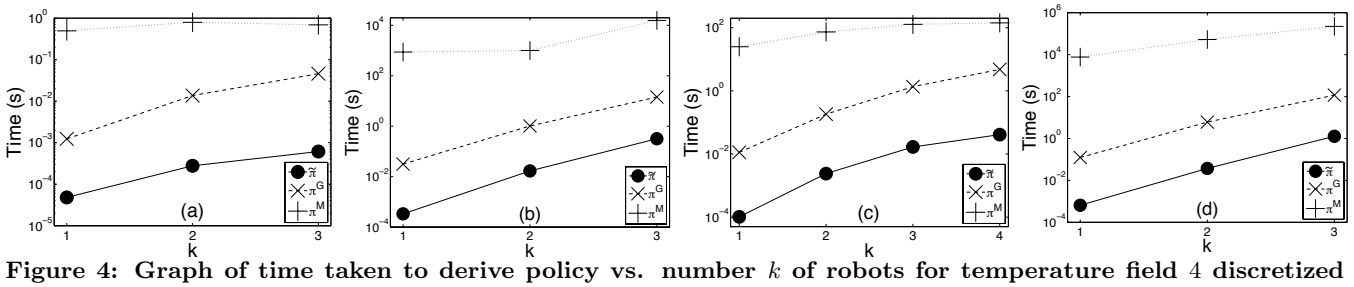


Figure 4: Graph of time taken to derive policy vs. number k of robots for temperature field 4 discretized into (a) 5×30 and (b) 13×75 grids and plankton density field discretized into (c) 8×45 and (d) 16×89 grids.

Table 3: Comparison of ENT(π) (left) and ERR(π) ($\times 10^{-3}$) (right) performance for plankton density field that is discretized into 8×45 grid.

	ENT(π)				ERR(π)			
	Number k of robots				Number k of robots			
Policy	1	2	3	4	1	2	3	4
$\widetilde{\pi}$	-359	-322	-196	-121	5.6124	2.2164	0.0544	0.0066
π^G	-359	-322	-196	-121	5.6124	2.2164	0.0544	0.0066
π^M	-230	-186	-70	-11	4.5371	0.5613	0.0472	0.0324

Table 4: Comparison of ENT(π) (left) and ERR(π) ($\times 10^{-3}$) (right) performance for plankton density field that is discretized into 16×89 grid.

ENT(π)	Number k of robots			ERR(π)	Number k of robots		
Policy	1	2	3	Policy	1	2	3
$\widetilde{\pi}$	-4278	-3949	-3681	$\widetilde{\pi}$	3.4328	0.0970	0.0546
π^G	-4238	-3964	-3686	π^G	1.5648	0.1073	0.0643
π^M	-4171	-3840	-3501	π^M	0.8186	0.0859	0.0348

is more time-efficient for *in situ*, real-time, high-resolution active sampling.

6. CONCLUSION

This paper describes an efficient Markov-based information-theoretic path planner for active sampling of GP-based environmental fields. We have provided theoretical guarantees on the active sampling performance of our Markov-based policy $\widetilde{\pi}$ for the transect sampling task, from which ideal environmental field conditions (i.e., small horizontal spatial correlation and noisy, less intense fields) and sampling task settings (i.e., large grid discretization width and short planning horizon) can be established to limit its performance degradation. Empirically, we have shown that $\widetilde{\pi}$ can generally achieve active sampling performance comparable to that of the widely-used non-Markovian greedy policies π^G and π^M under less favorable realistic field conditions (i.e., low noise-to-signal ratio) and task settings (i.e., small grid discretization width and long planning horizon) while enjoying huge computational gain over them. In particular, we have empirically observed that (a) small horizontal and large vertical correlations strongly favor $\widetilde{\pi}$; (b) though large horizontal and small vertical correlations do not favor $\widetilde{\pi}$, this problem can be mitigated by increasing the number of robots. In fact, deploying a large robot team often produces superior active sampling performance of $\widetilde{\pi}$ over π^M in our experiments, not forgetting the computational gain of > 4 orders of magnitude. Our Markov-based planner can be used to efficiently achieve more general exploration tasks (e.g., boundary tracking and those in [5, 6]), but the guarantees provided here may not apply. For our future work, we will "relax" the Markov assumption by utilizing a longer (but not entire) history of observations in path planning. This can potentially improve the active sampling performance in fields of moderate to large horizontal correlation but does not incur as much time as that of non-Markovian policies.

7. REFERENCES

[1] R. Korf. Real-time heuristic search. *Artif. Intell.*, 42(2-3):189–211, 1990.

[2] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

[3] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis. Collective motion, sensor networks and ocean sampling. *Proc. IEEE*, 95(1):48–74, 2007.

[4] K. H. Low. *Multi-Robot Adaptive Exploration and Mapping for Environmental Sensing Applications.* Ph.D. Thesis, Technical Report CMU-ECE-2009-024, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 2009.

[5] K. H. Low, J. M. Dolan, and P. Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proc. AAMAS*, pages 23–30, 2008.

[6] K. H. Low, J. M. Dolan, and P. Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, pages 233–240, 2009.

[7] K. H. Low, J. M. Dolan, and P. Khosla. Active Markov information-theoretic path planning for robotic environmental sensing. arXiv:1101.5632, 2011.

[8] K. H. Low, G. J. Gordon, J. M. Dolan, and P. Khosla. Adaptive sampling for multi-robot wide-area exploration. In *Proc. ICRA*, pages 755–760, 2007.

[9] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA, 2006.

[10] D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. Fratantoni, and M. J. Perry. Underwater gliders for ocean research. *Mar. Technol. Soc. J.*, 38(2):73–84, 2004.

[11] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *J. Applied Stat.*, 14(2):165–170, 1987.

[12] A. Ståhl, Ringvall, and T. Lämås. Guided transect sampling for assessing sparse populations. *Forest Science*, 46(1):108–115, 2000.

[13] D. R. Thompson and D. Wettergreen. Intelligent maps for autonomous kilometer-scale science survey. In *Proc. i-SAIRAS*, 2008.

[14] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, Cambridge, MA, 2005.

[15] R. Webster and M. Oliver. *Geostatistics for Environmental Scientists.* John Wiley & Sons, Inc., NY, 2nd edition, 2007.

Horde: A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction

Richard S. Sutton, Joseph Modayil, Michael Delp
Thomas Degris, Patrick M. Pilarski, Adam White
Reinforcement Learning and Artificial Intelligence Laboratory
Department of Computing Science, University of Alberta, Canada

Doina Precup
School of Computer Science, McGill University, Montreal, Canada

ABSTRACT

Maintaining accurate world knowledge in a complex and changing environment is a perennial problem for robots and other artificial intelligence systems. Our architecture for addressing this problem, called *Horde*, consists of a large number of independent reinforcement learning sub-agents, or *demons*. Each demon is responsible for answering a single predictive or goal-oriented question about the world, thereby contributing in a factored, modular way to the system's overall knowledge. The questions are in the form of a value function, but each demon has its own policy, reward function, termination function, and terminal-reward function unrelated to those of the base problem. Learning proceeds in parallel by all demons simultaneously so as to extract the maximal training information from whatever actions are taken by the system as a whole. Gradient-based temporal-difference learning methods are used to learn efficiently and reliably with function approximation in this off-policy setting. Horde runs in constant time and memory per time step, and is thus suitable for learning online in real-time applications such as robotics. We present results using Horde on a multi-sensored mobile robot to successfully learn goal-oriented behaviors and long-term predictions from off-policy experience. Horde is a significant incremental step towards a real-time architecture for efficient learning of general knowledge from unsupervised sensorimotor interaction.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Experimentation

Keywords

artificial intelligence, knowledge representation, robotics, reinforcement learning, off-policy learning, real-time, temporal-difference learning, value function approximation

Cite as: Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction, Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 761-768.

1. THE PROBLEM OF EXPRESSIVE AND LEARNABLE KNOWLEDGE

How to learn, represent, and use knowledge of the world in a general sense remains a key open problem in artificial intelligence (AI). There are high-level representation languages based on first-order predicate logic and Bayes networks that are very expressive, but in these languages knowledge is difficult to learn and computationally expensive to use. There are also low-level languages such as differential equations and state-transition matrices that can be learned from data without supervision, but these are much less expressive. Knowledge that is even slightly forward looking, such as 'If I keep moving, I will bump into something within a few seconds' cannot be expressed directly with differential equations and may be expensive to compute from them. There remains room for exploring alternate formats for knowledge that are expressive yet learnable from unsupervised sensorimotor data.

In this paper we pursue a novel approach to knowledge representation based on the notion of value functions and on other ideas and algorithms from reinforcement learning. In our approach, knowledge is represented as a large number of approximate value functions learned in parallel, each with its own policy, pseudo-reward function, pseudo-termination function, and pseudo-terminal-reward function. Learning systems using multiple approximate value functions of this type have previously been explored as temporal-difference networks with options (Sutton, Rafols & Koop 2006; Sutton, Precup & Singh 1999). Our architecture, called *Horde*, differs from temporal-difference networks in its more straightforward handling of state and function approximation (no predictive state representations) and in its use of more efficient algorithms for off-policy learning (Maei & Sutton 2010; Sutton et al. 2009). The current paper also extends prior work in that we demonstrate real-time learning on a physical robot.

Previous work on the problem of representing a general sense of knowledge while being grounded in and learnable from sensorimotor data goes back at least to Cunningham (1972) and Becker (1973). Drescher (1991) considered a simulated robot baby learning conditional probability tables for boolean events. Ring (1997) explored continual learning of a hierarchical representation of sequences. Cohen et al. (1997) explored the formation of symbolic fluents from simulated experience. Kaelbling et al. (2001) and Pasula et al. (2007)

explored the learning of relational rule representations in stochastic domains. All these systems involved learning significant knowledge but remained far from learning from sensorimotor data. Previous researchers who did learn from sensorimotor data include Pierce and Kuipers (1997), who learned spatial models and control laws, Oates et al. (2000), who learned clusters of robot trajectories, Yu and Ballard (2004), who learned word meanings, and Natale (2005), who learned goal-directed physical actions. All of these works learned significant knowledge but specialized on knowledge of a particular kind; the knowledge representation they used is not as general as that of multiple approximate value functions.

2. VALUE FUNCTIONS AS SEMANTICS

A distinctive, appealing feature of approximate value functions as a knowledge representation language is that they have an explicit semantics, a clear notion of truth grounded in sensorimotor interaction. A bit of knowledge expressed as an approximate value function is said to be true, or more precisely, *accurate*, to the extent that its numerical values match those of the mathematically defined value function that it is approximating. A value function asks a *question*—what will the cumulative future reward be?—and an approximate value function provides an *answer* to that question. The approximate value function is the knowledge, and its match to the value function—to the actual future reward—defines what it means for the knowledge to be accurate. The idea of the present work is that the value-function approach to grounding semantics can be extended beyond reward to a theory of all world knowledge. In this section we define these ideas formally for the case of reward and conventional value functions (and thereby introduce our notation), and in the next section we extend them to knowledge and general value functions.

In the standard reinforcement learning framework (Sutton & Barto 1998), the interaction between the AI agent and its world is divided into a sequence of discrete time steps, $t = 1, 2, 3, \ldots$, each corresponding perhaps to a fraction of a second. The state of the world at each step, denoted $S_t \in \mathcal{S}$, is sensed by the agent, perhaps incompletely, and used to select an action $A_t \in \mathcal{A}$ in response. One time step later the agent receives a real-valued reward $R_{t+1} \in \mathbb{R}$ and a next state $S_{t+1} \in \mathcal{S}$, and the cycle repeats. Without loss of significant generality, we can consider the rewards to be generated according to a deterministic *reward function* $r : \mathcal{S} \to \mathbb{R}$, with $R_t = r(S_t)$.

The focus in conventional reinforcement learning is on learning a stochastic action-selection *policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that gives the probability of selecting each action in each state, $\pi(s, a) = \mathbb{P}(A_t = a | S_t = s)$. Informally, a good policy is one that results in the agent receiving a lot of reward summed over time steps. For example, in game playing the reward might correspond to points won or lost on each turn, and in a race the reward might be -1 on each time step. In *episodic* problems, the agent–world interaction consists of multiple finite trajectories (episodes) that can terminate in better or worse ways. For example, playing a game may generate a sequence of moves that eventually ends with a win, loss, or draw, with each outcome having a different numerical value, perhaps $+1$, -1 and 0. A race may be completed successfully or end in disqualification, two very different outcomes even if the number of seconds elapsed is the same. Another example is optimal control, in which it is common to have costs for each step (e.g., related to energy expenditure) plus a terminal cost (e.g., relating to how far the final state is from a goal state). In general, a problem may have both a reward function as already formulated and also a *terminal-reward function*, $z : \mathcal{S} \to \mathbb{R}$, where $z(s)$ is the terminal reward received if termination occurs upon arrival in state s.

We turn now to formalizing the process of termination. In many reinforcement learning problems, particularly non-episodic ones, it is common to give less weight to delayed rewards, in particular, to *discount* them by a factor of $\gamma \in [0, 1)$ for each step of delay. One way to think about discounting is as a constant probability of termination, of $1-\gamma$, together with a terminal reward that is always zero. More generally, we can consider there to be an arbitrary *termination function*, $\gamma : \mathcal{S} \to [0, 1]$, with $1 - \gamma(s)$ representing the probability of terminating upon arrival in state s, at which time a corresponding terminal reward of $z(s)$ would be registered. The overall *return*, a random variable denoted G_t for the trajectory starting at time t, is then the sum of the per-step rewards received up until termination occurs, say at time T, *plus* the final terminal reward received in S_T:

$$G_t \;=\; \sum_{k=t+1}^{T} r(S_k) \;+\; z(S_T). \tag{1}$$

The conventional *action-value function* $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is then defined as the expected return for a trajectory starting from the given state and action and selecting actions according to policy π until terminating according to γ (thus determining the time of termination, T):

$$Q^\pi(s, a) \;=\; \mathbb{E}[G_t \mid S_t{=}s, A_t{=}a, A_{t+1:T-1} \sim \pi, T \sim \gamma]\,.$$

This expectation is well defined given a particular state-transition structure for the world (say as a Markov decision process). If an AI agent were to possess an approximate value function, $\hat{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, then it could be assessed for accuracy according to its closeness to Q^π, for example, according to the expectation of its squared error, $(Q^\pi(s, a) - \hat{Q}(s, a))^2$, over some distribution of state–action pairs. In practice it is rarely possible to measure this error exactly, but the value function Q^π still provides a useful theoretical semantics and ground truth for the knowledge $\hat{Q}$. The value function is the exact numerical answer to the precise, grounded question 'What would the return be from each state–action pair if policy π were followed?', and the approximate value function offers an approximate numerical answer. In this precise sense the value function provides a semantics for the knowledge represented by the AI agent's approximate value function.

Finally, we note that the value function for a policy is often estimated solely for the purpose of improving the policy. Given a policy π and its value function Q^π, we can construct a new deterministic *greedy* policy $\pi' = greedy(Q^\pi)$ such that $\pi'(s, \arg\max_a Q^\pi(s, a)) = 1$, and the new policy is guaranteed to be an improvement in the sense that $Q^{\pi'}(s, a) \geq Q^\pi(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, with equality only if both policies are optimal. Through successive steps of estimation and improvement, a policy that optimizes the expected return can be found. In this way the theory of value functions provides a semantics for goal-oriented knowledge (control) as well as for predictive knowledge.

3. FROM VALUES TO KNOWLEDGE (GENERAL VALUE FUNCTIONS)

Having made clear how a conventional value function provides a grounded semantics for knowledge about upcoming reward, in this section we show how *general value functions (GVFs)* provide a grounded semantics for a more general kind of world knowledge. Using the ideas and notation developed in the previous section, this is almost immediate.

First note that although the action-value function Q^π is conventionally superscripted only by the policy, it is equally dependent on the reward and terminal-reward functions, r and z. These functions could equally well have been considered inputs to the value function in the same way that π is. That is, we might have defined a more general value function, which might be denoted $Q^{\pi,r,z}$, that would use returns (1) defined with arbitrary functions r and z acting as *pseudo*-reward function and *pseudo*-terminal-reward function. For example, suppose we are playing a game, for which the base terminal rewards are $z=+1$ for winning and $z=-1$ for losing (with a per-step reward of $r=0$). In addition to this, we might pose an independent question about how many more moves the game will last. This could be posed as a general value function with pseudo-reward function $r=1$ and pseudo-terminal-reward function $z=0$. Later in this paper we consider several more examples from a robot domain.

The second step from value functions to GVFs is to convert the termination function γ to a pseudo form as well. This is slightly more substantive because, unlike the rewards and terminal rewards, which do not pertain to the state evolution in any way, termination conventionally refers to an interruption in the normal flow of state transitions and a reset to a starting state or starting-state distribution. For pseudo termination we simply omit this additional implication of conventional termination. The real, base problem may still have real terminations or it may have no terminations at all. Yet we may consider pseudo terminations to have occurred at any time. For example, in a race, we can consider a pseudo-termination function that terminates at the half way point. This is a perfectly well defined problem with a value function in the general sense. Or, if we are the racer's spouse, then we may not care about when the race ends but rather about when the racer comes home for dinner, and that may be our pseudo termination. For the same world—the same actions and state transitions—there are many predictive questions that can be defined in the form of general value functions.

Formally, we define a *general value function*, or GVF, as a function $q : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ with four auxiliary functional inputs π,γ,r, and z, defined over the same domains and ranges as specified earlier, but now taken to be arbitrary and with no necessary relationship to the base problem's reward, terminal-reward, and termination functions:

$$q(s,a;\pi,\gamma,r,z)=\mathbb{E}[G_t \mid S_t{=}s, A_t{=}a, A_{t+1:T-1}{\sim}\pi, T{\sim}\gamma],$$

where G_t is still defined by (1) but now with respect to the given functions. The four functions, π,γ,r, and z, are referred to collectively as the GVF's *question functions*; they define the question or semantics of the GVF. Note that conventional value functions remain a special case of GVFs. Thus, we can consider all value functions to be GVFs. In the rest of the paper, for simplicity, we sometimes use the expression "value function" to mean the general case, using "conventional value function" when needed to disambiguate. We also drop the 'pseudo-' prefix from the question functions when it can be done without ambiguity. In the robot experiments that we present later there are no privileged base problems, so there should be no confusion.

4. THE HORDE ARCHITECTURE

The Horde architecture consists of an overall agent composed of many sub-agents, called *demons*. Each demon is a independent reinforcement-learning agent responsible for learning one small piece of knowledge about the base agent's interaction with its environment. Each demon learns an approximation, $\hat{q}$, to the GVF, q, that corresponds to the demon's setting of the four question functions, π,γ,r, and z.

We turn now to describing Horde's mechanisms for approximating GVFs with a finite number of weights, and for learning those weights. In this paper we adopt the standard linear approach to function approximation. We assume that the world's state and action at each time step, S_t and A_t, are translated, presumably incompletely via sensory readings, into a fixed-size *feature vector* $\phi_t=\phi(S_t,A_t)\in\mathbb{R}^n$ where $n\ll|\mathcal{S}|$. We refer to the set of all features, for all state–action pairs, as Φ. In our experiments, the feature vector is constructed via tile coding and thus is binary, $\phi_t\in\{0,1\}^n$, with a constant number of 1 features (see Sutton & Barto 1998). We also focus on the case where $|\mathcal{S}|$ is large, possibly infinite, but $|\mathcal{A}|$ is finite and relatively small, as is common in reinforcement learning problems. These are convenient special cases, but none of them is essential to our approach. Our approximate GVFs, denoted $\hat{q}:\mathcal{S}\times\mathcal{A}\times\mathbb{R}^n\to\mathbb{R}$, are linear in the feature vector:

$$\hat{q}(s,a,\theta)=\theta^\top\phi(s,a),$$

where $\theta\in\mathbb{R}^n$ is the vector of weights to be learned, and $v^\top w=\sum_i v_i w_i$ denotes the inner product of two vectors v and w.

For learning the weights we use recently developed gradient-descent temporal-difference algorithms (Sutton et al. 2009, 2008; Maei et al. 2009, 2010). These algorithms are unique in their ability to learn stably and efficiently with function approximation from off-policy experience. Off-policy experience means experience generated by a policy, called the *behavior policy*, that is different from that being learned about, called the *target policy*. To learn knowledge efficiently from unsupervised interaction one seems inherently to face such a situation because one wants to learn in parallel about many policies—the different target policies π of each GVF—but of course one can only be behaving according to one policy at a time.

For a typical GVF, the actions taken by the behavior policy will match its target policy only on occasion, and rarely for more than a few steps in a row. For efficient learning, we need to be able to learn from these snippets of relevant experience, and this requires off-policy learning. The alternative—on-policy learning—would require learning only from snippets that are complete in that the actions match those of the GVF's target policy all the way to pseudo-termination, a much less common occurrence. If learning can be done off-policy from incomplete snippets of experience then it can be massively parallel and potentially much faster than on-policy learning.

Only in the last few years have off-policy learning algorithms become available that work reliably with function ap-

proximation and that scale appropriately for real-time learning and prediction (Sutton et al. 2008, 2009). Specifically, in this work we use the GQ(λ) algorithm (Maei & Sutton 2010). This algorithm maintains, for each GVF, a second set of weights $w \in \mathbb{R}^n$ in addition to θ and an eligibility-trace vector $e \in \mathbb{R}^n$. All three vectors are initialized to zero. Then, on each step, GQ(λ) computes two temporary quantities, $\bar{\phi}_t \in \mathbb{R}^n$ and $\delta_t \in \mathbb{R}$:

$$\bar{\phi}_t = \sum_a \pi(S_{t+1}, a)\phi(S_{t+1}, a),$$

$$\delta_t = r(S_{t+1}) + (1-\gamma(S_{t+1}))z(S_{t+1}) + \gamma(S_{t+1})\theta^\top \bar{\phi}_t - \theta^\top \phi(S_t, A_t),$$

and updates the three vectors:

$$\theta_{t+1} = \theta_t + \alpha_\theta \left(\delta_t e_t - \gamma(S_{t+1})(1-\lambda(S_{t+1}))(w_t^\top e_t)\bar{\phi}_t\right),$$

$$w_{t+1} = w_t + \alpha_w \left(\delta_t e_t - (w_t^\top \phi(S_t, A_t))\phi(S_t, A_t)\right),$$

$$e_t = \phi(S_t, A_t) + \gamma(S_t)\lambda(S_t)\frac{\pi(S_t, A_t)}{b(S_t, A_t)} e_{t-1},$$

where $b : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the behavior policy and $\lambda : S \to [0, 1]$ in an *eligibility-trace function* which determines the rate of decay of the eligibility traces as in the TD(λ) algorithm (Sutton 1988). Note that the per-time-step computation of this algorithm scales linearly with the number of features, n. Moreover, if the features are binary, then with a little care the per-time-step complexity can be kept a small multiple of the number of 1 features.

The approximation that will be found asymptotically by the GQ(λ) algorithm depends on the feature vectors Φ, the behavior policy b, and the eligibility-trace function λ. These three are collectively referred to as the *answer functions*. In this paper's experiments we always used constant λ, and all demons shared the same Φ and b. Finally, we note that Maei and Sutton defined a termination function, β, that is of the opposite sense as our γ; that is, $\beta(s) = 1 - \gamma(s)$. This is purely a notational difference and does not affect the algorithm in any way.

We can think of the demons as being of two kinds. A demon with a given target policy, π, is called a *prediction demon*, whereas a demon whose target policy is the greedy policy with respect to its own approximate GVF (i.e., $\pi = greedy(\hat{q})$, or $\pi(s, \arg\max_a \hat{q}(s, a, \theta)) = 1$) is called a *control demon*. Control demons can learn and represent how to achieve goals, whereas the knowledge in prediction demons is better thought of as declarative facts. One way in which the demons are not completely independent is that a prediction demon can reference the target policy of a control demon. For example, in this way one could ask questions such as 'If I follow this wall as long as I can, will my light sensor then have a high reading?'. Demons can also use each others' answers in their questions (as in temporal-difference networks). This allows one demon to learn a concept such as 'near an obstacle,' say as the probability of a high bump-sensor reading within a few seconds of random actions, and then a second demon to learn something based on this, such as 'If I follow this wall to its end, will I then be *near an obstacle*?' by using the first demon's approximate GVF in its terminal-reward function (e.g., $z(s) = \max_a \hat{q}(s, a, \theta_{\text{first\_demon}})$).

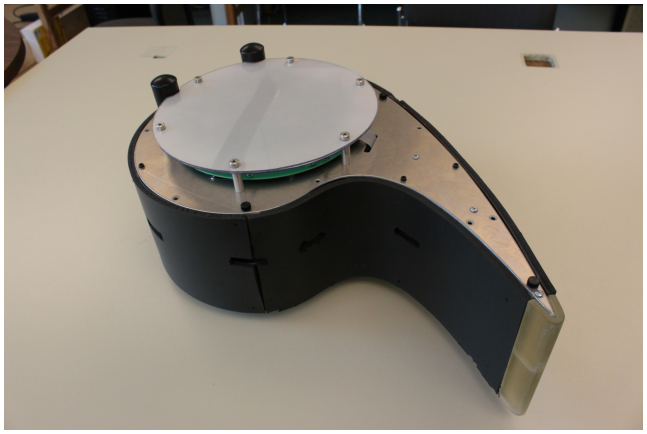


Figure 1. The Critterbot robotic platform.

5. RESULTS WITH HORDE ON THE CRITTERBOT

To evaluate the effectiveness of the Horde architecture, we deployed it on the *Critterbot*, a custom-built mobile robot (Figure 1). The Critterbot has a comma-shaped frame with a 'tail' that facilitates object interaction and is driven by three omni-directional wheels separated by 120 degrees. A diverse set of sensors are deployed on the top of the robot, including sensors for ambient light, heat, infrared light, magnetic fields, and sound. Another batch of sensors captures proprioceptive information including battery voltages, acceleration, rotational velocity, motor velocities, motor currents, motor temperatures, and motor voltages. The robot can detect nearby obstacles with ten infrared proximity sensors distributed along its sides and tail. The robot has been designed to withstand the rigors of reinforcement learning experiments; it can drive into walls for hours without damage or burning out its motors, it can dock autonomously with its charging station, and it can run continuously for twelve hours without recharging.

The Critterbot's sensors provide useful information about its interaction with the world, but this information can be challenging to model explicitly. For example, the sensor readings from the magnetometer may be influenced by the operation of data servers in the next room, and the ambient light sensors are affected by natural daylight, indoor florescent lights, shadows from looming humans, and reflections from walls. Manually modeling these interactions is difficult and potentially futile. The Horde architecture presents an alternative wherein each demon autonomously learns a little bit about the relationships among the sensors and actuators from unsupervised experience.

We performed a series of experiments to examine how well the architecture supports learning. In each experiment, the observations and actions were tiled to form a state–action feature representation Φ. A discrete set of actions were selected, matching the formulation of the GQ(λ) algorithm. With these choices, the entire architecture operates in constant time per step. We have run the Horde architecture in real-time with thousands of demons using billions of binary features of which a few thousand were active at a time, using laptop computers.

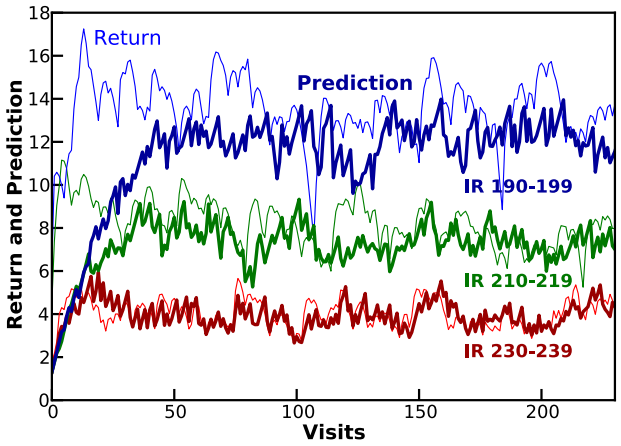


Figure 2. Accurately predicting time-to-obstacle. The robot was repeatedly driven toward a wall at a constant wheel speed. For each of three regions of the sensor space, for each time step spent in that region , we plot the demon prediction $\hat{q}$ on that step (bold line) and the actual return from that step (thin line).

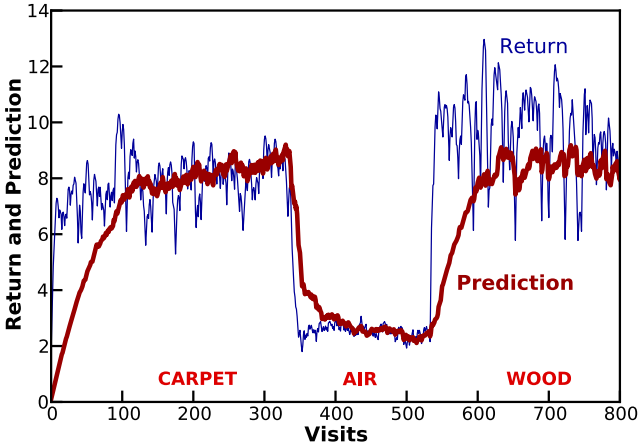


Figure 3. Accurately tracking time-to-stop. The robot was repeatedly rotated up to a standard wheel speed, then switched to a policy that always took the STOP action, on three different floor surfaces. Shown is the prediction $\hat{q}$ made on visits to a region of high velocity while stopping (bold line) together with the actual return from that visit (thin line). The floor surface was changed after visits 338 and 534.

5.1 Subjective prediction experiments

Our first two experiments dealt with Horde's ability to answer subjectively posed predictive questions. Figures 2 and 3 show results on the Critterbot with instances of the Horde architecture each with a single prediction demon. The specific questions posed are ones that might be useful in ensuring safety: 'How much time do I have before hitting an obstacle?' and 'How much time do I need to stop?'. In both cases accurate predictions were made, and in the latter case they were adapted so as to remain accurate as the experiment was changed from stopping on carpet, to stopping when suspended in the air, to stopping on a wood floor. The time step used in these experiment was approximately 30ms in length.

Figure 2 shows a comparison between predicted and observed time steps needed to reach obstacles when driving forward. Shown are the demon predictions $\hat{q}$ on each step (bold line) for each time step spent in a region of the sensor space (a visit), and the actual return from that step (thin line). The prediction was learned from a behaviour policy that cycled between three actions: driving forward, reverse, and resting. This is plotted for each of three regions of the sensor space: IR=190–199, IR=210–219, and IR=230–239. These represent three different value ranges of the Critterbot's front IR proximity sensor.

The question functions for this demon were: $\pi(s, \text{FORWARD}) = 1, r(s) = 1, z(s) = 0$, $\forall s \in \mathcal{S}$, and $\gamma(s) = 0$ if the value of the Critterbot's front-pointing IR proximity sensor was over a fixed threshold, else $\gamma(s) = 1$. The remaining answer functions were $\lambda(s) = 0.4, \forall s \in \mathcal{S}$, and Φ = a single tiling into twenty-six regions of the front IR sensor. The GQ(λ) step sizes were $\alpha_\theta = 0.3$ and $\alpha_w = 0.00001$. As shown in Figure 2, this demon learned to accurately predict the return (time steps to impact) for each range of its sensors.

Figure 3 demonstrates a demon's ability to accurately predict stopping times on different surfaces. Shown is the prediction $\hat{q}$ made on visits to a region of high velocity while stopping (bold line) together with the actual return from that visit (thin line). For this predictive question, we defined a single demon that predicts the number of timesteps until one of the robot's wheels approaches zero velocity (i.e., comes to a complete stop) under current environmental conditions. The robot's behaviour policy was to alternate at fixed intervals between spinning at full speed and resting. The floor surface, and thus the nature of the stopping problem, was changed after visits 338 and 534.

The question functions for this demon were: $\pi(s, \text{STOP}) = 1, r(s) = 1, z(s) = 0$, $\forall s \in \mathcal{S}$, and $\gamma(s) = 0$ if the wheel's velocity sensor was below a fixed threshold, else $\gamma(s) = 1$. The remaining answer functions were $\lambda(s) = 0.1, \forall s \in \mathcal{S}$, and Φ = a single tiling into eight regions of the wheel's velocity sensor. The GQ(λ) step sizes were $\alpha_\theta = 0.1$ and $\alpha_w = 0.001$. As illustrated in Figure 3, this demon learned to correctly predict the return (time steps to stopping) on carpet, then adapted its prediction when the environment changed to air and then to wood flooring.

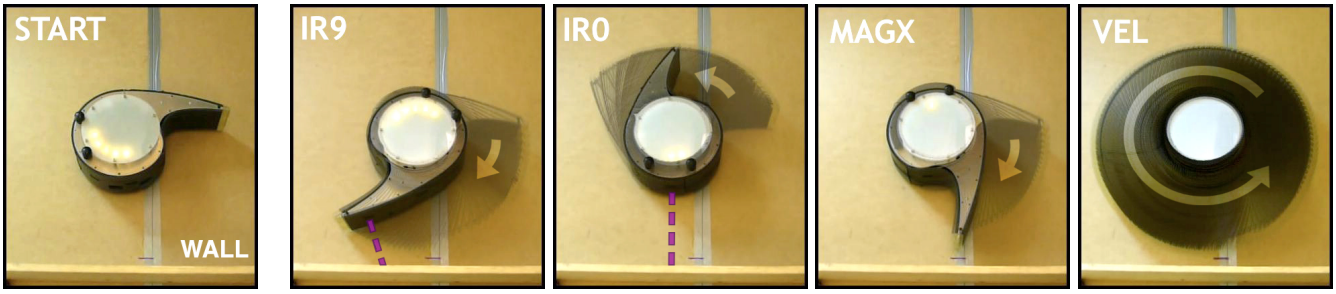


Figure 4. Illustration of policies learned by four control demons in the spinning experiment. The first panel shows the standard starting position, and the other four panels show the motions from that position produced when control was given to one of the eight learned demon policies each tasked to maximize a different sensor. By maximized sensor: IR9) Robot quickly rotates clockwise and stops in the position that maximizes the IR proximity sensor on the side of the robot's tail; IRO) Robot quickly rotates counterclockwise, overshoots a bit, then settles in a position that maximizes the proximity sensor between the robot's 'eyes'; MAGX) Robot rotates clockwise and stops at a position that maximizes the magnetic x-axis sensor; VEL) Robot spins continuously, maximizing the wheel velocity sensor.

5.2 Off-policy learning of multiple spinning control policies

Our third experiment examined whether control demons can learn policies in parallel while following a random behavior policy, in other words, whether the demons can learn off-policy, a crucial ability for the scalability of the architecture. The action set in this experiment was {ROTATE-RIGHT, ROTATE-LEFT, STOP}. The behavior policy was to randomly select one of the three actions, with a bias (50% probability) toward repeating the action taken on the previous time step. The result of this behavior policy was that the robot would spin in place in both directions with a variety of speeds and durations over time. The state space was represented with four overlapping joint tilings across three sensors: the magnetometer, one of the IR sensors, and the velocity of one of the wheels. Each sensor was divided into eight regions for the tilings, resulting in a total of $3 \times 4 \times 8^3 = 6144$ binary features. One additional feature was provided as a bias unit (always =1), and three additional binary features were used to encode the previous action. The time step corresponded to approximately 100ms. The other parameters were $\alpha_\theta = 0.1$, $\alpha_w = 0.001$, and $\lambda(s) = 0.4, \forall s \in \mathcal{S}$. Learning was done online, but the data was also saved so that the whole learning process could be repeated without using the robot if desired (this is one of the advantages of an off-policy learning ability).

In this experiment we ran eight control demons in parallel for 100,000 time steps of off-policy learning with actions selected according to the behavior policy. Each demon was tasked with learning how to maximize a different sensor value. That is, their question functions were $\pi = greedy(\hat{q})$ and, for all $s \in \mathcal{S}$, $\gamma(s) = 0.98$, $z(s) = 0$, and $r(s) =$ the value of one of eight sensors approximately normalized to a 0 to 1 range. The eight sensors used as rewards were four of the IR proximity sensors, the magnetometer, the velocity sensor for one of the wheels, one of the thermal sensors, and an IR beacon sensor for the charging station. To objectively measure the quality of the policies learned by the eight demons, we occasionally interrupted learning to evaluate them on-policy. That is, with learning turned off, the robot followed one of the eight learned demon policies for 250 time steps and we measured the demon's return. We repeated this for each demon ten times from each of three initial starting positions (angles) to produce 30 measures of the effectiveness of each demon's policy at that point in the training. These numbers were averaged together to produce the learning curves shown in Figure 5.

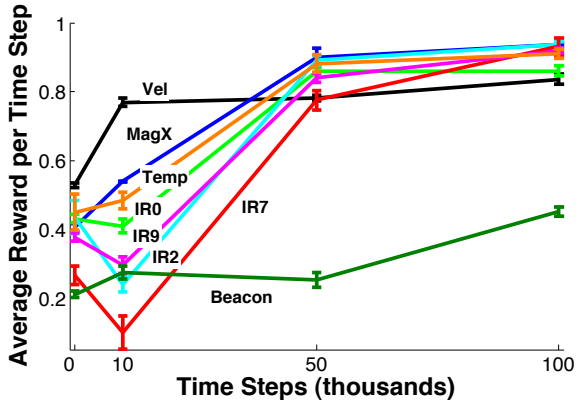


Figure 5. Learning curves for eight control demons learning off-policy in the spinning experiment. From extensive experience spinning, eight control demons learned different policies each maximizing a different sensor. The graph shows the performance of the policies, gathered in special on-policy evaluation sessions during which learning was turned off. All demons learned to perform near optimally. Rewards were scaled to the range $[0, 1]$, but because the beacon light flashes on and off, its maximal average was 0.5.

Examples of the final learned behavior from four of the demons are shown in Figure 4. These photos show typical behavior, which in the case of all eight demons appeared to successfully maximize the targeted sensor. In separate runs we found that it would take approximately 25,000 steps each to learn similarly competent control policies for a single demon while behaving according to its policy as it was learned (on-policy training). In only four times longer, we learned eight demons in parallel, and could potentially have learned thousands or millions more using off-policy learning.

Figure 6. Learning light-seeking behavior from random behavior. Shown are superimposed images of robot positions: Left) In testing, the robot under control of the demon policy turns and drives straight to the light source at the bottom of the image; Middle) Under control of the random behavior policy for the same amount of time, the robot instead wanders all over the pen; Right) Light sensor readings averaged over seven such pairs of runs, showing much higher values for the learned target policy.

5.3 Off-policy learning of light-seeking

A final experiment examined whether a control demon could learn a goal-directed policy when given a much greater breadth of experience. In particular, we chose question functions corresponding to the goal of maximizing the near-term value of one of the light sensors: $\pi = greedy(\hat{q})$, $\gamma(s) = 0.9$, $z(s) = 0$, $r(s) =$ a scaled reading from the front light sensor. The behavior policy was to pick randomly from the set $\{+10, -10, 0\}^3$ interpreted as velocities for the robot's three wheels, for a total of 27 possible actions. The state space was represented with 32 individual tilings over each of the four directional light sensors, where each tile covered about 1/8th of the range. With the addition of a bias unit, this made for a total of $27 \times (32 \times 4 \times 8 + 1) = 27,675$ binary features, of which $32 \times 4 + 1 = 129$ were active on each time step. The time step corresponded to approximately 500ms.

Using the random behavior policy, we collected a training set of 61,200 time steps (approximately 8.5 hours) with a bright light at nearly floor level on one side of the pen. During this time the robot wandered all over the pen in many orientations. We trained the control demon off-line and off-policy in two passes over the training set. To assess what had been learned, we then placed the robot in the middle of the pen facing away from the light and gave control to the demon's learned policy. The robot would typically turn immediately and drive toward the light, as shown in the first panel of Figure 6. This result demonstrates that demons can learn effective goal-directed behavior from substantially different training behavior.

Together, our results show that the Horde architecture can be applied to robot systems to learn potentially useful bits of knowledge in real-time from unsupervised experience. The approach works across a range of feature representations, parameters, questions, and goals. The robot is able to learn bits of knowledge that could serve as useful components for solving more complex tasks.

6. CONCLUSION

The Horde architecture is an experiment in knowledge representation and learning built upon ideas and algorithms from reinforcement learning. The approach is to express knowledge in the form of generalized value functions (GVFs) and thereby ground its semantics in sensorimotor data. This approach is promising because 1) value functions make it possible to capture temporally extended predictive and goal-oriented knowledge, 2) a large amount of important knowledge is of this form, 3) conventional knowledge representations of the grounded type (such as differential equations) have difficulty representing knowledge of this form, and 4) conventional methods that can capture this kind of knowledge (high-level, symbolic methods such as rules, operators, and production systems) are not as grounded and therefore not as learnable as value functions. Although value functions have always been potentially learnable, only recently have scalable learning methods become available that make it practical to explore the idea of GVFs with off-policy learning and function approximation. This work presents a first look at the application and interpretation of GVFs in an architecture with parallel off-policy learners.

In this paper we have focused on representing and learning knowledge as GVFs, and as such we have made only suggestive comments about how such knowledge could be used. Although this is an important limitation of our work, we believe that it is an appropriate way to break down the problem. The issues in learning and representation with GVFs that we address here are non-trivial and have not been adequately addressed before—certainly not in an embodied, robotic form. In addition, reinforcement-learning ideas such as value functions are already closely connected to known action-selection and planning methods; it is not a great leap to imagine several ways in which GVFs could be used to generate and improve behavior. We have briefly demonstrated some of these, such as passing control to the learned policy of single demons (e.g., the sensor-maximization demons in Section 5.2 and the light-seeking demon in Section 5.3), and indicated how several demons could be combined to

modulate an existing policy (e.g., varying behavior based on impact and stopping time predictions as suggested by Section 5.1). A rich and varied collection of demons and questions, as made possible by the Horde architecture, allows for a broad set of fusions of this kind. We have not developed here the natural possibility of using GVFs to represent multi-scale policy-contingent models of the world's dynamics (option models; Sutton, Precup & Singh 1999), and then using the models for planning as in dynamic programming, Monte Carlo tree search (see Chaslot 2010), or Dyna architectures (Sutton 1990). This is another natural direction for future work.

7. ACKNOWLEDGMENTS

The authors are grateful to Anna Koop, Mark Ring, Hamid Maei, and Chris Rayner for insights into the ideas presented in this paper. We also thank Michael Sokolsky and Marc Bellemare for assistance with the design, creation, and maintenance of the Critterbot. This research was supported by iCORE and Alberta Ingenuity, both part of Alberta Innovates – Technology Futures, by the Natural Sciences and Engineering Research Council of Canada, and by MITACS.

8. REFERENCES

Becker, J. D. (1973). A model for the encoding of experiential information. In *Computer Models of Thought and Language*, Schank, R. C., Colby, K. M., Eds. W. H. Freeman and Company.

Chaslot, G. M. J-B. (2010). Monte-Carlo tree search. PhD thesis, Dutch Research School for Information and Knowledge Systems.

Cohen, P. R., Atkin, M. S., Oates, T., Beal, C. R. (1997). Neo: Learning conceptual knowledge by sensorimotor interaction with an environment. In *Agents '97*, Marina del Rey, CA. ACM.

Cunningham, M. (1972). *Intelligence: Its Organization and Development*. Academic Press.

Drescher, G. L. (1991). *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press, Cambridge, MA.

Kaelbling, L. P., Oates, T., Hernandez, N., Finney, S. (2001). Learning in worlds with objects. *Working Notes of the AAAI Stanford Spring Symposium on Learning Grounded Representations*.

Maei, H. R., Sutton, R. S. (2010). GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proceedings of the Third Conference on Artificial General Intelligence*, Lugano, Switzerland.

Maei, H. R., Szepesvári, Cs., Bhatnagar, S., Precup, D., Silver, D., Sutton, R. S. (2009). Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems 22*, Vancouver, BC. MIT Press.

Maei, H. R., Szepesvári, Cs., Bhatnagar, S., Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel.

Natale, L. (2005). Linking action to perception in a humanoid robot: A developmental approach to grasping. MIT PhD thesis.

Oates, T., Schmill, M. D., Cohen, P. R. (2000). A method for clustering the experiences of a mobile robot that accords with human judgments. *Proceedings AAAI*, 846–851, AAAI/MIT Press.

Pasula, H., Zettlemoyer, L., Kaelbling L. (2007). Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research 29*:309–352.

Pierce, D. M., Kuipers, B. J. (1997). Map learning with uninterpreted sensors and effectors. *Artificial Intelligence 92*:169–227.

Ring, M. B. (1997). CHILD: A first step toward continual learning. *Machine Learning 28*:77–104.

Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning 3*:9–44.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pp. 216–224. Morgan Kaufmann, San Mateo, CA.

Sutton, R. S., Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvari, Cs., Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada.

Sutton, R. S., Precup D., Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence 112*:181–211.

Sutton, R. S., Rafols, E. J., Koop, A. (2006). Temporal abstraction in temporal-difference networks. *Advances in Neural Information Processing Systems 18*.

Sutton, R. S., Szepesvári, Cs., Maei, H. R. (2008). A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in Neural Information Processing Systems 21*.

Yu, C., Ballard, D. (2004). A multimodal learning interface for grounding spoken language in sensory perceptions. *ACM Transactions on Applied Perception 1*:57–80.

On Optimizing Interdependent Skills: A Case Study in Simulated 3D Humanoid Robot Soccer

Daniel Urieli, Patrick MacAlpine, Shivaram Kalyanakrishnan,
Yinon Bentor, and Peter Stone
Department of Computer Science, The University of Texas at Austin
1616 Guadalupe St Suite 2.408 Austin Texas 78701 USA
{urieli, patmac, shivaram, yinon, pstone}@cs.utexas.edu

ABSTRACT

In several realistic domains an agent's behavior is composed of multiple *interdependent* skills. For example, consider a humanoid robot that must play soccer, as is the focus of this paper. In order to succeed, it is clear that the robot needs to walk quickly, turn sharply, and kick the ball far. However, these individual skills are ineffective if the robot falls down when switching from walking to turning, or if it cannot position itself behind the ball for a kick.

This paper presents a learning architecture for a humanoid robot soccer agent that has been fully deployed and tested within the RoboCup 3D simulation environment. First, we demonstrate that individual skills such as walking and turning can be parameterized and optimized to match the best performance statistics reported in the literature. These results are achieved through effective use of the CMA-ES optimization algorithm. Next, we describe a framework for optimizing skills *in conjunction* with one another, a little-understood problem with substantial practical significance. Over several phases of learning, a total of roughly 100–150 parameters are optimized. Detailed experiments show that an agent thus optimized performs comparably with the top teams from the RoboCup 2010 competitions, while taking relatively few man-hours for development.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

General Terms

Algorithms, Design, Experimentation.

Keywords

Humanoid robotics, Robot soccer, Skill learning, CMA-ES.

Cite as: On Optimizing Interdependent Skills: A Case Study in Simulated 3D Humanoid Robot Soccer, Daniel Urieli, Patrick MacAlpine, Shivaram Kalyanakrishnan, Yinon Bentor, and Peter Stone, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 769-776.

1. INTRODUCTION

As agents gain complexity and autonomy, automatic learning and optimization methods become attractive, as (a) they can improve and refine human intuition, especially in complex, dynamic environments, and (b) they demand significantly less labor to adapt to changes in the agent and environment. As most complex systems naturally decompose into smaller sub-units, for learning within such systems, it becomes convenient, even beneficial, to explicitly recognize their decomposition. In this paper we investigate the learning of agent behavior that can be decomposed into a *sequence* of atomic skills. Specifically we focus on optimizing multiple skills *within* each agent, and present a learning architecture for a humanoid robot soccer agent, which is fully deployed and tested within the RoboCup [3] 3D simulation environment, as a part of our team, UTAustinVilla.

In general, factors such as nonstationarity make it hard to provide strong theoretical guarantees when learning multiple behaviors. Therefore it becomes relevant to investigate such learning through empirical means. Our case study is performed within a complex domain, with realistic physics, state noise, multi-dimensional actions, and real-time control. In our test domain, teams of six autonomous humanoid robots play soccer in a physically realistic environment. Although each robot is ultimately controlled through low-level commands to its joint motors, we devise primitives for skills such as walking, turning, and kicking. In turn, such skills are strung together for implementing higher-level behaviors such as GoToTarget() and DriveBallToGoal(). It is quite clear that a behavior such as DriveBallToGoal() will be more successful if the robot can walk fast, turn quickly and sharply, and kick the ball with speed and accuracy. On the other hand, a very fast walk might tend to lead to a fall when transitioning into a turn; kicks lose their potency if the robot cannot accurately position behind the ball through precise side-walking and turning. The key idea in this paper is that skills can be optimized while respecting the tight coupling induced over them by high-level behaviors.

Robot soccer has served as an excellent platform for testing learning scenarios in which multiple skills, decisions, and controls have to be learned by a single agent, and agents themselves have to cooperate or compete. Although there is a rich literature based on this domain, most reported work primarily addresses (a) low-level concerns such as perception and motor control [5, 17], or (b) high-level decision-making problems [11, 19]. Thus the first contribution of our paper is a general methodology for optimizing the intermediate stratum of skills in an agent's control architecture. The volume

of the space thus optimized (hundreds of parameters) indeed marks a qualitative shift from a predominantly hand-coded approach for agent development to one significantly based on learning.

A second contribution of our paper is the light it sheds on designing objective functions ("fitness" functions) for optimization. On the one hand, "raw" statistics such as the precision and speed of soccer skills do not yield skills that operate well in unison. On the other hand, true objectives such as goal difference and win-loss record are too noisy to use effectively as a signal for learning. We demonstrate that carefully designed intermediate objectives, which require optimizing sequences of skills, can promote learning to achieve high-quality performance. An example of such an objective is the minimization of the time to score a goal on an empty field.

Finally, as an empirical contribution, we conduct detailed and extensive experiments related to our investigation. In particular, we compare several existing optimization methods, and find CMA-ES [8], a relatively recent addition to the literature, to be the most robust and effective. We also show evidence that conjunctive skill optimization can yield a very competitive soccer agent. The agent we develop here, which is based on, and motivated by the UTAustinVilla 2010 RoboCup agent, ranks among the top 8 teams from the RoboCup 2010 competitions.

The remainder of this paper is organized as follows. In Section 2 we describe the 3D simulation environment for humanoid robot soccer, along with the architecture of our agent. Section 3 describes how individual skills are parameterized and set up for optimization through several candidate methods. Section 4 then presents our methodology for optimizing these skills in sequence. Comprehensive experimental results are presented both in Section 3 and in Section 4. We conclude the paper with a summary and discussion in Section 5.

2. DOMAIN DESCRIPTION

The RoboCup 3D simulation environment is based on SimSpark[4], a generic physical multiagent system simulator. SimSpark uses the Open Dynamics Engine[2] (ODE) library for its realistic simulation of rigid body dynamics with collision detection and friction. ODE also provides support for the modeling of advanced motorized hinge joints used in the humanoid agents.

The robot agents in the simulation are homogeneous and are modeled after the Aldebaran Nao robot [1], which has a height of about 57 cm, and a mass of 4.5 kg. The agents interact with the simulator by sending actuation commands and receiving perceptual information. Each robot has 22 degrees of freedom: six in each leg, four in each arm, and two in the neck. In order to monitor and control its hinge joints, an agent is equipped with joint perceptors and effectors. Joint perceptors provide the agent with noise-free angular measurements every simulation cycle (20 ms), while joint effectors allow the agent to specify the direction and speed (torque) in which to move a joint. Although there is no intentional noise in actuation, there is slight actuation noise that results from approximations in the physics engine and the need to constrain computations to be performed in real-time. Visual information about the environment is given to an agent every third simulation cycle (60 ms) through noisy measurements of the distance and angle to objects within a restricted vision cone (120°). Agents are also outfitted with noisy accelerometer and gyroscope perceptors, as well as force resistance perceptors on both feet. Additionally agents can communicate with each other every other simulation cycle (40 ms) by sending messages limited to 20 bytes. Figure 1 shows a visualization of the Nao robot and the soccer field during a game.

Agent Skills

At the lowest level of control, each robot is operated by specifying torques to its joints. As a more convenient abstraction, we implement PID controllers for each joint, which take as input a desired *target angle* and compute the appropriate torque for achieving it. In turn, skills use the PID controllers as primitives. The set of skills needed to develop a successful agent, and the focus of this paper, include walking (forwards, backwards, and sideways), turning, kicking, standing, goalie-diving and getting up after falling. Further, it is useful to explicitly breakdown skills such as walking forwards into several different speeds. Whereas we are able to manually program fairly successful goalie-diving and getting up skills, effective locomotion and kicking skills are harder to develop manually: in contrast to getting up and goalie-diving, successful locomotion and kicking require a combination of dynamic balancing, precision and high speed. Locomotion skills further need to be able to transition well to and from other skills. Thus, for these skills we devise templates with parameters, which are subsequently optimized.

Bipedal locomotion has long been an active area of research. Pratt's thesis [16] provides an excellent overview of the field; Katić and Vukobratović [12] specifically survey intelligent control techniques used therein. A majority of the literature on bipedal locomotion focuses on model-based approaches. For instance, a humanoid robot is commonly modeled as an inverted pendulum [9], whose dynamics can be analyzed and used to plan trajectories. Recent approaches have also considered learning more complicated models, such as Poincaré maps [15]. Analytical modeling has indeed resulted in classical techniques — such as monitoring the "Zero Moment Point" of the robot [21] — which can resist noise in sensing, planning, and actuation, and small irregularities on the walking surface [14]. Even without explicit modeling of the dynamics, deviations from the intended trajectory can be constantly corrected through "closed-loop" control [7].

"Open-loop" approaches that do not rely on corrective feedback are typically simpler to implement and tend to

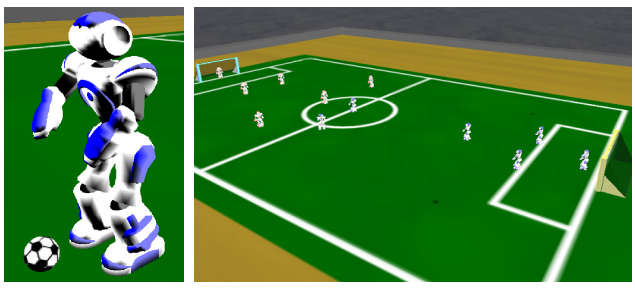


Figure 1: A screenshot of the Nao humanoid robot (left), and a view of the soccer field during a 6 versus 6 game (right).

yield faster walks, even if they are less robust to disturbances. However, in our simulation there is only minor noise in sensing or actuating joint angles (note that vision percepts are still noisy), and the soccer field is perfectly flat. Consequently we find it effective to develop open-loop skills for our agent. It must be noted that although the absence of significant actuation noise simplifies skill-development in our 3D simulation environment, in compensation the domain necessitates the development of an entire *suite* of soccer-related skills: multi-directional walks, turns, and kicks. Thus simulation enables us to investigate a concept that is relatively unexplored in the mainstream bipedal control literature. Even the few learning approaches within the 3D simulation environment have mainly been in the context of straight walking [18].

Each of our open-loop skills is implemented as a periodic state machine with multiple *key frames*, where a key frame is a static pose of fixed joint positions. To provide us flexibility in designing and parameterizing skills, we design an intuitive skill description language that facilitates the specification of key frames and the waiting times between them. Below is an illustrative example describing the WalkFront skill (further explained in Section 3).

```
SKILL WALK_FRONT

KEYFRAME 1
reset ARM_LEFT ARM_RIGHT LEG_LEFT LEG_RIGHT end
setTarget JOINT1 $jointvalue1 JOINT2 $jointvalue2 ...
setTarget JOINT3 4.3 JOINT4 52.5
wait 0.08

KEYFRAME 2
increaseTarget JOINT1 -2 JOINT2 7 ...
setTarget JOINT3 $jointvalue3 JOINT4 (2 * $jointvalue3)
wait 0.08
.
.
.
```

As seen above, joint angle values can either be numbers or be parameterized as `$<varname>`, where `<varname>` is a variable value that can be loaded after being learned. Note that due to left-right symmetry, some of these parameters influence multiple key frames.

Before proceeding to details about our skill optimization, it is relevant to observe that alternative parameterizations of skills could also be conceived. For example, rather than direct control of joints, foot trajectories could be parameterized and tracked using inverse kinematics [13]. We plan to explore such variations in future work.

3. OPTIMIZING INDIVIDUAL SKILLS

In this section we describe our optimization of the forward walking skill, which essentially illustrates the basic procedure adopted for optimizing any of our skills. As a starting point for subsequent optimization, we achieve a relatively stable front walk by programming the robot to raise its left and right feet alternately to a certain height above the ground, swinging them slightly forward, and then retracting them to their initial configurations. Such a hand-coding exercise for our various skills results in slow but stable skills, which are not very competitive themselves, but which serve as useful seeds for further optimization. Our walk consists of four key frames through which the agent periodically loops. General intuition for a straight and stable walk suggests that the legs should move in a symmetric and periodic manner. For this reason the joint positions of our first two frames are the same as our next two, except that the positions of the left and right legs are appropriately mirrored. Based on informal experimentation we decide to optimize three joint positions in each leg for each key frame, as they appear to be the most meaningful for a forward walk. These joints are the hip moving the leg forward and backwards, knee, and ankle moving the foot up and down. This provides a 12-dimensional parameter space to optimize, as we have 6 joint positions for each frame (3 for each leg), across two frames (as frames 3 and 4 are just mirrored values of frames 1 and 2). See Figure 2 for screenshots with the joints we are optimizing circled. We set the time to transition between key frames to be 80ms. This time was also determined by informal experimentation and gives the agent a walk cycle duration of 320ms (4×80ms).

In order to evaluate the performance of a forward walk, we measure the distance in the forward direction the agent can travel in 15 seconds. Our performance metric of displacement in the forward direction not only rewards speed, but it also encourages straight walks (as the shortest distance to walk is a straight line) and penalizes for lack of robustness (if the agent falls over it takes several seconds for it to stand up again). These measurements are taken in an automated fashion. Our setup on a distributed computing cluster allows us to run massive amounts of simulations in parallel, which is necessary in order for our learning algorithms to complete in a reasonable amount of time. In our experiments we used Condor [20] as a convenient tool for batch job processing on a cluster.

We compare the performance of four machine learning algorithms while trying to optimize the parameter values for our different skills. The algorithms we test are hill climbing (HC), cross-entropy method (CEM) [6], genetic algorithm (GA), and covariance matrix adaptation evolution strategy (CMA-ES) [8]. These algorithms are evolutionary (or "policy search") in nature and thus involve learning values incrementally across multiple generations of a fixed population size, where the individuals of a population consist of sets of parameter weights. As a baseline performance measure we also sample parameter values using random weight guessing (RWG). Due to noise in the simulator there can be considerable variance in the performance of a skill from one instance to the next using the same set of parameter weights. In order to account for this variability in performance we conduct multiple runs of the same parameter sets and take the average of these values when evaluating their performance.

Among these algorithms we try different configurations for the number of generations, population size, and the number

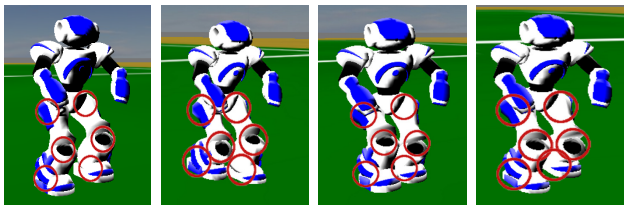


Figure 2: Nao robot walk frames with the joints we are optimizing circled.

of samples we average across to determine the performance of a set of parameter values. In order to make as fair a comparison as possible among the algorithms we allocate each of them the same "sample size" (the total number of fitness evaluations taken for different sets of parameter values). For the machine learning algorithms the sample size is equal to the product of the number of generations, the population size, and the number of measurements we average over in determining a parameter set's performance. For random weight guessing the sample size is equal to the number of guesses performed multiplied by the number of measurements used to compute average performance for each parameter set. For the experiments we shortly report, we fix this total sample size at 15,000 samples.

After testing the algorithms over many configurations we find CMA-ES to be the most successful for learning skills in our setup. Our results, shown in Figure 3(a), are averages over at least five runs of each algorithm using the configuration with which each performed the best. The distance values reported are the average measurement for ten runs of the best parameter set learned by each algorithm taken after the algorithm is finished running. The post-learning reevaluation of a parameter set's performance is necessary because of the noise in the simulator, and resulting potential bias toward configurations with less averaging samples to report an inflated performance value influenced by just a few lucky high outlier measurements. We find that GA and CEM do the best with 30 generations and a population size of 100 averaged across 5 samples, while HC and CMA-ES perform better with 50 generations and a population size of 30 averaged across 10 samples. Random weight guessing performs best when guesses are evaluated by averaging across 5 fitness trials.

Apart from good performance, another advantage we find with using CMA-ES is its low configuration overhead. All that is needed to be specified for CMA-ES are initial mean and standard deviations for each parameter. The mean values are just our seed values and we find that CMA-ES performs well over a reasonably large range of standard deviation values. The other algorithms' performances are more dependent on their algorithm-specific parameter settings. For HC we get the best values when using an initial step size of 10° and a linear step size decay. For GA we find that bounding the search space at a maximum of 30° from the seed joint angles gives us the best performance. CEM, like CMA-ES, also requires a standard deviation for each parameter. However, CEM's performance seems to be more dependent on the values chosen to initialize these standard deviations. In contrast, CMA-ES is less affected by these initial values due to the way it maintains and adjusts them across generations using covariance analysis. We determine 30° to be a good standard deviation for CEM. We also achieve our best performance using a standard deviation of 30° for random weight guessing which selects values from Gaussian distributions centered around our initial seed for each parameter.

As CMA-ES is found to perform significantly better than the other algorithms, we describe here in more detail the experiments conducted with it. Each experiment includes 15,000 sampling runs, in which we vary the learning configuration values of population-size, number of generations, and number of averaging runs that are executed for each parameter set generated by the algorithm. This means that for each configuration, the population-size times the number of generations times the number of averaging runs is fixed at 15,000. As the sample size is always fixed, when defining a configuration we face a trade-off: averaging over more runs gives a more confident fitness value for each parameter set, but decreases the number of generations and/or the population size we can use. Averaging over 1, 2, 5, and 10 runs, we try 14 different configurations, presented in Figure 3(b). The configuration that presents the best balance between its three factors, uses 50 generations, a population size of 30, and 10 averaging runs for each candidate parameter set. Its fitness value is 12.16 m/15sec (0.81 m/s), with a standard error of 0.38 m/s. A learning curve corresponding to this configuration is presented in Figure 3(c).

The highest speeds we are able to achieve when learning a front walk require a configuration with roughly three times the number of samples used in the experiments above (45,000). On our Condor-based system, such a run takes 5-7 hours. Table 1 shows the best results we achieve when optimizing each of our main skills. To the best of our knowledge these results are among the fastest that have been achieved in our domain. Unfortunately, there are not many references in the literature that describes other teams' walk speeds; and the only report we are aware of is that of Shafii *et al.* [18]. In comparing our learned skills with other teams' using the released agent binaries from RoboCup 2010, we observe a clear advantage of the performance statistics we report here over those of other teams' skills. As expected, our performance statistics also better those achieved in hardware on Nao robots [10] due to the simplified modeling of our simulator.

4. OPTIMIZING SEQUENCES OF SKILLS

Whereas the results from Table 1 signify that our parameterized skills can effectively be optimized using CMA-ES, the job of deploying these skills to play soccer remains unfinished. Fast locomotion skills, however stable they are when executed individually, result in frequent falls of the robot if integrated directly. To see why, consider a typical log of the skills invoked (every 320 ms, as described in the previous section) by the agent during soccer play:

> ... WalkFront, WalkFront, Turn(R), Turn(R), Turn(R), WalkFront, WalkFront, WalkFront, Turn(L), Turn(L), WalkBack, WalkBack, ...

The trace shows that skills are highly interleaved, with frequent transitions between them. In game scenarios, the same skill is seldom executed for more than a few consecutive cycles. Therefore, optimizing skills in isolation does not

Table 1: Performance statistics for various skills optimized using CMA-ES. In this table and all subsequent ones, entries within parentheses correspond to one standard error.

Skill	Statistic	Performance
WalkFront	Speed	1.07(.00) m/s
WalkBack	Speed	1.03(.00) m/s
WalkSide	Speed	.62(.01) m/s
Turn	Angular speed	112.03(.24) °/s
Kick	Ball displacement	5.09(.07) m

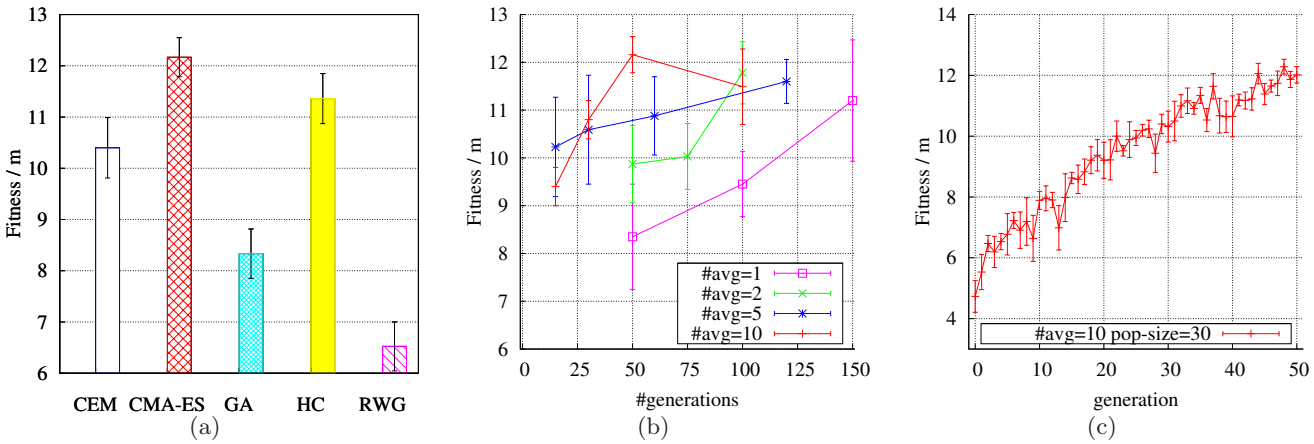


Figure 3: Experimental results from optimizing WalkFront. Plots show the fitness values — the distance traveled in 15 seconds — achieved by various learning algorithms and algorithm-specific parameter settings. In all algorithms the sample size is fixed to 15,000 simulation runs. For evolutionary algorithms this means that #generations × #avg × population-size = 15,000. Plot (a) shows the best performance achieved by various methods. Plot (b) shows the performance achieved by CMA-ES under various settings of #generations and #avg, while (c) shows the progress of learning under the best CMA-ES configuration with training time. Error bars in all plots correspond to one standard error.

necessarily benefit their *combined operation*.

In order to optimize sequences of skills to work together, carefully designed constraints are necessary. We begin by revising the evaluation criterion used by the learning process. Ideally, when learning a skill, it would be best to evaluate it with respect to our ultimate goal: the team's win-loss record or mean goal difference against a set of opponents. However, as these are extremely noisy measures, the number of runs needed in order to obtain reliable performance estimates becomes impractical. A much less noisy measure, which still aligns well with the team's objective, is the time taken by a single agent to score a goal on an empty field. We denote this behavior DriveBallToGoal(), and the associated evaluation metric time-to-score. Pseudo-code for DriveBallToGoal() is as follows:

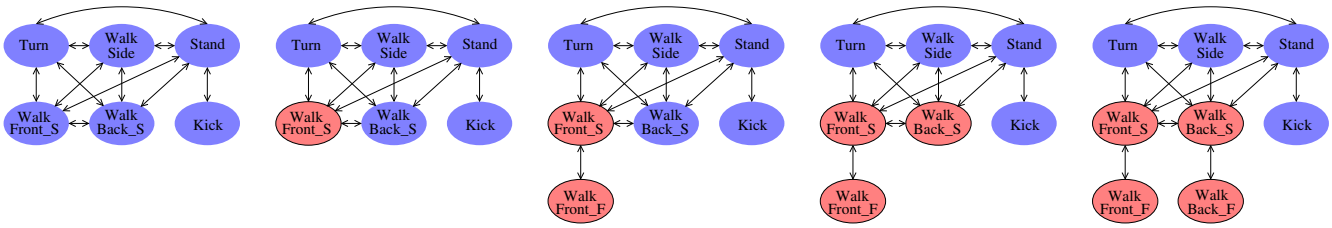
```
function DriveBallToGoal()
  if robotDistanceFromBall > threshold_0
    getRoughlyBehindBall()
  else
    chooseKickDirectionAndType()
    computeThresholdsForPositioning()
    # Position to kick / dribble:
    if distanceToPosition > threshold_1
      walkFront()
    elseif robotOffsetFromKickDirection > threshold_2
      turn()
    elseif lateralLegAlignementWithBall > threshold_3
      sideWalk()
    else
      kickOrDribble()
```

We use this behavior for our evaluations, as it achieves a good balance between eliminating noisy effects such as the actions of other players, while still requiring the agent to combine its basic skills in a complex, realistic manner. Later in this section, we show empirical results validating the choice of time-to-score as an evaluation metric while optimizing skills.

Several skills are used during a learning evaluation through DriveBallToGoal(). However, it would be inefficient to try and learn all of them at once, due to the high dimensionality of the search space (roughly 100 – 150 parameters). Instead we use a more efficient approach, which learns one skill (roughly 12 parameters) at a time, while keeping others fixed. This process results in a sequence of incremental improvements in the agent, with the crucial invariant property that at any time all the skills work well together. In particular the optimization process improves the agent's speed while keeping it stable, as falls typically result in poor time-to-score values. In turn, the amount each individual skill can be optimized is limited by the need to cooperate with other skills.

Apart from goalie dives and getting up skills, all the skills used by our final agent are optimized. Yet, for the purposes of this paper, we present an isolated study of our optimization procedure involving only forward and backward walks, namely WalkFront and WalkBack, respectively. We start with a base agent that uses basic, hand-coded versions of these skills. Let us call this agent A0. Under A0 these skills are not very fast, but they ensure relative stability during locomotion and skill transitions. The idea is to use A0 as a seed for successive optimizations. Figure 4(a) presents a skill transition diagram, which shows the main skills of agent A0 along with the legal transitions between them (marked by arrows). Notice that the agent can only invoke Kick if it is already standing; nor can it transition into a skill other than Stand after executing Kick. In Figure 4(a) the walking skills of A0 are suffixed "_S" to denote that they are "slow".

We improve upon A0 in five incremental steps, each step creating a new agent based on the agent that resulted from the previous step. We denote the resulting agents A1, A2, A3, A4, and A5. The first improvement, A1, is created from



(a) A0 (b) A1 (c) A2 (d) A3 (e) A4

Figure 4: Constraints on transitions between skills represented as state diagrams. For Agent A0 neither the WalkFront_S nor the WalkBack_S skills is optimized; the former is optimized (shown with thick border) under A1. Further skills are added and optimized subsequently under agents A2, A3, and A4. Agent A5 is identical to A4, except for retuning thresholds and the logic for selecting and invoking our new learned skills.

A0 by optimizing "WalkFront_S" using CMA-ES, under the time-to-score measure. Consider that while WalkFront_S is being optimized under this measure, we are searching for a set of parameters that both improve speed and maintain stability. The need to maintain stability while cooperating with all other skills puts multiple constraints on WalkFront_S and therefore limits how fast WalkFront_S can get. We address this problem in A2, by "decoupling" from WalkFront_S an additional skill called WalkFront_F ("F" denoting "fast"). As seen in Figure 4(c), we constrain the behavior of agent A2 such that WalkFront_F can only be invoked following WalkFront_S, and to transition to any other skills, it must first transition into WalkFront_S. The skills WalkFront_S and WalkFront_F have exactly the same template, and initially the same parameter values. However, optimizing the parameters of WalkFront_F after first optimizing WalkFront_S (under A1) allows the agent to achieve greater speed while retaining its stability. These properties result from the fact that WalkFront_F is unconstrained by most of the skills that constrain WalkFront_S.

Results in Section 4 demonstrate tangible gains consistent with our progressive refinements from A0 to A1 to A2. Indeed the trend is carried forward to agents A3 (Figure 4(d)) and A4 (Figure 4(e)), which are obtained based on a similar decoupling procedure applied to the WalkBack skill. Recall that agents A1 through A4 are all obtained solely by optimization of one skill at a time, starting from the seed agent A0. To obtain our final agent, A5, we take A4 and *manually* retune thresholds and the logic for selecting and invoking our new learned skills in order to utilize them to their full potential. For example, a change in skill speeds can change the robot's stopping distance, which in turn affects the threshold for the decision of whether to continue WalkFront, as can be seen in the DriveBallToGoal() pseudo-code. While the tuning is done here manually, it could potentially be automated and learned. However, in this paper we focus on skill learning, and leave the learned tuning as possible future work.[1]

Note that agents A0 through A5 all use the same skills, apart from WalkFront and WalkBack. The turns and side walks used were also optimized in the manner described above and were already integrated into our agent A0. It is worth mentioning, however, that time-to-score does *not* serve as an ideal fitness measure while optimizing kicks, as the kick skill is used only a small fraction of time, and most of the time is spent on locomotion and positioning behind the ball. Since Kick is only executed after an intermediate Stand skill, we optimize kicks by starting the robot behind the ball, using the distance covered by the ball in the kick direction as an informative evaluation measure.

[1]Videos showing optimized skills and behavior are provided at the following URL: `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/AustinVilla3DSimulationFiles/2010/html/skilloptimization2010.html`.

Experimental Evaluation

We have just described how we used two main ideas for learning and optimizing skills: the idea of optimizing a skill under the constraints of cooperating with other skills, and the idea of skill decoupling. The remainder of this section shows that our skill optimization process achieved tangible gains, that were reflected directly in the agent's performance with respect to its ultimate objective: its win-loss record or goal difference against a set of opponents.

We ran three sets of experiments, in which we measured our agent both with respect to the time-to-score measure and with respect to its actual game performance, and compared the results with released binaries from RoboCup 2010. In the first set of experiments we measured the progress achieved by each step of our optimization process, which started from the seed agent A0, continued by creating the agents A1-A4 by optimizing one skill at a time, and finally tuned A4 to be the final agent A5. Table 2 shows the results of playing agents A0-A5 against each other in full 6 vs. 6 games. In this setup, each of the players in a team is played as the same agent, namely one of A0, A1,..., A5. Each cell in the table shows the mean goal difference along with the standard error, averaged over 100 full games. It can be seen that every agent outperforms its predecessors. This result demonstrates how our skill-optimization process indeed achieved better game performance.

Table 2: Game results between agents A0 through A5. Entries show the goal difference (row – column) from 10 minute games.

	A0	A1	A2	A3	A4
A5	2.11(.10)	.77(.10)	.70(.10)	.58(.09)	.48(.08)
A4	1.66(.10)	.46(.08)	.15(.07)	.03(.07)	
A3	1.67(.10)	.28(.08)	.01(.08)		
A2	1.33(.10)	.20(.07)			
A1	1.23(.10)				

In the second set of experiments we compared the time-to-score performance of our initial agent A0, our final agent A5, and the set of all released agent binaries from RoboCup 2010 we were able to run on our computers. In each experiment, we placed the ball in the middle of the field, which is 9 m from the goal, and then placed the agent 1 meter behind the ball. We then measured the time it takes the agent to score a goal. Table 3 shows the mean time it takes the agents to score from this position, averaged over 500 runs, along with the standard error. Our agent A5 is ranked second with a mean time of 34.49 seconds, whereas the top agent's mean time to score is 31.08 seconds. Note that A0 is ranked in the middle of the table with a time of 63.52. Agents A1–A4, which are not shown in the table achieved times that rank them between A0 and A5.

In our third set of experiments, we tested our agents A0 and A5 in playing full 6 vs. 6 games against the released RoboCup 2010 agent binaries. The results are shown in Table 4. The leftmost column shows the row agent's rank in RoboCup 2010. The rightmost columns show the results achieved by agents A0 and A5, when playing against RoboCup binaries. Each cell shows the mean goal difference between a column agent and a row agent, averaged over 100 full games, along with the standard error. Note that negative values (in bold) mean a positive goal difference for our agent, therefore the bolded part of the table is where our agent performed better than the row agent.

Two interesting facts can be observed in Table 4. The first one is the correlation between the actual game performance and the time-to-score measure (Table 3). An agent, whether our agent or another team's agent, with good game performance usually had good time-to-score performance. Recall that while optimizing our agent's skills, we used the time-to-score measure along with the DriveBallToGoal() behavior as a less-noisy alternative for measuring real game performance. Here we confirmed that while doing so, much of the complexities of real game scenarios that are relevant to skills execution were still retained. Therefore the time-to-score measure is both effective, as it correlates with game performance, and efficient, due to the reduced noise. However, note that the correlation is not expected to be perfect: in real games there are factors like decision-making strategies, formations, defensive tactics and more, that affect the game performance, but do not reflect in the DriveBallToGoal() behavior. The second interesting fact is that our final agent, A5, was ranked in the table among the top 8 teams of RoboCup 2010. As this ranking was achieved mainly using our skill optimization process, with some additional tuning, this demonstrates the effectiveness of our suggested method of optimizing skills under constraints.

Table 3: Time to score on an empty field, starting the center of the field. Each row corresponds to A0, A5, or an agent from the RoboCup 2010 competition. Averages are over 500 runs.

Agent	Time-To-Score/s
Apollo3d	31.08 (1.46)
A5	**34.49 (0.89)**
RoboCanes	36.18 (1.40)
NaoTH	36.75 (1.63)
UTAustinVilla	37.20 (0.89)
FCPortugal	47.54 (1.94)
SEURedSun	52.11 (2.49)
A0	**63.52 (1.05)**
Little Green Bats	71.02 (1.96)
FutK	77.89 (4.19)
BeeStanbul	98.56 (3.63)
Nexus3D	152.76 (5.15)
RoboPub	291.86 (1.17)
NomoFC	295.48 (1.32)
Bahia3D	300.01 (0.00)
Alzahra	300.01 (0.00)

Table 4: Full game results, averaged over 100 games. Each row corresponds to an agent from the RoboCup 2010 competitions, with its rank therein achieved. The two rightmost columns correspond to our base agent A0 and final agent A5, respectively. Entries show the goal difference (column – row) from 10 minute games. Goal differences in favor of A0 and A5 are shown in bold.

Rank	Team	A0	A5
1	Apollo3d	-4.29 (.17)	-1.88 (.13)
2	NaoTH	-3.79 (0.14)	-1.85 (0.10)
4	BoldHearts	-3.15 (0.13)	-0.08 (0.11)
5-8	SEURedSun	-1.93 (0.13)	-1.16 (0.1)
5-8	RoboCanes	-1.81 (0.12)	-0.38 (0.09)
5-8	FCPortugal	-1.57 (0.11)	**0.43** (0.09)
9-16	UTAustinVilla	-1.54 (0.09)	**0.9** (0.09)
9-16	FutK	-0.23 (0.06)	**2.14** (0.1)
9-16	BeeStanbul	**0.76** (0.07)	**4.08** (0.11)
9-16	Nexus3D	**1.67** (0.06)	**4.08** (0.09)
9-16	Little Green Bats	**1.84** (0.08)	**5.0** (0.11)
9-16	NomoFC	**3.62** (0.09)	**7.07** (0.09)
17-20	Bahia3D	**3.59** (0.08)	**7.49** (0.1)
17-20	RoboPub	**5.25** (0.08)	**7.92** (0.1)
17-20	Alzahra	**6.39** (0.08)	**10.59** (0.09)

5. SUMMARY AND DISCUSSION

In several practical tasks an agent's behavior is composed of qualitatively distinct components. Can this natural decomposition be used as a means to scale learning to complex tasks? In this paper we presented a successful case study of doing so in the context of humanoid robot soccer. In particular we focused on the intermediate "skills" layer of a soccer agent's architecture. Together, the skills of a soccer agent constitute a rich and complex aspect of behavior, which it would be impractical to optimize as a single monolithic block. We carefully engineered skills and rules for transitions, and showed that optimizing components in an incremental manner could significantly improve performance. Each skill has 10–20 parameters; overall the number of parameters optimized is around 100–150.

We believe our case study is a compelling example for the methodology of decomposing a large learning problem into components and devising informative objective functions. Several practical systems resemble a soccer agent's control hierarchy, and often are indeed evaluated ultimately through success (win) and failure (loss). This paper also leads to recommendations for an optimization framework

and experimental support for the CMA-ES algorithm, which can serve as a useful starting point for related undertakings.

The RoboCup 3D simulation environment engenders the novel research question of developing a suite of interacting humanoid robotic skills, a relatively unexplored question in the literature, which this paper addresses. Our demonstration specifically finds appeal for developing humanoid robot soccer teams by investing significantly in learning and optimization. The architecture we presented here was a main building block in developing our team, UTAustinVilla, and the agents we presented here were motivated by, and based on, our UTAustinVilla 2010 RoboCup agent. Our detailed experimental results provide conclusive evidence for the improvements achieved with each incremental optimization, and the final agent we develop (agent A5) ranks among the top eight teams from the RoboCup 2010 competitions. The human labor involved in developing our agent is relatively low compared to the CPU time spent optimizing skills, which is on the order of 100,000 hours.

In future work we intend to further extend the scope of learning within our agent by replacing currently hand-coded components (such as fine positioning and getting up). For our basic locomotion skills, it is also relevant to consider alternative parameterizations that involve closed-loop control and inverse kinematics. Such approaches are likely to eventually extend the reach of our learning paradigm to hardware platforms by using simulators that model physical robots more precisely. Additionally we can seek to further refine our coupled set of learned skills by using them as a seed for our optimization framework. By continuing to optimize the coupled skills in an alternating and iterative manner, where they are learned in the context of previously optimized skills, it is likely that further improvements to them can be realized.

Acknowledgements

We thank Suyog Dutt Jain for his contributions to early versions of this work. This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), DARPA (FA8650-08-C-7812), and the FHWA (DTFH61-07-H-00030). This research was also supported in part by the NSF under CISE Research Infrastructure Grant EIA-0303609.

6. REFERENCES

[1] Aldebaran Humanoid Robot Nao. `http://www.aldebaran-robotics.com/eng/`.

[2] Open Dynamics Engine. `http://www.ode.org/`.

[3] RoboCup. `http://www.robocup.org/`.

[4] SimSpark. `http://simspark.sourceforge.net/`.

[5] S. Behnke, M. Schreiber, J. Stückler, R. Renner, and H. Strasdat. See, walk, and kick: Humanoid robots start to play soccer. In *Proceedings of the Sixth IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, pages 497–503. IEEE, 2006.

[6] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, February 2005.

[7] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng. Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot. *International Journal of Robotics Research*, 27(2):213–228, 2008.

[8] N. Hansen. *The CMA Evolution Strategy: A Tutorial*, January 2009. `http://www.lri.fr/~hansen/cmatutorial.pdf`.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by a simple three-dimensional inverted pendulum model. *Advanced Robotics*, 17(2):131–147, 2003.

[10] S. Kalyanakrishnan, T. Hester, M. Quinlan, Y. Bentor, and P. Stone. Three humanoid soccer platforms: Comparison and synthesis. In *RoboCup 2009: Robot Soccer World Cup XIII*, pages 140–152. Springer, 2010.

[11] S. Kalyanakrishnan and P. Stone. Learning complementary multiagent behaviors: A case study. In *RoboCup 2009: Robot Soccer World Cup XIII*, pages 153–165. Springer, 2010.

[12] D. Katić and M. Vukobratović. Survey of intelligent control techniques for humanoid robots. *Journal of Intelligent Robotic Systems*, 37(2):117–141, 2003.

[13] N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion. In *The Nineteenth National Conference on Artificial Intelligence*, pages 611–616. AAAI Press, 2004.

[14] C. Meriçli and M. Veloso. Biped walk learning through playback and corrective demonstration. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 1594–1599. AAAI Press, 2010.

[15] J. Morimoto and C. G. Atkeson. Nonparametric representation of an approximated Poincaré map for learning biped locomotion. *Autonomous Robots*, 27(2):131–144, 2009.

[16] J. E. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2000.

[17] M. Riedmiller, T. Gabel, R. Hafner, and S. Lange. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, 2009.

[18] N. Shafii, L. P. Reis, and N. Lao. Biped walking using coronal and sagittal movements based on truncated Fourier series. In *Proceedings of the Fifth Doctoral Symposium in Informatics Engineering, (DSIE 2010)*, pages 79–90, Porto, Portugal, January 2010. Faculdade de Engenharia, Universidade do Porto.

[19] P. Stone. *Layered Learning in Multi-Agent Systems*. PhD thesis, School of Computer Science, Carnegie Mellon Univerity, Pittsburgh, PA, USA, December 1998.

[20] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the Condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.

[21] M. Vukobratović and B. Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2005.

Metric Learning for Reinforcement Learning Agents

Matthew E. Taylor, Brian Kulis, and Fei Sha
Lafayette College, taylorm@lafayette.edu
University of California, Berkeley, kulis@eecs.berkeley.edu
University of Southern California, feisha@usc.edu

ABSTRACT

A key component of any reinforcement learning algorithm is the underlying representation used by the agent. While reinforcement learning (RL) agents have typically relied on hand-coded state representations, there has been a growing interest in *learning* this representation. While inputs to an agent are typically fixed (i.e., state variables represent sensors on a robot), it is desirable to automatically determine the optimal relative scaling of such inputs, as well as to diminish the impact of irrelevant features. This work introduces HOLLER, a novel distance metric learning algorithm, and combines it with an existing instance-based RL algorithm to achieve precisely these goals. The algorithms' success is highlighted via empirical measurements on a set of six tasks within the mountain car domain.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Miscellaneous

General Terms

Algorithms, Performance

Keywords

Reinforcement Learning, Distance Metric Learning, Autonomous Feature Selection, Learning State Representations

1. INTRODUCTION

In *Reinforcement Learning* (RL) problems, an agent must learn to select sequences of actions to maximize a reward signal. The agent's decision process is state-dependent — the effects of an action will depend on the agent's location in an environment. The agent's state representation is a critical component in a successful agent, but state representations are typically designed by a human domain expert. The goal of this paper is to introduce a robust method to allow more autonomy in designing state representation, allowing the agent to scale dimensions of the state representation, as well as to potentially ignore irrelevant dimensions.

There has been some exciting recent work on learning to construct or scale state variables (c.f., proto-value functions [10]) but such methods typically assume a model of the task is known. Other work focuses on the placement and tuning of individual basis functions (c.f., learning where to place kernels [2]). In contrast, this work assumes that 1) the agent must efficiently sample the state space and construct its representation on-line and 2) the agent should learn a metric that should generalize across the entire state space, not just the region explored.

Rather than constructing new state variables, we assume that the state variables provided to the agent are sufficient to learn the current task, but that we do not know their relative weighting. For example, consider a robot that has a laser range finder that reads distances in meters and a sonar that reads distances in feet. It is likely that the two state variables will need to be scaled differently to accurately integrate their information. Likewise, if an agent is provided both its speed in meters/second and its acceleration in meters/second^2, the relative importance of these two variables on its estimate of location will need to be treated very differently.

Traditionally, state variables are scaled by normalizing all state variables to have the same range (e.g., $[-1, 1]$). For instance, consider the CMAC [1] function approximator, a type of tile coding used successfully in the mountain car domain [14]. CMACs can take an arbitrary groups of continuous state variables and lay infinite, axis-parallel tilings over them; a continuous state space is discretized while maintaining the capability to generalize via multiple overlapping tilings. However, the number of tiles and width of the tilings are hardcoded by a domain expert, which necessitates knowing both the ranges (to normalize) and relative importance of the different state variables (to determine the spacing and number of tiles per dimension).

This work shows that it is possible to use *distance metric learning*, a popular supervised learning technique, to scale and select state variables automatically from data gathered via agent experience. Experiments show that our theoretically grounded on-line metric learning can result in significantly improved learning in a set of RL tasks situated in the mountain car domain. Our hope is that this work will encourage additional research into the integration of metric learning and RL, as well as to provide a powerful tool to help automatically determine effective state representations.

2. BACKGROUND

This section first introduces Reinforcement Learning, the setting for the paper. Next, Fitted R-MAX is discussed, an instance-based RL algorithm that will be used in this paper's experiments. Last, an introduction to distance metric learning provides background to understand HOLLER, our novel learning algorithm.

2.1 Reinforcement Learning

Reinforcement learning problems are typically framed as *Markov decision processes* (MDPs) defined by the 4-tuple $\{S, A, T, R\}$.

Cite as: Metric Learning for Reinforcement Learning Agents, Matthew E. Taylor, Brian Kulis, and Fei Sha, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 777-784.

An agent perceives the current *state* of the world $s \in S$ (possibly with noise). Tasks are often episodic: the agent executes actions in the environment until it reaches a terminal or goal state, at which point the agent is returned to a starting state. The set A describes the *actions* available to the agent, although not every action may be possible in every state. The *transition function*, $T : S \times A \mapsto S$, takes a state and an action as input and returns the state of the environment after the action is performed. The agent's goal is to maximize its reward, a scalar value defined by the *reward function*.

A learner chooses which action to take in a state via a policy, $\pi : S \mapsto A$. π is modified by the learner over time to improve performance, defined as the expected (discounted) total reward. Instead of learning π directly, many RL algorithms instead approximate the action-value function, $Q : S \times A \mapsto \mathbb{R}$, which maps state-action pairs to the expected real-valued return [16]. In tasks with small, discrete state spaces, Q and π can be fully represented in a table. As the state space grows, using a table becomes impractical, or impossible if the state space is continuous. Agents in such tasks typically factor the state using *state variables* (or *features*), so that $s = \langle x_1, x_2, \ldots, x_n \rangle$. In such cases, RL methods use *function approximators*, such as artificial neural networks or tile coding, where parameterized functions representing π or Q are tuned via supervised learning methods. The parameterization and bias of the function approximator define the state space abstraction, allowing observed data to update a region of state-action values rather than a single state/action value.

2.2 Fitted R-Max

The experiments in this paper focus on integrating a learned distance metric with Fitted R-MAX, an instance-based RL algorithm [7]. Fitted R-MAX approximates the action-value function, Q, for large or infinite state spaces by constructing an MDP over a small (finite) sample of states $X \subset S$. For each sample state $\mathsf{x} \in X$ and action $a \in A$, Fitted R-MAX estimates the dynamics of the transition function, $T(\mathsf{x}, a)$, using all available data for action a. The data from multiple nearby states will need to be integrated and generalized as it is unlikely that points in a continuous state space will be sampled enough to approximate all action transitions. A probability over predicted successor states in S, $T(\mathsf{x}, a)$, is first approximated. The distribution of successor states is then approximated with a distribution of states in X, resulting in a MDP defined over a finite size (X) that is formed based on data from the environment (S). Q is then approximated via dynamic programming.

For the purposes of the current work, the most important feature of Fitted R-MAX is that when T and R are estimated for a point x, data from nearby points are averaged together, weighted by their relative distances. That is, recorded instances that are (spatially) closer to x are assumed to be more predictive than instances further away. Rather than assuming that the similarity between points in the state space is Euclidean, this work learns a distance metric for Fitted R-MAX to use. A full description of Fitted R-MAX and its implementation can be found elsewhere [7].

2.3 Distance Metric Learning

Distance metric learning is a core machine learning problem that attempts to learn an appropriate distance function for a given task. Because distances or similarities are used in a variety of tasks — including clustering, similarity searches, and many classification algorithms — there has been significant interest in the design of algorithms for tuning distance functions. Typically these algorithms are at least partially supervised; in addition to the data, the algorithm receives constraints for the desired distance metric. Examples include constraints of the form "points x and y should have a small/large distance" or "points v and w should have a smaller distance than points v and x."

Metric learning algorithms typically attempt to construct a transformation of the data (either linear or non-linear) such that the constraints are satisfied after applying a standard distance function such as the Euclidean distance to the transformed data. The most popular approach is to learn a linear transformation of the data; these methods are often called *Mahalanobis metric learning* methods, and is the approach we employ in this work (c.f., [4, 5, 6, 19, 22]). These methods are desirable in that they show good generalization performance on a variety of problems, including in vision, text, and music domains (c.f., [3, 15]).

Recently, there has been interest in applying metric learning over large-scale data, or in cases when the standard methods that process a large set of constraints in a batch mode are inadequate. Such *online* algorithms instead process a single constraint at a time, and are designed to give comparable performance as compared to their offline counterparts. There has been recent theoretical progress in proving regret bounds for online learning methods, which provide worst-case guarantees on the performance of an online algorithm as compared to any corresponding offline algorithm [13, 23]. We pursue an online approach in this paper to avoid the computational cost of repeatedly applying offline learning methods to our data.

3. LEARNING THE DISTANCE METRIC

Algorithm 1 summarizes the process of learning and using a distance metric in an RL agent. There are three main steps which will be detailed in the following sections:

1. Collect data while the agent explores the environment.
2. Decide which states are "more similar," based on the relatedness of agent transitions.
3. Use state relatedness to calculate a distance metric: states which have similar transitions should be closer than states which have dissimilar transitions.

3.1 Collecting Data

Algorithm 1 is the top-level algorithm. It first initializes an agent (lines 1–4) and then has it interact with its environment for a single episode (lines 5–11), collecting data to be used for distance metric learning. Lines 12–31 consider triples of vectors, where a vector is defined by a pair of states which the agent has moved between (i.e., the difference between s' and s). Lines 18 and 19 consider sets vectors recorded at similar times (e.g., +/- *NumPts* actions).We restrict the vectors to be temporally similar under the assumption that transitions which occur in rapid succession are likely to be more similar than transitions that happen at very different times. This assumption is domain dependent, but will often be true, particularly when *NumPts* is set so that these vectors are also close spatially. However, even in "well behaved" domains there will be regions of the state space where this assumption will be violated (e.g., an agent may often move without obstruction, but be constrained when adjacent to a wall).

We only consider sets of three vectors $\langle v, w, x \rangle$ which have the same action (line 22), as transitions for different actions may be dissimilar. The similarities between vectors v and w, and between vectors v and x are calculated on lines 23 and 24, as discussed in the following section. Lines 27 and 30 add the triple to the set of current constraints, which are in the form "v is more similar to x than v is to w." Finally, after all the data from an episode has been processed, the distance metric is updated with the set of constants.

On lines 33 and 34, the algorithm can decide if more data needs to be collected. For instance, if any W_a has changed significantly

Algorithm 1 Main Algorithm (η)

1: $\pi \leftarrow$ random policy
2: *# initialize the dist. metric for each action*
3: $\forall a \in A, W_a \leftarrow$ Identity matrix (i.e., Euclidean distance)
4: $i \leftarrow 0$
5: $s \leftarrow$ initial state *# Begin an episode*
6: **repeat**
7: Execute $a = \pi(s)$
8: Observe r and s'
9: Save tuple $V_i \leftarrow (s, a, s')$
10: $s \leftarrow s'$
11: $i \leftarrow i + 1$
12: **until** s is a terminal state *# the episode ends*
13: **for** $j \in \{0, \ldots, i-1\}$ **do**
14: *# get vector for transition between state s_j and s'_j*
15: $v \leftarrow V_j.s' - V_j.s$ *#the vector from s to s'*
16: $a \leftarrow V_j.a$ *# the action in question*
17: $C_a \leftarrow \emptyset$ *# Set of constraints used to update W_a*
18: **for** $k \in \{j - NumPts, \ldots, j + NumPts\}$ **do**
19: **for** $l \in \{j - NumPts, \ldots, j + NumPts\}$ **do**
20: $w \leftarrow V_k.s' - V_k.s$ *# transition k vector*
21: $x \leftarrow V_l.s' - V_l.s$ *# transition l vector*
22: **if** $(a = V_k.a = V_l.a)$ and $(v, w, x$ are distinct) **then**
23: $re_w \leftarrow$ CALCRELATEDNESS(W_a, v, w)
24: $re_x \leftarrow$ CALCRELATEDNESS(W_a, v, x)
25: **if** $re_w > re_x$ **then**
26: *# Relatedness(v,w) > Relatedness(v,x)*
27: $C_a \leftarrow C_a \cup \langle v, w, x \rangle$
28: **else**
29: *# Relatedness(v,x) > Relatedness(v,w)*
30: $C_a \leftarrow C_a \cup \langle v, x, w \rangle$
31: *# update the distance metric*
32: $W_a \leftarrow$ HOLLER(W_a, C_a, η)
33: **if** more data needed for distance learning **then**
34: **goto** line 4
35: Learn a policy using an RL algorithm and W

during the last updated from the constraints, it is possible that more data is needed for W_a to converge. In this paper we instead run the algorithm with different numbers of data collection episodes to show how gathering additional data improves the estimate of W_a and, therefore, the speed of learning (line 35).

In general, collecting data from the environment can be interleaved with distance metric learning and with learning an action-value function. Algorithm 1 simplifies this approach. Rather than updating the distance metric on every time step, it is updated at the end of every episode. This is primarily an implementation detail to reduce the number of times the distance metric learning code (implemented in MATLAB) was called by the simulator (implemented in C).

3.2 Transition Similarity

Algorithm 1 reasons about pairs of vectors, where these vectors describe transitions in the state space: $s \rightarrow s'$. Algorithm 2 calculates the similarity of two vectors, given the current distance metric, where the relatedness of two vectors is at most 1.0 (if they are identical in direction and magnitude). This similarity will be used in the next section to calculate the distance metric under the assumption that states that have similar transitions (for the same action) should be closer in the state space than states that have dissimilar transitions.

Algorithm 2 CALCRELATEDNESS(W, x, y)

1: $\|x\| \leftarrow \sqrt{x^T W x}$
2: $\|y\| \leftarrow \sqrt{y^T W y}$
3: $m \leftarrow \frac{min(\|x\|, \|y\|)}{max(\|x\|, \|y\|)}$
4: $c = \frac{x^T W y}{\|x\| \|y\|}$
5: **return** $c \cdot m$

Algorithm 3 HOLLER(W, C, η)

1: **for** each constraint $\langle v, w, x \rangle \in C$ **do**
2: $W_{next} \leftarrow$ minimum over all W_{next} of:

$$D_{\ell d}(W_{next}, W) + \eta \cdot \max(d_{W_{next}}(v, w) - d_{W_{next}}(v, x) + 1, 0)$$

3: $W \leftarrow W_{next}$

3.3 The HOLLER Algorithm

HOLLER (Hinge loss Online Logdet LEarner for Relative distances), as presented in Algorithm 3, is used to learn a distance metric d_W from a list of constraints C and a learning rate η. Recall that each constraint $\langle v, w, x \rangle$ indicates that v should be closer to w than v is to x. The metric learning algorithm follows a standard online updating scheme: each constraint is visited once and the metric is updated after seeing each constraint. As in most online algorithms, we trade off conservativeness with correctiveness when updating the metric. That is, we balance 1) keeping the metric from changing too much from update to update, with 2) updating the metric to satisfy the constraint. This tradeoff is controlled by the learning rate η, and each update to the metric solves an optimization problem that encodes this balance appropriately.

More specifically, we aim to learn a Mahalanobis distance function, which is parameterized by a positive semi-definite matrix W, and is given by $d_W(v, w) = (v - w)^T W (v - w)$. Learning the distance function corresponds to learning the matrix W. Note that since W is positive semi-definite, $W = G^T G$ for some matrix G, and it is straightforward to show that the Mahalanobis distance function d_W is simply the squared Euclidean distance after applying the transformation G to the data points. When updating W to W_{next}, we measure our conservativeness using the LogDet divergence,

$$D_{\ell d}(W_{next}, W) = \text{tr}(W_{next} W^{-1}) - \log\det(W_{next} W^{-1}) - n,$$

where tr refers to the matrix trace and n is the number of rows or columns of W. This divergence measure is natural since positive semi-definiteness of W is automatically maintained, and it has several properties such as scale-invariance which are desirable for metric learning problems. Further, the LogDet divergence has been used extensively in the context of metric learning (e.g., [4, 6]). For correctiveness, we attempt to enforce the constraint $d_W(v, w) \leq d_W(v, x) - 1$, or equivalently, $d_W(v, w) - d_W(v, x) + 1 \leq 0$, as is standard for relative-distance metric learning algorithms [19]. This constraint ensures that the distance between v and w should be much smaller than the distance between v and x. Given these two components, we attempt to find the updated distance parameterized by W_{next} that minimizes the sum of the LogDet divergence between W_{next} and W (conservativeness) plus the error of W_{next} not satisfying the current constraint using the hinge loss (correctiveness), where the sum is balanced by the learning rate η. In particular, we look for a matrix W_{next} that minimizes

$$D_{\ell d}(W_{next}, W) + \eta \cdot \ell(d_{W_{next}}, v, w, x), \quad (1)$$

where $\ell(d_W, v, w, x) = \max((d_W(v, w) - d_W(v, x) + 1, 0)$ is

the *hinge loss* for the constraint $d_W(v,w) \leq d_W(v,x) - 1$. The solution of the minimization problem to compute W_{next} can be computed in closed-form in a manner similar to the online metric learning algorithm of [6]. In particular, a pleasant and surprising aspect of the update for our algorithm is that the solution to W_{next} can be computed as a rank-two update to the matrix W; this can be shown by taking the gradient of (1), setting it to zero, and solving for W_{next}. Details of the update can be found in our publicly available MATLAB code, which show how to handle the gradient at the "hinge" location.[1]

One key advantage of the above online algorithm is that one can prove online regret bounds for this algorithm with appropriate learning rate selection that guarantee that the metric produced by the online algorithm performs similarly to the output of the best possible offline metric learning algorithm (i.e., an algorithm that performs updates of the metric in a batch mode using all constraints). Briefly, one defines the total loss of an online algorithm as the sum of the losses over all T timesteps/constraints. Denote the sequence of W matrices constructed by the online algorithm as $W_1, ..., W_T$, and similarly denote the sequence of v, w, and x vectors from each constraint as $v_1, .., v_T$, $w_1, .., w_T$, and $x_1, ..., x_T$. Then we can define the total loss as

$$\sum_{t=1}^{T} \ell(W_t, v_t, w_t, x_t).$$

Analyses of online learning algorithms focus on the *regret*, which is the difference between the total loss of the online learning algorithm with the total loss of the best possible offline algorithm:

$$Reg = \sum_{t=1}^{T} \ell(W_t, v_t, w_t, x_t) - \text{argmin}_{W_*} \sum_{t=1}^{T} \ell(W_*, v_t, w_t, x_t).$$

The goal is to bound the regret as a function of T, the total number of constraints processed. Our approach, which combines the hinge loss with a convex regularizer, can be viewed as a special case of the online learning framework discussed in Shalev-Shwartz and Singer [13] (see Section 6, equation 38). In particular, with the appropriate selection of learning rates as discussed in Shalev-Shwartz and Singer, we can achieve regret that is bounded by $O(\sqrt{T})$. Finally, note that, while the proposed algorithm shares similarities to existing methods (c.f., [6, 9]) and has been studied theoretically in the context of a large class of online learning methods, we are not aware of metric learning work based on LogDet conservativeness and the standard hinge loss over relative distance constraints.

4. EMPIRICAL VALIDATION

This section introduces a set of six experiments showcasing the benefits of combining HOLLER with Fitted R-MAX.

4.1 2D Mountain Car Domain

This section introduces our experimental domain, a generalized version of the well-studied mountain car task [14]. Mountain car is particularly appropriate for this work as it is a simple domain with continuous state space and can be easily parameterized to highlight the strengths of HOLLER.

In mountain car, the agent must generalize across continuous state variables in order to drive an underpowered car up a mountain to a goal state. To make the problem more challenging than the original formulation, the agent begins at rest at the bottom of the hill.[2] The reward for each time step is -1. The episode ends, and the agent is reset to the start state, after 500 time steps or if it reaches the goal state.

In practice, one of the most difficult challenges for the agent is to find the goal state the first time. After the goal state has been seen at least once, RL algorithms are typically able to quickly learn to consistently find the goal (albeit with different numbers of steps, which determines reward). Effective exploration and generalization is thus critical for agents to quickly find high-performing policies.

In the standard two dimensional mountain car task, two continuous variables fully describe the agent's state. The horizontal position (x) and velocity ($\dot{x}$) are restricted to the ranges $[-1.2, 0.6]$ and $[-0.07, 0.07]$ respectively. The state variables are automatically scaled (linearly) to $[-1, 1]$, as consistent with past work in this domain [7, 14, 18]. If the agent reaches $x = -1.2$, ($\dot{x}$) is set to zero, simulating an inelastic collision. On every time step the agent selects from three actions, {`Left`, `Neutral`, `Right`}, which change the velocity by -0.001, 0, and 0.001, respectively. Additionally, gravity is simulated by adding $-0.025(\cos(3x))$ to $\dot{x}$, which depends on the local slope of the mountain. The goal states are those where $x \geq 0.5$. Our implementation mimics the publicly available version of this task.[3]

4.2 Experimental Procedure

In order to learn in the 2D Mountain Car Domain, we first tune the Fitted R-MAX learning parameters on the standard 2D task without metric learning, and then tune the HOLLER learning parameters on the standard 2D task. The primary consequence of this approach is that the Fitted R-MAX parameters have not been tuned to take advantage of the state variables after metric learning: results we present are therefore biased against HOLLER. Additionally, neither the Fitted R-MAX nor HOLLER parameters are tuned for the variants of the 2D mountain car problem, enabling a fair comparison on the more complex task variants (discussed in Section 4.3).

1: The Standard 2D Mountain Car task is run where agents use Fitted R-MAX with a variety of parameters. The parameters tuned were *minFraction*, which determines if the agent is allowed to end its nearest neighbor approximation early, *modelBreadth*, which sets how fine a uniform grid is used to generalize the state space, and *resolutionFactor*, which determines the size of the regularly spaced grid used to approximate saved instances. We found that values of *minFraction* = 0.01, *modelBreadth* = 0.03, and *resolutionFactor* = 5 produced high-valued policies with few samples and allowed for very fast experiments (in terms of wall clock time). These parameter settings are similar to those used in past experiments in this domain and are explained in detail elsewhere [7, 17].

In order for HOLLER to learn a distance metric, it must have data recorded from the task. To record this data, we allowed the agent to explore the task (with a fully random policy) for different numbers of episodes. The more episodes used for learning the metric, the more likely it will be accurate. However, the episodes spent collecting data will count against the agent's performance (as discussed further in Section 4.3). After trying 6 different values, we decided to experiment with 1, 5, and 10 episodes of data for HOLLER, affecting Algorithm 1, lines 33 and 34.

2: Given the data collected, HOLLER is then used to learn a distance metric. We experimented with 10 values of η (a parameter for Algorithm 1) from 0.0001–0.5 and found that 0.01 and 0.05 produced the best behavior on the 2D Mountain Car task for 1, 5, and 10 episodes. The performance of 0.01 and 0.05 were not dis-

[1] See `cs.lafayette.edu/~taylorm/MetricLearn`

[2] The mountain car task is typically deterministic: to introduce randomness among trials, the initial position of the car in each trial's start state is perturbed by a random number in $[-0.005, 0.005]$, as was done previously in this domain [17].

[3] See `http://library.rl-community.org/wiki/Mountain_Car_(Java)`

tinguishable, suggesting that HOLLER's performance is not overly dependent on this parameter. Experiments in the following sections use η = 0.05. We also tested four values of *NumPts*, the parameter that determines how many temporally similar states to compare, and found that a value of 10 produced slightly better results than 1, 5, or 20.

3: Although HOLLER is designed to be an on-line algorithm, it can be run multiple times over the same constraints if the data is not immediately discarded (Algorithm 1, line 32). In our experiments we tried iterating over the collected data for 1, 2, 3, 5, and 10 times. For 1, 5, and 10 episodes, iterating over the data twice produced slightly better results than the other parameters, but the differences between the final performance (as measured in the following sections) were small. In our experiments, we run iterate over the collected data twice.

4: Having determined all the necessary parameters, HOLLER can be used to learn a distance metric. Initially we learned a single distance metric per action. However, in the Mountain Car domain, the action outcomes are similar enough that the learned distance metrics for the different actions were indistinguishable. Therefore, the experiments below focus on learning a single distance metric, $W_{Neutral}$ (using only instances where the agent randomly executed the `Neutral` action) and using that metric for all W_a when learning an action-value function.

5: To evaluate HOLLER, we then learn the 2D Mountain Car task using Fitted R-MAX, with and without the learned distance metrics. The effect of the distance metric is compared in the following sections by evaluating the final and total rewards using both the Euclidean distance and using the learned W_a.

4.3 2D Mountain Car Results

First, consider the distance metric, W, learned by HOLLER from 10 episodes worth of data. Examining the 10 trials, we find that

$$W = \begin{bmatrix} 0.119 \pm 0.012 & -0.006 \pm 0.003 \\ -0.006 \pm 0.003 & 0.096 \pm 0.008 \end{bmatrix},$$

where the $\pm$ terms show the standard error. The values on the diagonal show that x, the first state variable, is slightly more important than $\dot{x}$, the second state variable. The off-diagonal values are very small, showing that linear combinations of the two state variables are not critical in this domain. However, it is impossible to say whether this distance metric is "correct" – instead, the utility of this metric is in the observed performance of the RL agent.

Figure 1(a) shows learning curves for learning the 2D Mountain Car task with Fitted R-MAX, both with (for 1, 5, or 10 episodes of data) and without (No Metric Learning) HOLLER. The x-axis shows the episode and the y-axis shows the average reward for that episode number. Error bars show the standard error over 10 independent trials. All experiments are averaged over 10 trials and all experiments in this section are ended after 100 episodes. The three trials that use HOLLER after collecting data for 1, 5, and 10 episodes learn to reach the goal very quickly, quickly outperforming learning with the no distance metric. However, this analysis does not account for the number of episodes spent collecting data (Algorithm 1, lines 5–11).

Figure 1(b) explicitly shows the time spent collecting data for HOLLER; for instance, when collecting data for 10 episodes, the learning curve begins on episode 10, as episodes 0-9 are assumed to have reward -500. To make the graph more readable, a 5-episode sliding window is used and error bars are not shown. Additionally, the performance of Sarsa (a popular model-free learning algorithm) with CMAC function approximation is compared by using the same parameters as those in the literature [7, 14, 17]), showing that Sarsa agents take longer to discover the goal state, but that eventually achieve a slightly higher reward.

One reasonable dimension along which to evaluate the effectiveness of HOLLER would be the average reward at a set amount of data (e.g., after 100 episodes). However, such a metric ignores the "speed" of learning — Sarsa has a higher performance at episode 100 but suffers from a slow start. Analyzing the cumulative rewards also shows that using Fitted R-MAX with HOLLER learning from 1 episode of data outperforms the other learning methods.

In the standard 2D Mountain Car problem, HOLLER with 1, 5, and 10 episodes of data outperforms Fitted R-MAX without HOLLER in terms of the final average reward and the cumulative reward. Additionally, the difference in cumulative rewards is statistically significant. While Sarsa outperforms Fitted R-MAX on this test both in terms of final and cumulative reward, previous work has shown that it is difficult for Sarsa to scale to higher-dimensional versions of this problem [18]. Experiments showing the superiority of Fitted R-MAX are replicated later in Section 4.4. A summary of this and other experiments can be found in Table 1.

4.3.1 Variant 1: Inflated State Variable

As a second task, we consider the more general case where the range of the second state variable is not known. The state variable $\dot{x}$ still ranges from [-0.007, 0.007], but we assume that in order to ensure that all data is scaled so that all state variable ranges are within the expected range of [-1, 1], $\dot{x}$ is divided by 0.7 (rather than 0.007), causing the observed range to become [-0.01, 0.01]. Such non-optimal scaling could occur if the human designer did not know the true variable range and was being careful. Alternatively, the range could be automatically determined through sampling the minimum and maximum values, but two noisy readings (one high and one low) could throw off the scaling. As seen in the previous subsection, the x and $\dot{x}$ state variables are both important for accurately predicting the transition function and we would expect that Fitted R-MAX, using parameters set for the standard 2D mountain car task, will not perform as well as when it is coupled with a learned distance metric.

As shown in Figure 2(a), the episodes spent learning the distance metric initially hurt the learners: Fitted R-MAX without a distance metric initially outperform an agent that collected 10 episodes of data for HOLLER. However, the final average reward and average cumulative reward is better for all three settings of the HOLLER agents, although the differences are only statistically significant about half of the time (see Table 1 for Student's t-test results).

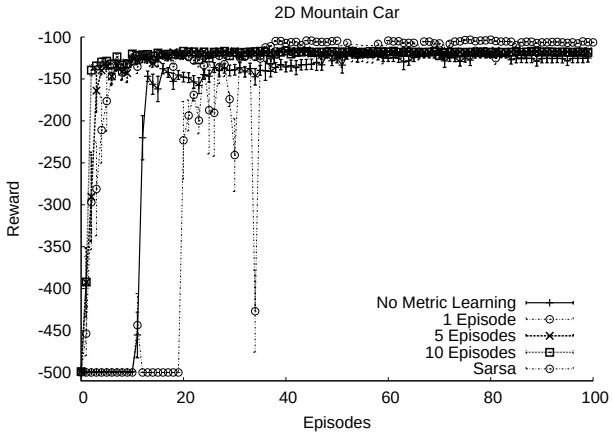
4.3.2 Variant 2: Sensor and Actuator Noise

To test the efficacy of HOLLER in the presence of noise, we next consider a variant of mountain car that includes partial observability and stochasticity. As before, the position and velocity state variables are scaled to the range $[-1, 1]$ and then Gaussian noise is added to the agent's observation, drawn randomly on each time step from $\mathcal{N}(0, 0.1)$. Similarly, on every time step, the agent's velocity is multiplied by zero-mean noise drawn from $\mathcal{N}(0, 0.01)$.

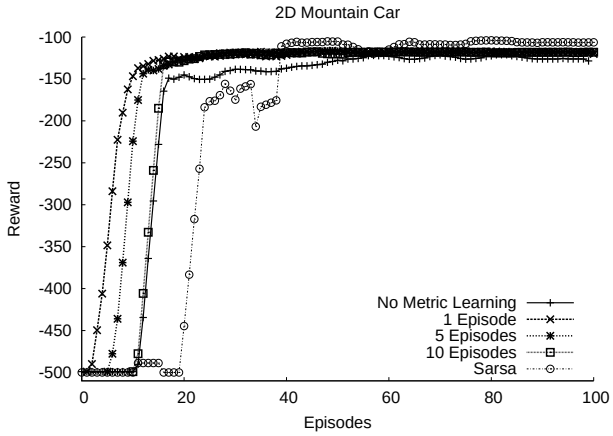
Figure 2(b) shows that although the noise makes learning more difficult for all learners (i.e., their reward is lower than agents in Figure 1(b)), HOLLER is able to learn distance metric functions that allow the agents to outperform the default scaling. This is a particularly important test as it shows that HOLLER is robust to noise, as desired. Using HOLLER produces a higher final and cumulative reward in all three cases, although only the differences between the cumulative rewards are statistically significant.

4.3.3 Variant 3: Irregular Action Function

Next, consider the situation where the transition function is highly



(a) The episodes used by HOLLER are ignored, with standard error



(b) 2D Normal

Figure 1: These figures show the same learning curve data where the x-axis is the episode number and the y-axis shows the reward. In (a), the y-axis shows the average reward on a given episode (higher is better) with the standard error. (b) also shows the average reward per episode, but accounts for the episodes spent learning the distance metric and uses a 5-episode sliding window.

dependent on the state, as was done in the 2009 Reinforcement Learning Competition (c.f., `http://2009.rl-competition.org/` and [21]). In particular, the actions 0–2 (`Left`, `Neutral`, and `Right`) were mapped such that the action executed by the agent depended on $\dot{x}$ and a (the action selected by the agent). The action executed by the car in the simulator was

$$\left(a + \left(\frac{\dot{x} + 0.07}{0.14} \cdot 99.0\right)\right) \mod 3.$$

As expected, Figure 3(a) shows that learning a metric significantly improves learning, both in terms of the final reward and cumulative reward, as the learned metric can automatically increase the resolution to $\dot{x}$, allowing it to better approximate a transition function significantly more complex than for the standard 2D mountain car.

4.3.4 Variant 4: A Third, Irrelevant, State Variable

As a final variant for the 2D Mountain Car task, we consider adding an additional irrelevant state variable. Although the transition and reward functions still depend only on x and $\dot{x}$, the agent is provided a random number as a third feature on every time step. This state variable is drawn uniformly in [-0.025, 0.025]. As Figure 3(b) shows, this additional state variable significantly degrades the performance of Fitted R-MAX with a Euclidean distance metric as it must now generalize its data over an extra dimension (i.e., it suffers from the"curse of dimensionality"). However, HOLLER allows this third state variable to be de-valued, allowing the agents learn almost as well as in the standard 2D mountain car task.

HOLLER is not dependent on the number of state variables: although Fitted R-MAX can generally not scale to high-dimensional spaces, using HOLLER would allow an experimenter to eliminate irrelevant state variables, potentially enabling this and other methods to scale to much higher dimensional spaces.

4.4 4D Mountain Car

The 4D Mountain Car task extends the 2D task so that there are four state variables $(x, \dot{x}, y, \dot{y})$ and the agent selects from five actions (`Neutral`, `West`, `East`, `South`, `North`) [18]. The transition function is similar to the 2D case, but now takes into account the extra dimensions. Likewise, the goal region is now $x \geq 0.5$ and $y \geq 0.5$. Our task implementation is based on a publicly available implementation.[4] This task is much more difficult than the 2D task because of the increased state space size and additional actions. After initial experimentation without distance metric learning, we set the parameters of Fitted R-MAX to be similar to past work [17] *minFraction* = 0.3, *modelBreadth* = 0.3, *resolutionFactor* = 3, and agents train for a total of 250 episodes.

As shown in Figure 3(c), the final and cumulative performance of learners using HOLLER is higher than those that rely on the Euclidean distance metric. Also, note that Sarsa, using the same parameters set in the literature [18], does much wore than Fitted R-MAX, due to the high-dimensional space. Sarsa agents do not consistently find the goal state until after 2,000 episodes, requiring roughly two orders or magnitude more data than the instance-based learning method (with or without metric learning).

Taken as a whole, and summarized in Table 1, these experiments show that HOLLER can successfully improve learning performance on a variety of tasks, both in terms of final and cumulative reward.

5. RELATED WORK

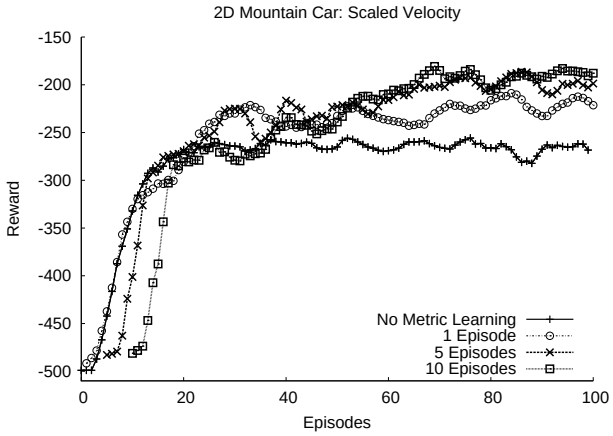
The most similar distance metric learning work has been discussed earlier in Sections 2.3 and 3.3. This section focuses on the most relevant existing reinforcement learning algorithms.

Graph-based approaches to learning state representations, such as using *proto-value functions* [10], typically focus on using a known connectivity graph (e.g., a transition function) to learn a (near-) optimal set of features. By using the eigenvectors of the connectivity graph's Laplacian, very accurate representations of an MDP's value function can be learned. However, proto-value function work does not typically consider the sample complexity of learning such a connectivity graph — our work is directly concerned with minimizing the amount of environmental samples needed to learn a state representation and thus attempt to maximize the on-line reward.

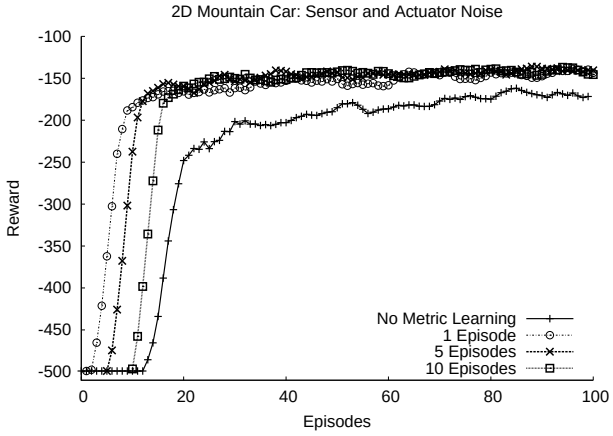
The Bellman Error Basis Functions (BEBF) [12] method relies on iteratively adding basis functions, where each basis function is constructed to improve the Bellman error over the previous set of basis functions. BEBF differs from the current work primarily in its aim — while the BEBF work examines relatively simple RL tasks with the goal of constructing very accurate value functions from hundreds of thousands of samples, HOLLER instead aims to construct a distance metric with relatively little data that can be used to both guide exploration and improve value function estimation.

In a supervised learning setting, unlike in RL, training sets provide the correct target label, enabling a more straightforward appli-

[4] `http://library.rl-community.org/wiki/Mountain_Car_3D_(CPP)`



(a) 2D, Scaled Velocity



(b) 2D, Sensor and Actuator Noise

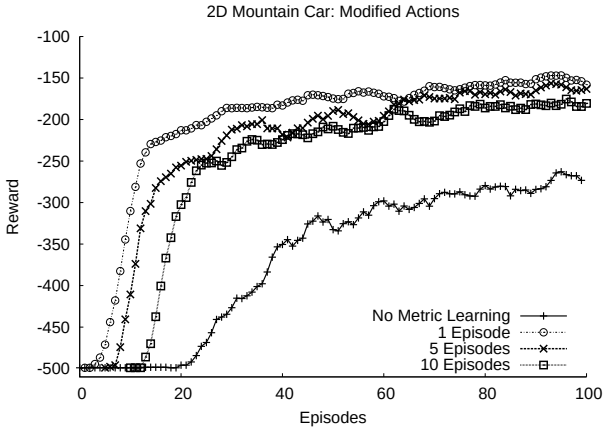
Figure 2: A learned distance metric improves both the total and final reward when the velocity state variable is incorrectly scaled (a) and when there is noise in both the sensors and actuators (b).

cation of distance metric learning. For instance, *Metric Learning for Kernel Regression* [20] (MLKR) is a metric learning method designed for regression problems.

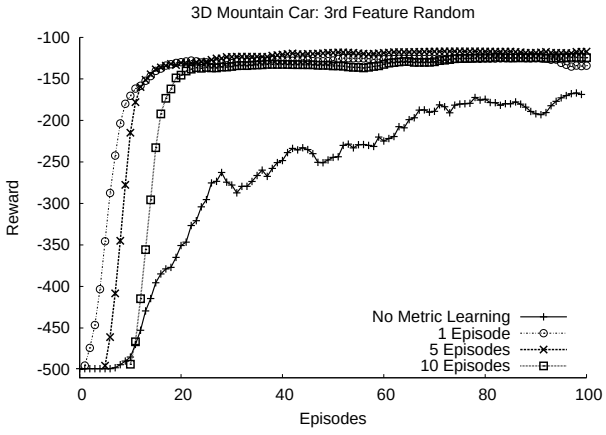
Three recent papers presented at ECML-10 also tackle similar problems. Nouri and Littman [11] build upon MLKR to create the *Dimension Reduction in Exploration* algorithm. The algorithm constructs a set of "factorized" MLKR problems (F-MLKR), under the assumption that individual state features for resulting states are independent of each other, where one MLKR problem is constructed per state feature, per action, for a total of $||A|| \times ||S||$ F-MLKR regressors. F-MLKR agents must also be provided the reward function, unlike in HOLLER, where the reward is learned. Additionally, agents that use HOLLER benefit from dimensionally reduction as well as proper scaling of state variables, and can be combined with existing RL methods.

The second recent paper, Jung and Stone [8], trains multiple Gaussian processes in batches to approximate the transition function. The GP-RMAX algorithm requires a deterministic transition function, must be provided the reward function. In contrast to both F-MLKR and GP-RMAX, HOLLER learns a distance function for the entire state space based on few samples, which means that HOLLER can quickly generalize over the entire state space.

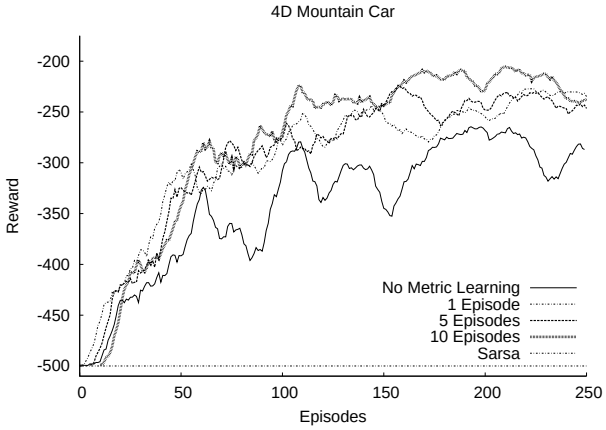
The third paper [2] presents an actor-critic method to determine where to place basis functions and what parameterization they should



(a) 2D, Custom Action Mapping



(b) 3D, Irrelevant State Variable



(c) 4D Mountain Car: Performance

Figure 3: Figures (a) and (b) show how HOLLER produces better learning in task with a custom action mapping and with an irrelevant state variable, respectively. In (c), learning curves are averaged over ten trials with a 10-episodes sliding window.

have, rather than learning a single metric that is useful across the state space (independent of the function approximator parameterization). Additionally, we note that the authors test their algorithm on an easier version of mountain car (where the agent starts at a random state rather than the bottom of the hill, making exploration sig-

Domain	Algorithm	Final Ave. Reward	Stat. Sig.	Cumulative Reward	Stat. Sig.
2D: Standard	Fitted R-MAX	-126		-17600	
	HOLLER-1	-118		-13620	✓
	HOLLER-5	-118		-14783	✓
	HOLLER-10	-117		-16440	✓
	Sarsa	-106	✓	-19755	(✓)
2D: Scaled	Fitted R-MAX	-268		-28050	
	HOLLER-1	-227		-25380	
	HOLLER-5	-199	✓	–24740	✓
	HOLLER-10	-199	✓	–26000	
2D: Noisy	Fitted R-MAX	-157		-23600	
	HOLLER-1	-136		-16840	✓
	HOLLER-5	-141		-17240	✓
	HOLLER-10	–150		-18733	✓
2D: Convoluted Actions	Fitted R-MAX	-260		-36190	
	HOLLER-1	-154	✓	-19990	✓
	HOLLER-5	-161	✓	–22660	✓
	HOLLER-10	-177	✓	-25460	✓
3D: Irrelevant Feature	Fitted R-MAX	-164		-26360	
	HOLLER-1	-128	✓	–14500	✓
	HOLLER-5	-117	✓	-14840	✓
	HOLLER-10	-124	✓	-17630	✓
4D: Standard	Fitted R-MAX	-291		-36190	
	HOLLER-1	-225		-19990	✓
	HOLLER-5	-239		-22663	✓
	HOLLER-10	-241		-25460	✓
	Sarsa	-500	(✓)	-50000	(✓)

Table 1: This table summarizes all experiments, averaging over ten independent trials. The third column shows the average reward at the end of the trial (250 episodes for the 4D task, 100 episodes for all others). The fourth column has a check if the difference in the final reward is statistically significantly different from learning with Fitted R-MAX without a learned distance metric, as determined by $p < 0.05$ on Student's t-test results. The fifth and sixth columns report the average cumulative reward and whether the difference in the cumulative rewards and Fitted R-MAX are statistically significant.

nificantly easier), but their algorithm takes thousands of episodes to converge.

6. CONCLUSION AND FUTURE WORK

This paper has introduced HOLLER and shown how it can be combined with an off-the-shelf instance based RL algorithm. Empirically, this novel distance metric learning algorithm significantly improves learning efficacy in a number of different tasks, including noise and irrelevant state variables. One of the key benefits of HOLLER is that very little data is required to learn an appropriate state representation and thus the on-line reward can be significantly improved relative to learning with a Euclidean distance metric.

In the future, we intend to try to fully integrate learning W and a control policy simultaneously. While such an integration would not be critical in domains where the distance metric can be quickly learned, it may prove useful in more complex and higher-dimensional tasks. We also are interested in attempting to further improving the efficacy of HOLLER by trying establish appropriate decay rates for η (rather than using a fixed learning rate), combining the updates from multiple actions (rather than learning each W_a in isolation), and trying to tune exploration to learn W as quickly as possible (rather than relying on random exploration). Lastly, while this paper has focused on Fitted R-MAX, we expect that HOLLER would be beneficial to other instance-based RL methods, as well as model-free methods. For instance, future work could examine how W could be used by Sarsa to help select, or parameterize, its function approximator so that the value function can better match the underlying topology of the state space without relying on human intuition or simple estimates of state variable ranges. Lastly, it would be interesting to empirically compare our Mahalanobis distance approach, with the LogDet loss function, to alternative approaches.

Acknowledgements

The authors would like to the anonymous reviewers and Tobias Jung for useful comments and suggestions.

7. REFERENCES

[1] J. S. Albus. *Brains, Behavior, and Robotics*. Byte Books, Peterborough, NH, 1981.
[2] D. D. Castro and S. Mannor. Adaptive bases for reinforcement learning. In *ECML*, 2010.
[3] J. Davis and I. Dhillon. Structured metric learning for high-dimensional problems. In *KDD*, 2008.
[4] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.
[5] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS*, 2005.
[6] P. Jain, B. Kulis, I. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *NIPS*, 2008.
[7] N. K. Jong and P. Stone. Model-based Function Approximation for Reinforcement Learning. In *AAMAS*, 2007.
[8] T. Jung and P. Stone. Gaussian processes for sample efficient reinforcement learning with RMAX-like exploration. In *ECML*, 2010.
[9] B. Kulis and P. Bartlett. Implicit online learning. In *ICML*, 2010.
[10] S. Mahadevan and M. Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.
[11] A. Nouri and M. L. Littman. Dimension reduction and its application to model-based exploration in continuous spaces. In *ECML PKDD*, 2010.
[12] R. Parr, C. Painter-Wakefield, L. Li, and M. L. Littman. Analyzing feature generation for value-function approximation. In *ICML*, 2007.
[13] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning Journal*, 2(69):115–142, 2007.
[14] S. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158, 1996.
[15] M. Slaney, K. Weinberger, and W. White. Learning a metric for music similarity. In *ISMIR*, 2008.
[16] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.
[17] M. E. Taylor, N. K. Jong, and P. Stone. Transferring instances for model-based reinforcement learning. In *ECML PKDD*, 2008.
[18] M. E. Taylor, G. Kuhlmann, and P. Stone. Autonomous transfer for reinforcement learning. In *AAMAS*, 2008.
[19] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006.
[20] K. Q. Weinberger and G. Tesauro. Metric learning for kernel regression. In *AI-STATS*, 2007.
[21] S. Whiteson, B. Tanner, and A. White. The reinforcement learning competitions. *AI Magazine*, 31(2):81–94, 2010.
[22] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2002.
[23] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

Energy Applications

Cooperatives of Distributed Energy Resources for Efficient Virtual Power Plants

Georgios Chalkiadakis[1] Valentin Robu[2] Ramachandra Kota[3] Alex Rogers[2]
Nicholas R. Jennings[2]
{gc2,vr2,rck05r,acr,nrj}@ecs.soton.ac.uk
[1] Dept. of Electronic and Computer Engineering, Technical University of Crete, Greece
[2] School of Electronics and Computer Science, University of Southampton, UK
[3] Secure Meters Ltd., Winchester, UK

ABSTRACT

The creation of Virtual Power Plants (VPPs) has been suggested in recent years as the means for achieving the cost-efficient integration of the many distributed energy resources (DERs) that are starting to emerge in the electricity network. In this work, we contribute to the development of VPPs by offering a game-theoretic perspective to the problem. Specifically, we design *cooperatives* (or "cooperative VPPs"—CVPPs) of rational autonomous DER-agents representing small-to-medium size renewable electricity producers, which coalesce to profitably sell their energy to the electricity grid. By so doing, we help to counter the fact that individual DERs are often excluded from the wholesale energy market due to their perceived inefficiency and unreliability. We discuss the issues surrounding the emergence of such cooperatives, and propose a pricing mechanism with certain desirable properties. Specifically, our mechanism guarantees that CVPPs have the incentive to truthfully report to the grid accurate estimates of their electricity production, and that larger rather than smaller CVPPs form; this promotes CVPP efficiency and reliability. In addition, we propose a scheme to allocate payments within the cooperative, and show that, given this scheme and the pricing mechanism, the allocation is in the core and, as such, no subset of members has a financial incentive to break away from the CVPP. Moreover, we develop an analytical tool for quantifying the uncertainty about DER production estimates, and distinguishing among different types of errors regarding such estimates. We then utilize this tool to devise protocols to manage CVPP membership. Finally, we demonstrate these ideas through a simulation that uses real-world data.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Economics, Experimentation

Keywords

energy and emissions, incentives for cooperation, coalition formation, simulation

Cite as: Cooperatives of Distributed Energy Resources for Efficient Virtual Power Plants, G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, N. R. Jennings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 787-794.

1. INTRODUCTION

The vision of a *"Smart Grid"* [12], and the resulting creation of a robust, intelligent electricity supply network which makes efficient use of energy resources and reduces carbon emissions, is a challenge that has been recently taken up by a growing number of researchers [6, 3, 7, 14, 15]. In this context, one of the main problems facing the energy supply industry is how to best achieve the utilization of the *distributed energy resources (DERs)* that, in recent years, have appeared in the electricity network. Such DERs range from electricity storage devices to small and medium capacity (2kW-2MW) renewable energy generators.

In principle, employing DERs to produce energy could reduce reliance on conventional power plants even by half [10]. Unlike conventional power plants that lie on the transmission network and are "dispatched" (i.e., called in to produce energy when needed) by the national electricity transmission network operators (termed *the Grid* herein), DERs lie in the distribution network and, due to their small size, they (and their capacity) are "invisible" to the Grid. Thus, they cannot be easily dispatched to meet demand. Moreover, due to their decentralized nature and small size, DERs are either invisible to the electricity market as well, or, lack the capacity, flexibility or controllability to participate in a cost-efficient way [10].

Now, the *reliability of supply* is a major concern of the Grid. It is essential that independent suppliers are reliable, since the failure to meet production targets could seriously compromise the smooth operation of the network as a whole. In contrast, given the unpredictability of renewable energy sources, the DERs would usually struggle to meet power generation targets when operating alone. This normally prohibits them from striking profitable deals with the Grid, and keeps them out of the electricity market for fear of suffering penalties specified in contracts (driving them to sign low-profit contracts with third-party market participants instead) [10]. To address this issue, in recent years many countries (e.g., in the EU) have enacted policies that guarantee the sale of electricity from small-scale producers to the Grid in pre-determined *feed-in tariffs* that are generally above market prices. Such policies were conceived by the need to promote the incorporation of renewable energy sources into the Grid, so that they generate appreciable percentages of total demand. However, with the number of DERs expected to rise to hundreds of thousands, and with the variable generation seen as another uncertainty to be addressed in real time through active Grid management, this is clearly unsustainable.

To counter these problems, the creation of *Virtual Power Plants (VPPs)* to aggregate DERs into the virtual equivalent of a large power station, and thus enable them to cost-efficiently integrate into

the market, has been proposed in recent years. [7, 10]. A VPP is a broad term that intuitively represents the aggregated capabilities of a set of DERs. For example, it can be thought of as a portfolio of DERs, as an independent entity or agent that coordinates DERs pooling their resources together, or as an external aggregator that "hires" DERs to profit from their exploitation.

In our work here, we propose that power-producing DERs coalesce together to form *cooperatives* of agents that can profitably be integrated into the Grid, such a cooperative corresponding to a virtual power plant. Viewing the DERs as autonomous agents is natural, due to their distributed nature and inherent individual rationality, and enables them to realize their full potential as self-interested market units (as it allows for the possibility that even the smallest of DERs can carry out certain communication and intelligent decision making tasks on their own, but without imposing this as a requirement). We call these coalitions of DER-agents "cooperatives" because of *(a)* their completely decentralized nature; *(b)* their ability to sell their production without relying on any external entity that profits by using their members' resources; and *(c)* the fact that members willingly participate in a coalition, as it is in their best interests to do so. Of course, the mechanisms described in this work can also be readily used by any company that wishes to attract DERs as suppliers, aiming to resell their energy to the Grid. In the rest of the paper, we will use the terms "cooperative" and "cooperative VPP (CVPP)" interchangeably.

Given the issues discussed above, it is only natural that the Grid should encourage the emergence of cooperatives, by guaranteeing the purchase of CVPP energy at competitive rates. To this end, in this paper we incorporate ideas from mechanism design and cooperative game theory, and put forward an energy pricing mechanism to be employed by the Grid. The mechanism can be seen as an efficient alternative to feed-in tariffs, and so promotes the incorporation of the DERs (as CVPPs) in the Grid. In some detail, our mechanism promotes supply reliability, guaranteeing that CVPPs truthfully provide the Grid with estimates of their electricity production that are as accurate as possible. Further, they are rewarded for increased production, while the Grid maintains the ability to decide the flexibility of the mechanism and its degree of independence from market fluctuations. Building on that key contribution, we then propose a payment scheme to allocate payments within the cooperative, and show that, given this scheme and the pricing mechanism, a CVPP can guarantee payments to its members such that no subset of them has a financial incentive to form a CVPP of its own. Formally, we guarantee that, provided DER production estimates are accurate, the payments to CVPP members lie in the set of *core* allocations of the corresponding coalitional game [9]. We then develop a method that quantifies the uncertainty regarding DER production estimates and distinguishes between different types of errors in predicted production (i.e., those specific to individual DERs, and those common within whole DER clusters), and employ it to devise CVPP membership management protocols.

This is the first paper to discuss the formation of VPPs from a game-theoretic standpoint, extrapolating as it does mechanism design and cooperative game theory concepts and techniques to this domain. As such, this work demonstrates that multiagent research can provide the energy industry with solutions regarding the successful integration of DERs in the supply network. Note that this research has the potential of short to mid-term applicability in realistic settings, as several power trading companies that buy electricity from small scale producers to sell to the Grid already exist. Examples include *Flexitricity* in the UK and *Tata Power Trading Company Ltd.* in India (business description available online).

2. RELATED WORK

Here we briefly review existing related work that provides intelligent agents—and, more generally, AI research—solutions to energy-related problems. To begin, we note that researchers in the community have recently presented economics-inspired work to tackle such problems. Specifically, Vytelingum *et al.* [15] proposed strategies for the management of distributed micro-storage energy devices that adapt to the electricity market conditions. In separate work, they developed a market-based mechanism to automatically manage the congestion within the system by pricing the flow of electricity, and proposed strategies for traders in the Smart Grid [14].

However, ideas from *cooperative game theory* in particular—i.e., from the branch of game theory that studies the problem of forming *coalitions* of cooperating agents—have been used in the broader energy domain for more than a decade. Yeung *et al.* [16] employ coalitional game theory in a multiagent system model of the trading process between market entities that generate, transmit and distribute power. Also, Contreras *et al.* [2, 1] presented a *bilateral Shapley value* negotiation scheme to determine how to share the costs for the expansion of power transmission networks among coalescing rational agents.

Turning our attention to VPP-specific literature, Pudjianto *et al.* [10] stress the need to integrate DERs into the electricity network in an organized and controllable manner through participation in a VPP structure, and discuss the subsequent technical and commercial benefits to the electricity network as a whole. They also clearly outline the economic advantages to DERs, demonstrating as they do through specific examples that VPPs can be used to facilitate DER access to the electricity market. Dimeas and Hatziargyriou [6] also call for the emergence of VPPs, and essentially suggest an organizational structure that makes use of interacting coalitions to this purpose. Similarly, Mihailescu *et al.* [8] propose the use of coalition formation to build VPPs, but do not provide the details of the formation process or offer specific game-theoretic solutions—as they do not discuss issues of individual rationality or incentive compatibility. Though all of those papers advocate the creation of VPPs, they do not describe specific mechanisms for the market-VPP interface or the interactions among VPP members.

In contrast, the *PowerMatcher* (see [7] for an overview) is a decentralized system architecture that has been proposed as a means to balance demand and supply in clusters of DERs. It attempts to implement optimal supply and demand matching by organizing the DER-agents into a logical tree, assigning them roles and prescribing strategies to use in their interactions. The aspect of this system most relevant to us is the one proposing the aggregation of individual agents' supply offers in a cluster, serving as a VPP through the use of an *objective agent*. Such an agent has the task of implementing a "business logic" that would guide the VPP's actions. However, the authors stop short of proposing a specific business logic. Our approach can be seen as a detailed description of just such a logic, employing game-theoretic ideas and tools to this purpose.

3. AGENT COOPERATIVES

An agent cooperative (CVPP) is a collection of participating DER agents, each of which registers with the CVPP when joining. The CVPP may possess and employ any rules, tools and functionality necessary to ensure its unconstrained and profitable operation as an enterprise. We now present briefly some key CVPP characteristics and functionality most relevant to our work here.

In most countries, the day is divided into 48 half-hour electricity trading intervals, or *settlement periods*. For each of these, electricity prices are set in the market, and specific electricity production targets are specified for the various generators the day before, given

predicted supply and demand. A DER i can estimate an *expected production* value $\widetilde{\mathrm{prod}}_{i,t_j}$ for any half-hour period t_j. This is the energy it expects to be able to supply during t_j, given any known external factors (such as the prevailing meteorological conditions) and its expected technical state. Thus, the main profile parameter that describes the production of a DER i throughout each day is its *expected production vector* $\widetilde{\boldsymbol{prod}}_i = \langle \widetilde{\mathrm{prod}}_{i,t_j} \rangle$, describing the DER's production for every half-hour period.

Note that, besides this estimated production, there is an *actual production vector* associated with each DER i: $\boldsymbol{prod}_i = \langle \mathrm{prod}_{i,t_j} \rangle$. The value for each prod_{i,t_j}, however, becomes known only after the corresponding period elapses. We will be using the simplified notation prod_i and $\widetilde{\mathrm{prod}}_i$ to refer to i's production when the period t_j of reference is evident or of no significance. Furthermore, we will be using prod_C and $\widetilde{\mathrm{prod}}_C$ to denote the production and expected production of a cooperative C of DER agents. The difference between the t_j-values of the estimated and actual production vectors, gives the DER (or, similarly, CVPP) *prediction error* for the t_j period. Note that $\mathrm{prod}_C = \sum_i \mathrm{prod}_i$, as the total CVPP production is just the sum of the production of its DERs. Further, we assume that $\widetilde{\mathrm{prod}}_C = \sum_i \widetilde{\mathrm{prod}}_i$.[1]

Now, essential functionality for the CVPP operation includes rules and procedures for *(a)* the distribution of revenues, *(b)* the aggregation of individual production estimates into CVPP-wide ones, and *(c)* membership management (admitting and expelling members). That functionality might be located on some central agent responsible for "running" the CVPP, or it could be potentially distributed over several agents. The functionality localization details are unimportant to our work here. Instead, we proceed to describe the aforementioned CVPP operational activities in depth.

4. TRUTHFUL AND RELIABLE CVPPS

In this section, we present a payment mechanism that can be employed by the Grid to promote the formation of DER cooperatives. The mechanism addresses the main hurdles the Grid faces with respect to DERs' integration—namely, the *unreliability* of their production (given DERs' dependance on uncontrollable factors, like the weather), and their *large numbers* (given that it is anticipated that hundreds of thousands of DERs would be eventually embedded within a given country's distribution network).

To begin, we elucidate the main requirements of the Grid with respect to its interaction with CVPPs, and proceed to show how they translate into the features of our payment mechanism.

(a) Reliability of supply: The Grid operators are responsible for compiling production schedules to pass to the large power plants. Currently, these are based on the predicted demand for electricity. As more supply originates from smaller generators, their predicted output will also need to be incorporated into the Grid production scheduling process. Hence, the Grid requires any entity interacting with it (such as a DER or a CVPP) to provide it with reliable production estimates, and to be able to honour any agreement to supply a specific amount. Subsequently, the Grid would be willing to reward producers that are proven to be reliable suppliers.

(b) Need to minimize the number of entities the Grid interacts with: As already mentioned, widespread small-scale production will result in a huge number of DERs being connected to the Grid. However, the Grid would obviously prefer to interact with a small number of electricity producers, as this makes it easier to manage and settle accounts. This requirement mirrors the scenario on the consumption side, where the Grid interacts with only a few large utility companies, which, in turn, interact with the millions of individual consumers. Thus, it is imperative for the Grid to promote the formation of *large* CVPPs, each with a sizeable production capacity. Larger CVPPs make it possible for the Grid to interact with a smaller number of entities, and also promote supply reliability.

[1]It is conceivable that CVPP-wide estimates do *not* necessarily equal $\sum_i \widetilde{\mathrm{prod}}_i$. This would have no impact in our results.

4.1 Payment Mechanism

With this list of requirements in mind, we now put forward a pricing mechanism that the Grid can use when making payments to the CVPPs for their contributed energy. As discussed, the CVPPs provide their estimated production for each day-ahead settlement period to the Grid authority. As stated above, $\widetilde{\mathrm{prod}}_C$ is the estimated production declared by CVPP C, and prod_C its actual production in the given time interval. Let price be the electricity *base price* (per kWh). The "Grid-to-CVPP" payment from the Grid G to C is:

$$V_{G,C} = \frac{1}{1 + \alpha|\widetilde{\mathrm{prod}}_C - \mathrm{prod}_C|^{\beta}} \cdot log(\mathrm{prod}_C) \cdot \mathrm{price} \cdot \mathrm{prod}_C \quad (1)$$

The three first factors of this payment function (or pricing mechanism) represent the *actual price* being offered by the Grid to C. Multiplying them with the actual CVPP production (the fourth factor, prod_C) gives the actual payment to C. The mechanism has specific properties that satisfy the requirements mentioned above:

(1) The first factor, $\frac{1}{1+\alpha|\widetilde{\mathrm{prod}}_C - \mathrm{prod}_C|^{\beta}}$, depends on the accuracy of the estimates provided by the CVPP. This *accuracy factor* is a bell-shaped function of $\widetilde{\mathrm{prod}}_C$ given the actual production prod_C parameter, as the one whose graph is depicted in Fig. 1. It simplifies to 1 when $\mathrm{prod}_C = \widetilde{\mathrm{prod}}_C$, proportionally decreases as the difference between them increases. Importantly, this decrease is independent of whether prod_C is greater than $\widetilde{\mathrm{prod}}_C$ or vice versa. Parameters α and β are functions of prod_C and determine the exact shape of the curve, and can be tuned so that the factor approaches zero for $\widetilde{\mathrm{prod}}_C$ estimates that are at a distance of prod_C away from the actual prod_C production. The use of this factor guarantees that the CVPP has the incentive to truthfully provide a highly accurate estimate of its production, as acting otherwise leads to a loss of revenue (at least in expectation).

(2) The second factor, $log(\mathrm{prod}_C)$, increases with production and thus encourages a large CVPP size. Therefore, CVPPs with more DER members generate more energy and obtain a better overall price than smaller ones. Nevertheless, being a *log* function, the factor flattens eventually at very high production amounts. This means that, though the formation of large CVPPs is encouraged, the emergence of a single CVPP containing all DERs is not a necessary consequence. Even though small CVPPs have an incentive to merge initially, they will not merge *ad infinitum*, as there is no visible benefit after some point due to the limit linearity of the function. Of course, other reasons to prevent merging, such as geographical or technical restrictions, might exist.

(3) The third factor, price, is determined by the Grid either through supply and demand in the electricity market or through other means, and will be the same for all CVPPs participating in the market.

It is evident that this pricing mechanism promotes cooperative participation in the market, and captures the aforementioned list of requirements. First, it promotes supply *reliability*, by guaranteeing that CVPPs receive higher revenues for accurate estimates. A CVPP has an incentive to provide as accurate an estimate as possible, and has no incentive to strategize about it, as the estimate is only used by the function to check how far off the actual production was from the promised supply target. As shown above, wilfully providing a wrong or biased estimate does not improve and mostly

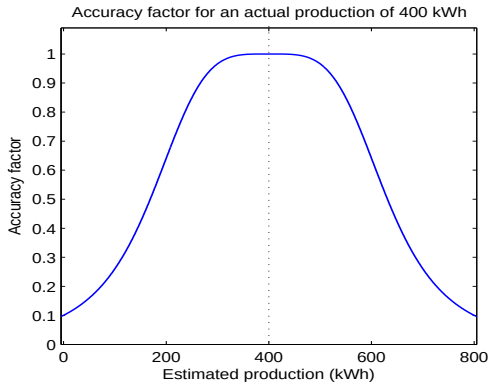


Figure 1: An accuracy factor function diagram

decreases the payment to the CVPP for the same amount of actual production. Thus, the mechanism promotes *truthfulness* on behalf of the CVPPs. Similarly, the mechanism also promotes *efficiency* at the Grid level by incentivizing the formation of large cooperatives, each attaining a substantial production aggregate.

4.2 Truthful and Reliable DER-Agents

The above mechanism incentivises the CVPP to provide accurate estimates about its production. As discussed earlier, the production of a CVPP is nothing but the aggregate of the production of all the DERs composing it. Therefore, the CVPP requires accurate estimates of production from the DERs in order to be able to calculate the production estimate to provide to the Grid. Given this, the payment from the CVPP to the DERs should encourage the DERs to truthfully provide good estimates of their production. Evidently, this mirrors the scenario between the Grid and the CVPP. Taking cue from that, we use the same principle for this "CVPP-to-DER" payment function as the Grid-to-CVPP one. Thus, the payment from CVPP C to member i for supplied energy prod_i is:

$$V_{C,i} = \frac{z}{1+\alpha|\widetilde{\text{prod}}_i - \text{prod}_i|^\beta} \cdot \frac{\text{prod}_i}{\text{prod}_C} \cdot V_{G,C} \qquad (2)$$

We now describe the function in detail, demonstrating how it elicits truthful and as accurate as possible predictions from the DERs.

(1) As in Eq. 1, the first factor, $\frac{z}{1+\alpha|\widetilde{\text{prod}}_i - \text{prod}_i|^\beta}$, is an *accuracy factor*, encouraging the DER to provide the CVPP with its best possible production estimate. It equals z if the estimate was accurate, and drops following a bell curve otherwise. Notice that z is simply a normalization factor used to redistribute the entire $V_{G,C}$ amount back to the members. Redistribution is in proportion to the members' production and prediction accuracy—this can be easily seen with $z = \frac{\text{prod}_C}{\sum_i \text{prod}_i/(1+\alpha|\widetilde{\text{prod}}_i - \text{prod}_i|^\beta)}$. Alternatively, the CVPP can set $z = 1$ and use the residual profits to pay for maintenance costs, recruiting new members, or other such purposes.

(2) The second factor, $\frac{\text{prod}_i}{\text{prod}_C}$, gives the proportion of energy contributed by this DER w.r.t. the total CVPP production, making the payment distribution fair across all DERs.

(3) The last factor, $V_{G,C}$, denotes the total amount that is to be divided among the constituent DERs, and corresponds to the payment received by the CVPP from the Grid.[2]

To recap, by employing this payment function the CVPP promotes truthful and highly accurate predictions from its constituent DERs. A DER has an interest to truthfully and accurately report, since otherwise it does not receive the full payment corresponding to the energy it actually produced.

[2]Of course, this could be reduced by subtracting an amount if this is required to account for CVPP fees or maintenance costs.

4.3 Payment Schemes Render Stable CVPPs

Here we provide a further, game-theoretic justification for the payment scheme used by the Grid to reward CVPPs, and for that used by a CVPP to reward its members. Specifically, we show that, given the functions used to reward the cooperatives and the members payment scheme described above, and assuming that all CVPP members' stated production estimates are accurate, no members' subset has an incentive to break away and form a smaller cooperative. In addition, this result promotes the goal of large CVPP sizes.

To demonstrate this, we employ the concept of *the core* [9], the strongest of the game-theoretic solution concepts used to describe coalitional stability. Some preliminaries from cooperative game theory are in order. To begin, let N, $|N| = n$, represent a set of players; a *coalition* is a subset $S \subseteq N$. Then, a *(transferable utility) coalitional game* $G(N; v)$ is defined by its *characteristic function* $v : 2^N \mapsto \mathbb{R}$ that specifies the *value* $v(S)$ of each coalition S [13]. Intuitively, $v(S)$ represents the maximal payoff the members of S can jointly receive by cooperating, and the agents can distribute this payoff among themselves in any way. While the characteristic function describes the payoffs available to coalitions, it does not prescribe a way of distributing these payoffs. An *allocation* is a vector of payoffs $\boldsymbol{x} = (x_1, \ldots, x_n)$ assigning some payoff to each $i \in N$. Then, the *core* is the set of $\boldsymbol{x}$ payoff allocations with the property that no coalition of agents can guarantee all of its members a payoff higher to what they currently receive under $\boldsymbol{x}$. As such, no coalition would ever be motivated to break away from the grand coalition of all agents. Now, let $x(S)$ denote the payoff allocated by $\boldsymbol{x}$ to agents $S \subseteq N$, i.e., $x(S) = \sum_{i \in S} x_i$. Then, formally,

DEFINITION 1. *An allocation $\boldsymbol{x}$ is in the* core *of $G(N; v)$* iff $x(N) = v(N)$ **and** *for any $S \subseteq N$ we have $x(S) \geq v(S)$.*

That is, the value $v(N)$ of the grand coalition is efficiently distibuted by $\boldsymbol{x}$ among all agents, and the payments specified by $\boldsymbol{x}$ are such that any S already receives at least its value $v(S)$. The core of a game can be non-empty. Worse than that, it is in general **NP**-hard to determine the non-emptiness of the core (see, for example, [5]).

Returning to our setting, consider the formation of a CVPP as a coalitional game, with the characteristic function describing the value that any subset of DERs can derive by working together as a team, and the CVPP intuitively corresponding to the grand coalition of all agents. In our case, interestingly, assuming truthful and accurate DER estimates, the form of the characteristic function, $v(S) = log(\text{prod}_S) \cdot \text{price} \cdot \text{prod}_S$, allows us to prove that the payments allocated by Eq. 2 constitute a core-stable allocation, which also implies that the core of the game is always non-empty.

THEOREM 1. *Let $C = \{1, \ldots, n\}$ be a cooperative of $|C| = n$ agents, and let $G(C; v)$ be the coalitional game with characteristic function $v(S) = log(\text{prod}_S) \cdot \text{prod}_S \cdot \text{price}$ determining the value of each subset $S \subseteq C$ of agents. Consider the payoff allocation $\boldsymbol{x}$ where each agent i in C is paid according to Eq. 2—i.e., proportionally to i's contribution to the production of the CVPP (given $\widetilde{\text{prod}}_i = \text{prod}_i$). Then, $\boldsymbol{x} \in core(G)$.*

PROOF. We will show that $\boldsymbol{x}$ is in the core. We know that $\boldsymbol{x}$ distributes all payoff to the agents efficiently and therefore $x(C) = v(C)$, where $v(C) = V_{G,C}$, so the first condition of Def. 1 holds. Assume for the sake of contradiction that $\boldsymbol{x}$ is not in the core. Then, there exists some $S \subseteq C$ s.t. $v(S) > x(S)$. But $x(S) = \sum_{i \in S} x_i = \frac{\text{prod}_S}{\text{prod}_C} v(C)$ (this is easy to see by setting $\widetilde{\text{prod}}_i = \text{prod}_i$ for all i in Eq. 2). Thus: $v(S) > \frac{\text{prod}_S}{\text{prod}_C} \cdot \text{prod}_C \cdot log(\text{prod}_C) \cdot$

$\text{price} \Leftrightarrow \text{prod}_S \cdot log(\text{prod}_S) \cdot \text{price} > \frac{\text{prod}_S}{\text{prod}_C} \cdot \text{prod}_C \cdot log(\text{prod}_C) \cdot \text{price} \Leftrightarrow log(\text{prod}_S) > log(\text{prod}_C)$. But, since $S \subseteq C$, this is impossible. Thus, $\boldsymbol{x}$ is in the core of $G(C; v)$. □

Thus, the choice of the Grid-to-CVPP and CVPP-to-DER payment schemes described above is well justified from a game-theoretic, coalitional stability point of view also.

5. QUANTIFYING PREDICTION ERRORS

In Section 4.1 we introduced the payment function of CVPPs to their members, based partially on the accuracy of their predictions. Here we propose several methods for quantifying the uncertainty in DER predictions, and distinguishing between different types of prediction errors. This will prove helpful for devising methods to handle CVPP membership (in Section 6). To begin, consider the examples of a virtual power plant that aggregates the supply from several DER wind farms (belonging to different stakeholders) distributed in a geographical area, or from a set of solar panels installed by different houses in an extended neighbourhood. Each DER can make an error in the prediction of its future output for a given half-hour period. It is useful to distinguish between two main classes of errors:

(a) **Systematic errors:** This error type is caused by the inherent uncertainty in predicting an outside variable that is used as an input by several DERs while calculating their production estimates. For renewables, this is most likely a weather-related variable, such as wind speed or solar power. So, for example, if the meteorological office is innacurate in its prediction of wind speed at a certain time in a local area, then all the wind turbines in that area may register an error in their predicted production. We call this type of error *systematic*, as it is common to all energy resources that rely on that factor, and it is outside the control of individual DERs.

(b) **Residual errors (DER specific):** Besides the systematic errors, the predictions of an individual DER may be affected by errors caused by factors specific to itself, and (at least partially) under its control. In the example discussed above, even if a wind turbine is supplied with very accurate predictions of wind speed, its prediction of its actual output may not be that accurate (because it is an older turbine, requires maintenance work, and so on).

Against this background, we now propose a statistical method for distinguishing between the different types of prediction errors. Consider a dataset consisting of m DERs in a CVPP, which belong to the same category of energy producers (e.g., wind turbines from the same area). For each of these DERs, n half-hour data points are available within some large time period $T = \{1, \ldots, n\}$ (n can be quite large as the data can span several days, weeks or months).

Formally, let $\widetilde{\text{prod}}_{i,t}$ and $\text{prod}_{i,t}$ denote the estimated and actual production of DER i in a half-hour interval t. Moreover, let $\Delta_{i,t} = \text{prod}_{i,t} - \widetilde{\text{prod}}_{i,t}, \forall i = \{1, \ldots, m\}, \forall t \in T$ denote i's prediction errors in t. Given the 2-dimensional error matrix with entries $\Delta_{i,t}$ as defined above, we can define the *average* prediction error across all DERs for some $t \in T$ as: $\mu_t^{\Delta} = \frac{\sum_{i=1}^{m} \Delta_{i,t}}{m}$.

In what follows, we denote by $\boldsymbol{\Delta}_T^i$ the n-vector of errors corresponding to energy producer i for every interval $t \in T$ ($\boldsymbol{\Delta}_T^i$ is a row of $\Delta_{i,t}$ error matrix entries corresponding to i), and by $\boldsymbol{\mu}_T^{\Delta}$ the n-vector containing the average prediction errors across all DERs for all time steps $t \in T$. We can now compute the *Pearson correlation coefficient* ρ_i^{Δ} between vectors $\boldsymbol{\Delta}_T^i$ and $\boldsymbol{\mu}_T^{\Delta}$ as:

$$\rho_i^{\Delta} = \frac{cov(\boldsymbol{\Delta}_T^i, \boldsymbol{\mu}_T^{\Delta})}{\sigma(\boldsymbol{\Delta}_T^i)\sigma(\boldsymbol{\mu}_T^{\Delta})} = \frac{\sum_{i=1}^{n}(\Delta_{i,t} - \overline{\Delta_i})(\mu_t^{\Delta} - \overline{\mu_t^{\Delta}})}{\sqrt{\sum_{i=1}^{n}(\Delta_{i,t} - \overline{\Delta_i})^2}\sqrt{\sum_{i=1}^{n}(\mu_t^{\Delta} - \overline{\mu_t^{\Delta}})^2}} \quad (3)$$

where $cov(\boldsymbol{\Delta}_T^i, \boldsymbol{\mu}_T^{\Delta})$ denotes the statistical covariance between the two vectors $\boldsymbol{\Delta}_T^i$ and $\boldsymbol{\mu}_T^{\Delta}$, $\sigma(\boldsymbol{\Delta}_T^i)$ and $\sigma(\boldsymbol{\mu}_T^{\Delta})$ are their standard deviations, and $\overline{\Delta_i} = \frac{\sum_{t=1}^{n} \Delta_{i,t}}{n}$ and $\overline{\mu_t^{\Delta}} = \frac{\sum_{t=1}^{n} \mu_t^{\Delta}}{n}$ their means.

Intuitively, for each energy producer i, $\rho_i^{\Delta} \in [0, 1]$ shows how correlated its errors in predicted production were with the average errors made by the energy producers in the same category in the CVPP. In our wind turbine example, if the coefficient ρ_i^{Δ} for wind tubine i is high, it means that this turbine tends to make a prediction error when all other wind turbines in its area make a prediction error of similar proportions. Thus, its error is mostly of a "systematic" nature. If there is an uncertain, outside factor (e.g., wind speed prediction) causing an error for all these turbines, then the errors can be assigned to this factor. Conversely, if ρ_i^{Δ} is low, the errors of this wind turbine are caused by its own functioning/prediction capabilities, and appear unrelated to those of similar producers.

With this at hand, statistical theory [4] allows us to define two important measures for the error vector of each producer i: the *fraction of variance explained* by the systematic factor (also called the coefficient of determination), $FVE_i^{\Delta} = (\rho_i^{\Delta})^2$, and the *fraction of variance unexplained* (or, the fraction of residual variance) $FVU_i^{\Delta} = 1 - (\rho_i^{\Delta})^2$. In essence, these measures determine the percentage of the variance in DER i's prediction errors that can be explained by systematic factors. Thus, we can separate the variance $\sigma(\boldsymbol{\Delta}_T^i)$ in the prediction errors of each i over period T into the systematic and the residual variance, the latter defined as:

$$\sigma_{res}(\boldsymbol{\Delta}_T^i) = FVU_i^{\Delta}\sigma(\boldsymbol{\Delta}_T^i) = [1 - (\rho_i^{\Delta})^2]\sigma(\boldsymbol{\Delta}_T^i) \quad (4)$$

Thus, the residual variance provides us with a tool to determine whether the prediction error of a specific DER i is due to factors that do not affect other energy producers of the same nature and in the same area. As we shall see, this tool can be used to inform CVPP membership management decisions.

6. MANAGING CVPP MEMBERSHIP

In Section 4.3 we showed that, given the payment function described in Eq. 1, coalitions representing CVPPs are stable, in the sense that DERs do not have a financial incentive to abandon them. However, this result only holds when the DERs composing the coalition are always able to provide accurate, error-free estimates of their production. In general, cooperatives do not have an incentive to expel members, given that more members means greater expected production and thus greater expected revenues. At the same time, given Eq. 1, it is also true that, if certain DER members are consistently unreliable in their production estimates, then the additional penalty that the CVPP suffers due to increased unreliability can in the long term offset any benefits from an increased overall production. Therefore, a CVPP should perform a regular evaluation of its individual members' performance, based on which it may decide to expel some of them. In this section, we provide methods for such an evaluation.

Formally, as in Section 5 above, we consider the performance of m DERs belonging to a CVPP in a discretized time period T consisting of $t = 1, \ldots, n$ half-hour periods. Furthermore, we denote by $C \backslash i$ the CVPP C if DER i was not its member. Given the Grid-to-CVPP payment of Eq. 1, we define the *marginal contribution* (or *marginal value*) of DER i to cooperative C in period t to be:

$$V_{i \to C;t}^{mg} = V_{G,C;t} - V_{G,C \backslash i;t} \quad (5)$$

Intuitively, the marginal contribution of DER i to the cooperative at any time interval is the difference between the payment that a cooperative actually receives, and the payment it would have received had i not been part of the cooperative.[3] Note that this marginal

[3]Incidentally, although perhaps intuitively appealing, using the

value is influenced by both the estimated and actual productions, $\widetilde{\text{prod}}_{i,t}$ and $\text{prod}_{i,t}$, of DER i (and, implicitly, by its errors $\Delta_{i,t}$).

Given this, we now propose a method to assess the long-term performance of i within a time frame of interest T. The same mechanism could be applied to the process of deciding whether to accept a new member in the CVPP, if historical data regarding its predictions' reliability were available.

Note that a first, simple solution would be to assess the members' performance by ranking them according to their marginal contribution during a time period T consisting of intervals $t = 1, \ldots, n$. That is, we can simply add the marginal contributions of DER i for the intervals $t \in T$: $V^{mg}_{i \to C;T} = \sum_{t=1,\ldots,n} V^{mg}_{i \to C;t}$. Then, each producer can be ranked by its marginal contribution to the revenues of the CVPP, as described by $V^{mg}_{i \to C;T}$ across the period T of interest. This method captures the exact contributions of members, but does not account for systematic errors. So, for example, a DER situated in an area with poor wind/solar power prediction for a given period, would be penalized for elements outside its control.

A fairer method would be to use the residual variance specific to each DER. Such a method involves ranking the producers according to their residual variances, as computed in Eq. 4, over a period T. The least accurate producers could then possibly be expelled from the CVPP, as a high residual variance shows their prediction accuracy underperforms that of others in the same area for a considerable period of time. However, that would have the disadvantage that it completely disregards the contributions of individual DERs to the CVPP revenues. Indeed, a CVPP could be reluctant to expel a member that, though consistently inaccurate, still contributes significantly to the CVPP production and, therefore, revenues.

Thus, here we propose a method that actually *weighs* the marginal contribution of a DER by its residual variance (normalized to $[0, 1]$ through division by the sum of residual variances across all m agents). Specifically, C calculates, for each i over T, the following:

$$\text{score}^i_T = (1 - \frac{\sigma_{res}(\mathbf{\Delta}^i_T)}{\sum_{j=1}^{m} \sigma_{res}(\mathbf{\Delta}^j_T)}) V^{mg}_{i \to C;T} \quad (6)$$

Intuitively, DERs with higher residual variance have their marginal contribution disregarded more, while still taking some credit for it. The CVPP then ranks the DERs in terms of their score, and has the option to expel members with low performance. The advantage of this method is that it avoids punishing individual DERs for systematic errors, while taking into account their marginal contributions at the same time.

7. EMPIRICAL EVALUATION

We tested our payment mechanisms by examining the incentives of a set of individual DERs to form a cooperative, in the context of a renewables generation scenario. The data used in our analysis comes from the *Sotavento* experimental wind farm, in Galicia, Spain, and is made freely available for research purposes from their website (http://www.sotaventogalicia.com/). The farm produces roughly the energy required to serve 12,000 homes. In what follows, we first discuss how we constructed individual wind turbine profiles from the available data, and describe our prediction scenarios. We then apply our mechanisms to this setting, demonstrate the benefits to individual turbines from forming a cooperative, and evaluate our method for ranking DERs according to prediction performance.

To begin, the main characteristic of a wind turbine is its *power curve*, describing, for a given level of wind speed, its electrical output (in MW). The generic power curve of wind turbines is typically a *sigmoid function*, with a threshold level, beyond which the power output increases more sharply. A turbine's *nominal capacity* describes its maximum power (and, subsequently, energy produced per hour) output for "optimal" wind speed.

The Sotavento farm contains 24 wind turbines, with an installed nominal capacity of 17.5 MW which jointly produce an average of 38,500 MWh yearly. The available dataset we used in our experiments contains, for each *hourly* slot for the entire year from 2 September 2009 to 2 September 2010, both the actual wind speeds recorded, as well as the farm production (in kWh) for that time slot. There are 8600 records/year provided in total, due to some records being corrupt. Fig. 2 shows a scatter plot of all the yearly data points from Sotavento, as well as the power curve (the sigmoid function) we derived based on this data.

Next, we divided the derived curve for the entire farm (i.e., in our terminology, the CVPP) into 24 identical power curves, one for each individual wind turbine (DER). Note that, while no detailed data was available about individual turbines, considering them equal in nominal production capacity is realistic, and sufficient to illustrate the functioning of our model. Therefore, based on the real data, each of the 24 turbines has a nominal capacity of $\sim$700 kW (or, it can supply $\sim$500 homes). If w_t is the wind speed at an hourly timepoint t (in m/s), the *generic* power curve of each turbine is:

$$prod^{generic}_{i,t}(w_t) = \frac{700}{1 + e^{0.66*(9-w_t)}} \quad (7)$$

The shape of this function for each individual turbine is the same as that in Fig. 2, but with a maximal capacity of 700 kW.

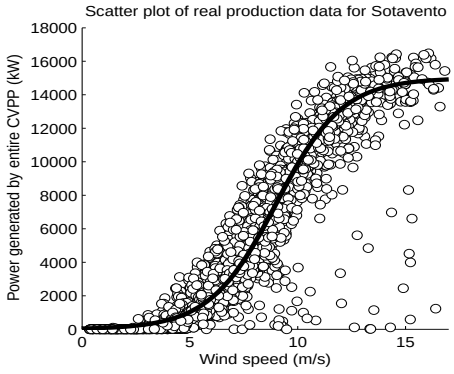


Figure 2: Scatter plot of the yearly data points from Sotavento.

7.1 Forming CVPPs of Wind Turbines

Although the Sotavento site provides real data about production and wind speeds, it does not provide us with any long-term data about the *predictions* of individual turbines. Furthermore, all wind turbines in Sotavento are owned by the same entity (a government agency). By contrast, our goal is to examine more decentralized settings, with these turbines belonging to individual stakeholders. Specifically, our aim here is to verify experimentally that, given our payment mechanisms, "self-interested" turbines (DERs) with different abilities have an incentive to coalesce into a CVPP.

To this end, we consider experimental scenarios in which the main parameter varied is the prediction ability of individual turbines regarding future production. Formally, given a wind speed

marginal contributions to distribute the CVPP revenues to the DERs is problematic as an approach, because it compromises DER truthfulness. Specifically, it provides agents with a reason to strategize and base their reports on those of others, since their payment would be based on whether they can accurately predict and "correct" the reports of others, so that they are awarded the marginal gains resulting from improved CVPP performance. Though the study of such collective "auto-correction" mechanisms is perhaps interesting, it is out of the scope of this work.

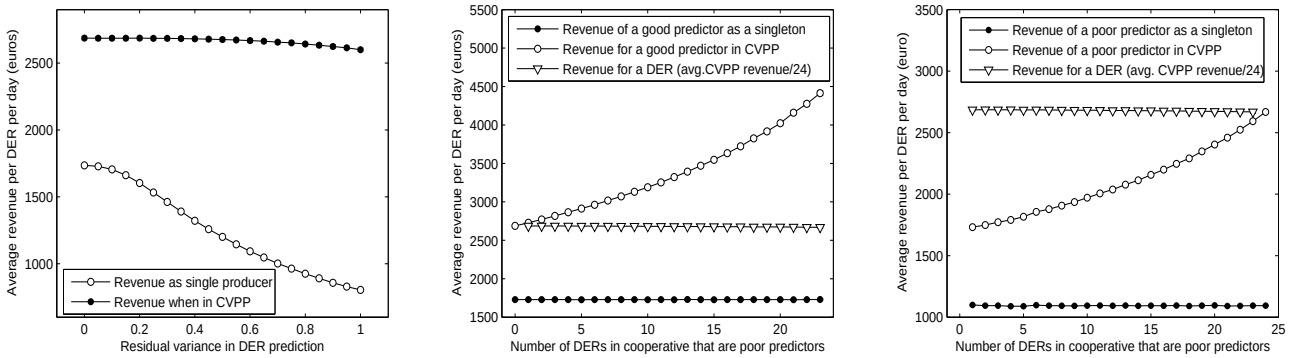


Figure 3: Benefits from joining a CVPP vs. selling to the Grid as an independent producer (singleton): (a) symmetric case; (b) asymmetric case: performance of *good predictors*; and (c) asymmetric case: performance of *poor predictors*. Averages are over 86,000 steps (8600 hourly time points available in a year & yearly simulation run 10 times). Error bars too small to be visible.

prediction w_t, we first compute a *generic* (idealized) production $prod_{i,t}^{generic}$ of each wind turbine i at time t using Eq. 7.[4] Then, the *actual* production for each DER $i = 1 \dots 24$ is given by

$$prod_{i,t} = prod_{i,t}^{generic} \cdot \mathcal{N}(1, \sigma_{syst})$$

where the variance factor σ_{syst} captures the systematic error that is common to all turbines (i.e., the actual wind speed is not the same as predicted). While the actual productions are drawn independently for each DER i, the deviation σ_{syst} of the normal perturbation distribution is the same for all, reflecting the fact that they are all subject to the same uncertain, outside factor (wind speed).

Now, the DERs can have rather different capabilities w.r.t. deriving future production estimates. This is captured by a DER-specific (or residual) error factor σ_{res}^i. Then, the estimated production reported by each DER $i = 1 \dots 24$ is:

$$\widetilde{prod}_{i,t} = prod_{i,t}^{generic} \cdot \mathcal{N}(1, \sigma_{res}^i)$$

Against this background, we use two simulation settings to explore the benefits to individual DERs from being in a CVPP. In both settings, the number of DERs is fixed at 24 (as in Sotavento), each with generic production functions as in Eq. 7, and with the systematic error variance set to $\sigma_{syst} = 0.1$. We set $\text{price} = 0.05$; this is combined with the first two factors of Eq. 1 to give the *actual price* in euros/kWh. We consider the following cases:

(a) **The symmetric case:** All DER-agents are *equally good* or *equally bad* in predicting their own production. In other words, the residual deviation σ_{res}^i is the same accross all agents i.

(b) **The asymmetric case:** The agents in the cooperative are divided into 2 classes: one of *good predictors*, having a low residual deviation $\sigma_{res(low)} = 0.05$ regarding their production estimates, and a second class of *poor predictors*, having a high residual deviation of $\sigma_{res(high)} = 0.6$. The relative proportion of the two class sizes varies from 0/24 to 24/24 (out of the 24 agents in the CVPP).

For both scenarios, we ran a series of experiments where the real wind data for all hourly intervals for an entire year was used. The simulation of the hourly wind speeds over the entire year was repeated 10 times[5] to reduce the outcomes' variance, resulting to $86,000$ tests for each data point shown in the results of Fig. 3.

[4] As already discussed, our simulation uses the real wind speeds for each hour for the 365 days in the year.

[5] The simulation parameters were chosen with the computational requirements of the various experimental settings in mind, but in all cases our results are statistically significant.

Joining a CVPP is beneficial in the symmetric case.

Turning our attention to Fig. 3(a) which depicts the results for the symmetric predictions scenario, we can see that, whatever the residual uncertainty in prediction is, individual DERs have an incentive to join together to form a CVPP. For small values of the deviation in prediction error σ_{res}^i (i.e., when all agents predictions are almost entirely accurate), this effect is due to the superadditive structure of the reward function of Eq. 1. This was not surprising, given the result of Theorem 1. Interestingly, however, the impact of our payment schemes is even more profound when highly inaccurate DERs (i.e., those with high values of residual variance) are considered. In this case, the revenue for singleton DERs more than halves when compared to their average gains when participating in a CVPP (from 1700 to 800 euro/day), as the agents are punished by the Grid for their inability to predict their production accurately.

As expected, when agents interact with the Grid as a CVPP, the cooperative's revenue also drops when its members become less accurate in prediction. However, the drop is much smaller, from 2700 to about 2600 euro/day for each of the 24 members. This is mainly because, if added over the entire cooperative, residual prediction errors cancel each other. Thus, quite interestingly, even a virtual power plant consisting of 24 DERs with poor prediction ability is able to issue a reasonably accurate estimate to the Grid.

Results for the asymmetric case.

We now examine a setting in which DERs can be separated into two distinct classes, one of *good* and one of *poor* predictors (with a residual variance of $\sigma_{res(low)} = 0.05$ and $\sigma_{res(high)} = 0.6$ respectively). The main experimental parameter varied here is the number of agents of each type that make up the CVPP; these are varied from 0 to 24.

Simulation results appear in Fig. 3(b) and 3(c). We observe that, in general, both types are better off being in a CVPP than interacting with the Grid as singletons. This is regardless of whether the other participants are good predictors or poor. However, there are some additional interesting observations to be made in this setting.

Somewhat surprisingly, *good predictors* actually do much better if the rest of the cooperative members are poor. The reason for this is the way the CVPP-to-DER payment redistribution function works. If an agent is the only accurate one (or among the few accurate ones) in the cooperative, it gets a large proportion of the joint payments, as it is among the few with a low error factor, and thus enjoys high returns following the (normalized to reward accuracy,

as explained in Section 4.2) redistribution of CVPP's revenues.

In general, *poor predictors* also have a strong incentive to join the CVPP, as the results in Fig. 3(c) show. An interesting point to note is that it would appear from these results that both poor and good predictors prefer the other agents in the cooperative to be poor predictors (unless their errors are all biased towards the same direction and thus do not cancel out—an improbable scenario for large CVPPs). However, as shown in our figures, a *random* member of the cooperative would on average expect to do slightly better if the number of good predictors is high, as the cooperative as a whole gains more revenue on average in that case.

7.2 Ranking DERs by Prediction Performance

For the last set of results, we use a similar setting as the asymmetric case described above. We divide the DER-agents into two categories: *good predictors* (with $\sigma_{res(low)} = 0.05$) and *poor predictors* (with $\sigma_{res(high)} = 0.3$). The number of each agent type in the cooperative was varied from 1 to 23 (out of 24 agents in total). Recall that in Section 6, two methods for assessing the contribution of a DER i to the CVPP were discussed: one based on only its marginal contribution to the cooperative, and the other taking into account both i's marginal contribution and the residual error variance $\sigma_{res}(\Delta_T^i)$. In our experiments, we compare these two methods, taking T to be one year of hourly data, as before.

The graph in Fig. 4 shows, for settings with $k = 1 \ldots 24$ poor predictors, the number of *real* poor predictors detected by each method (i.e., how many actual poor predictors are among the k lowest scoring agents returned by each method used). Note that the ranking shown is actually an average over 25 runs, sufficient to reduce the results' variance to very low levels (since, in fact, each data point represents the results from 25 years of real hourly data). As we can see, the method that weighs marginal values by residual variance (Eq. 6), making use of the techniques of Section 5, is clearly better in distinguishing poor predictors than ranking by marginal contributions alone; in fact, it rarely identifies a predictor wrongly in this setting. In contrast, the strategy of ranking solely by marginal values does degrade, especially when the number of good predictors roughly equals that of poor ones. In any case, the results in this setting show that both methods manage to distinguish poor predictors from good ones with a very high degree of accuracy.

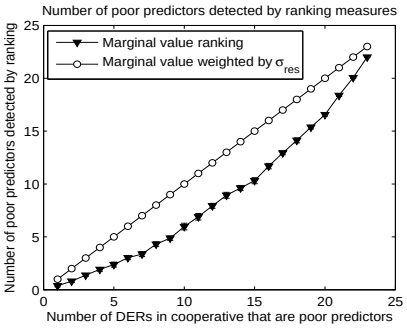


Figure 4: Results for the efficiency of ranking measures.

8. CONCLUSIONS

In this paper we applied several game-theoretic ideas in the energy domain. We presented a pricing mechanism that can be used as an alternative to feed-in tariffs, in order to promote the creation and cost-efficient operation of DER cooperatives. We also proposed a method to allocate CVPP revenues to its members, and showed that this method promotes CVPP stability (assigning payoffs that are core-stable, under the condition of DER accuracy). We also showed that the payment functions incentivize truth-telling when CVPPs interact with the Grid and when DERs interact with the CVPP; and that our methods promote supply reliability and production efficiency. Moreover, we provided a generic method for CVPP membership management, which was experimentally shown to be successful in ranking DERs w.r.t. predictions' accuracy. Crucially, our ideas were evaluated on scenarios using data from a real-world wind-farm. Our results confirm that joining CVPPs which make use of our proposed payment schemes is almost always beneficial to any individual DER.

In future work, we intend to study alternative pricing schemes to the one proposed here. For instance, residual errors-related information could perhaps be incorporated in the payment function. Doing so optimally and in a fair manner is not straightforward, since determining the residual part of the error requires the study of an agent's performance over an extended period, while the payment function rewards the agent for its immediate performance. We also intend to examine alternative ways to distribute rewards among CVPP members, perhaps by utilizing their *Shapley value* [9]. Although its exact calculation is an intractable problem, the use of *bilateral* Shapley value approximation schemes could be an option. Furthermore, assuming DERs could provide production estimates in the form of a full distribution (rather than just an expected value), it would be interesting to devise *scoring rules* [11] to elicit those estimates, and to reward both estimates that turn out to be accurate, and those provided with high precision (low variance). Moreover, we would be interested in implementing a web service to accommodate CVPP formation and member management activities.

9. REFERENCES

[1] J. Contreras, M. Klusch, and J. Yen. Multi-agent coalition formation in power transmission planning. In *Proc. of the 4th Int. Conf. on Artificial Intelligence in Planning Systems*, 1998.

[2] J. Contreras, F. F. Wu, M. Klusch, and O. Shehory. Coalition formation in a power transmission planning environment. In *Proc. of Conf. on Practical Applications of Intelligent Agents and Multiagent Systems*, 1997.

[3] E. M. Davidson, M. J. Dolan, S. D. J. McArthur, and G. W. Ault. The use of constraint programming for the autonomous management of power flows. In *Proc. of the 15th Int. Conf. on Intelligent System Applications to Power Systems*, pages 1–7, 2009.

[4] M. DeGroot and J. Schervish. *Probability and Statistics*. Addison-Wesley, 2002.

[5] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[6] A. Dimeas and N. Hatziargyriou. Agent based control of virtual power plants. In *Proc. of the 14th Int. Conf. on Intelligent System Applications to Power Systems*, pages 1–6, 2007.

[7] K. Kok, M. Scheepers, and R. Kamphuis. Intelligence in electricity networks for embedding renewables and distributed generation. In *Intelligent Infrastructures*. Springer, 2009.

[8] R. C. Mihailescu, M. Vasirani, and S. Ossowski. Towards agent-based virtual power stations via multi-level coalition formation. In *Proc. of the 1st Int. Workshop on Agent Technologies for Energy Systems*, pages 107–108, 2010.

[9] R. Myerson. *Game Theory: Analysis of Conflict*. Harvard Univ. Press, 1991.

[10] D. Pudjianto, C. Ramsay, and G. Strbac. Virtual power plant and system integration of distributed energy resources. *IET Renewable Power Generation*, 1(1):10–16, 2007.

[11] L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.

[12] U.S. Department of Energy. Grid 2030: A national vision for electricity's second 100 years, 2003.

[13] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, 1944.

[14] P. Vytelingum, S. D. Ramchurn, T. D. Voice, A. Rogers, and N. R. Jennings. Trading agents for the smart electricity grid. In *Proc. of AAMAS 2010*, pages 897–904, 2010.

[15] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based micro-storage management for the smart grid. In *Proc. of AAMAS 2010*, pages 39–46, 2010.

[16] C. S. K. Yeung, A. S. Y. Poon, and F. F. Wu. Game theoretical multi-agent modeling of coalition formation for multirateral trade. *IEEE Transactions on Power Systems*, 14(3):929–934, 1999.

How Agents Can Help Curbing Fuel Combustion – a Performance Study of Intersection Control for Fuel-Operated Vehicles*

Natalja Pulter
FZI Forschungszentrum Informatik
Karlsruhe, Germany
pulter@fzi.de

Heiko Schepperle
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
heiko@schepperle.de

Klemens Böhm
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
klemens.boehm@kit.edu

ABSTRACT

Traffic causes pollution and demands fuel. When it comes to vehicle traffic, intersections tend to be a main bottleneck. Traditional approaches to control traffic at intersections have not been designed to optimize any environmental criterion. Our objective is to design mechanisms for intersection control which minimize fuel consumption.

This is difficult because it requires a specialized infrastructure: It must allow vehicles and intersections to communicate, e.g., vehicles send their dynamic characteristics (position, speed etc.) to the intersection more or less continuously so that it can estimate the fuel consumption. In this context, the use of software agents supports the driver by reducing the necessary degree of direct interaction with the intersection.

In this paper, we quantify the fuel consumption with existing agent-based approaches for intersection control. Further, we propose a new, agent-based mechanism for intersection control, with minimization of fuel consumption as an explicit design objective. It reduces fuel consumption by up to 26% and waiting time by up to 98%, compared to traffic lights. Thus, agent-based mechanisms for intersection control may reduce fuel consumption in a way that is substantial.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems, intelligent agents*

General Terms

Algorithms, Economics, Measurement

Keywords

Agents, fuel consumption, traffic control, intersection control

*This work was part of the project DAMAST (Driver Assistance using Multi-Agent Systems in Traffic) which has been partially funded by init innovation in traffic systems AG (`http://www.initag.com/`).

Cite as: How Agents Can Help Curbing Fuel Combustion – a Performance Study of Intersection Control for Fuel-Operated Vehicles, N. Pulter, H. Schepperle, and K. Böhm, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 795-802.

1. INTRODUCTION

Mobility is a challenge for public authorities. Traffic causes pollution and – next to other factors – the climate change. Further, emissions of vehicles are closely linked to fuel consumption. The unsteady oil price and the expected oil shortage in the future make fuel consumption not only an issue for society but also for individual drivers.

When it comes to city traffic, intersections tend to be a main bottleneck. Traditional approaches for intersection control like traffic lights or roundabouts aim to increase throughput and to reduce waiting time. But they have not been designed with the intent to do any optimization with regard to an environmental criterion. This, with regard to fuel consumption, is the objective of this article.

If a vehicle does not know when it will be allowed to cross an intersection, it approaches it and – if necessary – decelerates or stops just before. Afterwards, it accelerates again. If a vehicle was informed about when to cross the intersection in advance, it could reach the intersection just in time, with less deceleration and acceleration. This decreases fuel consumption [7]. It also allows the intersection-control mechanism to orchestrate vehicles entering from different directions flexibly and efficiently. Thus, intersections should inform vehicles about their exact time slot in advance.

Doing so not only allows a vehicle to arrive at the intersection just in time, but also with sufficient speed. This leads to shorter time slots and to a higher throughput.

The fuel consumption of a vehicle depends on characteristics like size, engine capacity and rolling resistance. Dynamic parameters like speed and acceleration are important as well. With existing intersection-control mechanisms, those various parameters are unknown to the mechanism. The mechanisms envisioned have to consider not only static, but also dynamic parameters which can change at any time. Thus, the mechanisms sought require a specialized infrastructure both in vehicles and at intersections which allows them to communicate.

Another observation that is important here is that it is easy to arrive at the intersection at a certain time or with a certain speed. But doing *both* is difficult for human drivers without any driver-assistance system. This means that the infrastructure does not only have to support communication, but should also provide sophisticated driver-assistance techniques. The design and the validation of such an environment is not trivial.

Software agents are a key technology for the infrastructure envisioned. Intersection agents and what we call driver-assistance agents can negotiate the time to cross an intersection in advance. Recent proposals already feature agent-based intersection control, to reduce average waiting time or other target variables [5, 16]. These approaches yield good results. However, though the authors expect positive environmental effects, they have not investigated them systematically.

The contribution of this article is twofold. First, we investigate the effects of existing agent-based mechanisms for intersection control on fuel consumption. We show that these mechanisms reduce fuel consumption by up to 28% compared to traffic lights (TL). This is a significant reduction given that city traffic requires crossing intersections frequently. Second, we propose a novel agent-based mechanism for intersection control with minimization of fuel consumption as an explicit design objective. The reduction is between 22% and 26% compared to TL. This is significant as well, but less than what we had expected, in the light of the first contribution. We further show that our new mechanism reduces average waiting time in certain situations by up to 98% compared to TL and is better than the existing approaches of [5] and [16]. Summing up, our study shows that agent-based mechanisms for intersection control may result in a reduction of fuel consumption that is substantial.

Paper outline: We discuss related work in Section 2. Then, we describe agent-based intersection control in Section 3. In Section 4, we present our estimation model for fuel consumption. We introduce the various mechanisms for intersection control in Section 5. We evaluate these mechanisms in Section 6 and conclude in Section 7.

2. RELATED WORK

This section reviews related work on intersection control whose purpose is to reduce fuel consumption. We start with simple approaches which are already used in the real world, like roundabouts with and without traffic lights, and continue with more complex ones. Finally, we review agent-based approaches on intersection control.

[18] shows that roundabouts reduce fuel consumption by 28%, by avoiding waiting time during off-peak hours. On the other side, the waiting time for some vehicles increases during peak hours. This problem is addressed in [2], by additional usage of traffic lights during peak hours. This ensures that vehicles coming from directions with little traffic do not have to wait too long. In this case the signals are red when the vehicle queue in one direction reaches the queue detector. This creates a gap in the circulation flow.

Another approach which does not need any construction changes of the intersection is introduced in [11]. There, the cycle length is optimized by minimization of a performance index. This index does not only take into account the delay and the number of stops but also the fuel consumption. Orthogonally to our approach, [11] examines the optimal cycle length based on the traffic density and traffic volume. It is however determined a priori and does not change with new vehicles arriving. Our approach in turn determines dynamically which vehicle should cross the intersection next, based on the current state.

A more advanced way to optimize/synchronize the signal settings is to use real-time video-traffic monitoring. [13] suggests to use color-image sequences combined with a definition of search windows around areas of interest. This allows to anticipate the arrival of vehicles at an intersection and gives way to adaptive and predictive traffic-light control. A high-level traffic-light controller can use these images to reduce waiting time and fuel consumption.

[8] combines the real-time video-traffic monitoring with induction loops and a multi agent control system. Every intersection is controlled by an autonomous agent, which communicates with adjacent agents. Vehicle queues represent each incoming intersection lane. When a vehicle leaves the intersection, the adjacent intersection agent in the direction of the vehicle is informed about the probability that the vehicle will arrive there. In this way, the intersection agent can identify the best traffic-light phase possible.

3. AGENT-BASED INTERSECTION CONTROL

The mechanisms discussed in this paper use agent technology. It lets intersections and vehicles negotiate the time slot when to cross an intersection, and vehicles can adapt their speed autonomously when approaching an intersection. As a prerequisite, vehicles are equipped with an additional control unit, subsequently referred to as *driver-assistance system*. Further, intersections have a traffic-control unit, referred to as *intersection-control unit*. These control units consist of hardware and of software components.

Driver-assistance systems and intersection-control units have to communicate. To this end, they use *intersection agents*, which represent intersection-control units, and *driver-assistance agents*, which represent driver-assistance systems. These agents are a software component of the respective control unit.

While driving, a driver-assistance system can recommend a certain speed to the driver. If the driver does not overrule the driver-assistance system, it may also directly control the driving behavior of the vehicle [17] (Figure 1).

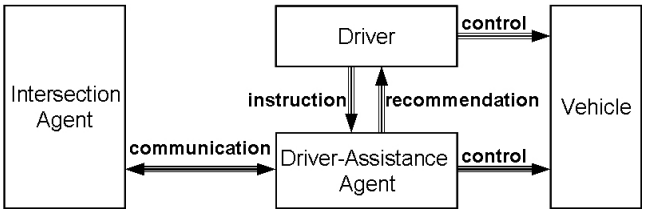


Figure 1: Agent-based traffic control

The driver-assistance system can be seen as an extension of an adaptive cruise control (ACC) system, which is state of the art in vehicles. ACC systems assist the driver to keep a certain distance to vehicles in front. He does not have to react when vehicles in front accelerate or decelerate. This is done by the ACC system. In addition, the driver-assistance system described here also adjusts the speed to reach the intersection at a certain time. [17] calls such a system adaptive cruise and crossing control (A3C) system.

If vehicles are equipped with driver-assistance agents, we can design mechanisms for intersection-control where driver-assistance agents and intersection agents negotiate the right to cross an intersection. A *time slot* is the right to cross an intersection in a certain direction within a certain period of time. Each driver-assistance agent tries to obtain its *next*

free time slot, i. e., the earliest slot which the vehicle can still reach in time, and which the intersection can assign to the vehicle. Vehicles typically have different next free slots.

The intersection agent is responsible for the allocation of a time slot. Because vehicles can cross an intersection concurrently, the allocation of time slots follows certain rules. There already exist various allocation rules like 'priority to right', 'four-way-stop' or 'preference road'. For agent-based intersection control, new allocation rules are possible. [17] proposes four different allocation rules, with a distinction on the degree of concurrency allowed. To formulate these rules in a clean way, we use the following terminology: An intersection consists of several *intersection lanes*. If two intersection lanes share common space, we say that they are *conflicting*. We call the shared space *conflict area*. If two intersection lanes emerge from one incoming lane, the conflict area is *diverging*.

With *intersection exclusive*, the intersection agent allows one vehicle to enter the intersection after all other vehicles have left it. With *lane exclusive*, a vehicle may enter the intersection only when all vehicles on the desired lane and on all conflicting lanes have left the intersection. *Lane shared* lets a vehicle enter the intersection if there are no more vehicles on other conflicting intersection lanes. However, a vehicle may enter the intersection while other vehicles cross the intersection on conflicting intersection lanes with diverging conflict areas. *Conflict-area exclusive* only blocks the conflict areas of an intersection. Vehicles may cross the intersection concurrently as long as not more than one vehicle is in each conflict area. Clearly, the possible throughput increases from intersection exclusive to conflict-area exclusive. Because lane shared is already state of the art, we only consider lane shared and conflict-area exclusive in what follows.

These degrees of concurrency are particularly meaningful in the context of agent-based intersection control. In principle, it would be possible to build traffic lights as strict as intersection or lane exclusive. However, traffic lights usually allow vehicles to cross an intersection in a way similarly to lane shared. Several vehicles can enter the intersection from the same lane while they have green light. It is not possible to use standard traffic lights for conflict-area exclusive. This is because conflict-area exclusive switches between vehicles from different directions too quickly.

4. MODELS TO ESTIMATE FUEL CONSUMPTION

To consider the fuel consumption of vehicles approaching an intersection, we need an estimation model. In this section we describe the model used here in detail.

4.1 Existing Models

Various models have been proposed in order to estimate the fuel consumption of vehicles. These models can be categorized based on the parameters used to estimate the fuel consumption. For example, *average speed models* are based on the average speed of a vehicle [4]. In contrast, *nonlinear regression models* distinguish between acceleration and deceleration phases of a drive [1]. Similarly to the nonlinear regression models, *modal models* split a trip into four driving modes: idle, acceleration, deceleration and cruising mode [4, 10]. The focus of these models is on the strict distinction between the driving modes. However, they do not specify explicitly the way to determine the fuel consumption within a mode. *Energy-based models* take the energy demand of a vehicle while driving as the basis for estimating the fuel consumption [4, 12, 14, 15].

We have analyzed different models to estimate fuel consumption. We have compared the necessary degree of detail and the availability of calibration data. As a result, the Instantaneous Model [4], which is both an energy-based and a modal model, has turned out to be most suitable within an intersection-control mechanism.

4.2 Instantaneous Model

The Instantaneous Model [4] determines the fuel consumption based on the energy demand of a vehicle. In order to compute the energy demand it uses the instantaneous speed v (in m/s) and acceleration a (in m/s^2). In this way, it reflects the different situations within a drive and is able to provide a very accurate prediction of the fuel consumption of an individual vehicle.

Using the Instantaneous Model, we can determine the fuel consumption of a vehicle by the following equation:

$$F = \begin{cases} \alpha + \beta_1 R_{tract} v + [\beta_2 a R_{inertial} v]_{a>0} & \text{for } R_{tract} > 0 \\ \alpha & \text{for } R_{tract} \leq 0 \end{cases}$$

where F is the fuel consumption in ml/s. This formula combines three different fuel-demand types:

Idle.

The fuel consumption which is needed just to run the engine is the *idle fuel consumption* α of a vehicle (in ml/s).

Movement.

The additional fuel consumption for the movement at constant speed is the product of the *efficiency parameter* β_1 (in ml/J) and the *tractive energy demand* $R_{tract} \cdot v$. R_{tract} denotes the *tractive force*. If it is not positive, the movement causes no additional fuel consumption.

Acceleration.

The additional fuel consumption of an accelerating vehicle is the product of the *efficiency parameter* β_2 (in $ml/(J \cdot m/s^2)$) and the *inertial energy demand* $a \cdot R_{inertial} \cdot v$. $R_{inertial}$ denotes the *inertial force*. If the acceleration is not positive, no additional fuel consumption has to be taken into account.

The tractive force R_{tract} is the sum of *drag force* R_{drag}, *inertial force* $R_{inertial}$ and *grade force* R_{grade}. R_{drag} comprises *rolling resistance* $R_{rolling}$ and *air drag force* R_{air}: $R_{drag} = R_{rolling} + R_{air}$. The *inertial force* $R_{inertial}$ is the product of *vehicle mass* m (in kg) and acceleration a:

$$R_{inertial} = m \cdot a$$

Grade force combines *gravitational acceleration* ($g = 9.81 \frac{m}{s^2}$), vehicle mass and *road grade* G (in %):

$$R_{grade} = m \cdot g \cdot G$$

4.3 Refinement of the Instantaneous Model

In [4], average values, calibrated on the basis of a certain vehicle fleet, are used for idle fuel consumption α, air drag force R_{air} and rolling resistance $R_{rolling}$. These average values are not very accurate, because they only are aggregate

values of a certain test fleet. Therefore, we compute the actual values for each vehicle like speed, frontal area etc. from the data available instead of using average values.

Idle fuel consumption.

The idle fuel consumption α (in ml/s) of a vehicle can be derived from its *engine capacity* V_h (in l) [9]:

$$\alpha = \frac{0.220}{10^3 s} V_h - \frac{0.0193}{10^3 s\, l} V_h^2$$

For trucks, we always use $\alpha = 0.7ml/s$ as idle fuel consumption [3].

Air drag force.

The air drag force R_{air} is based on *air density* ρ (in kg/m^3), *drag coefficient* C_D, *frontal area* A (in m^2) and instantaneous speed v of a vehicle:

$$R_{air} = 0.5 \cdot \rho \cdot C_D \cdot A \cdot v^2$$

Air density relates to air temperature and to the height above sea level. To keep things manageable, the temperature is assumed to be 15°C and the height above sea level $200\,m$ [9]. According to [9], this results in an air density of $\rho = 1.2\,kg/m^3$. The drag coefficient as well as the frontal area of a vehicle can be determined relatively easily, because they are often stated in the specification of a vehicle. If the values are not included in the specification at least the frontal area can be derived for passenger cars from maximum height h and maximum width w of the vehicle as follows [9]:

$$A = 0.9 \cdot h \cdot w$$

Rolling resistance.

The computation of the rolling resistance is intricate because it is based on properties like road surface and tires used. Because this data is very hard to obtain, an average value, calibrated in [4], is used:

$$R_{rolling} = 333\,N$$

5. MECHANISMS

In this section we present different mechanisms for intersection control. First, we describe the mechanism *Traffic Light* (TL). Then, we describe *Time-Slot Request* (TSR) which allocates the next free time slot to cross an intersection to the first driver-assistance agent which requests a time slot from the intersection agent. Thereafter, we present ITSA Valuation which allocates the next free time slot to the vehicle with the highest valuation of reduced waiting time. Then, we introduce a new environment-aware mechanism ITSA Fuel Consumption. It allocates the next free time slot to the vehicle which causes the minimal total increase of fuel consumption. Finally, we describe ITSA Delay as a variation of ITSA Fuel Consumption.

5.1 Traffic Light

Traffic lights (TL) are one of the most common intersection-control mechanisms. Therefore, TL serves as our yardstick for the environment-aware ITSA Fuel Consumption.

Using TL the green light phases are computed in advance based on the expected traffic volume. For TL we use a static traffic-light mechanism. There also are dynamic mechanisms which adapt the duration of the green light phases according to the current traffic volume. Because the expected volume does not change within a run of our evaluation, a static mechanism is adequate. Note that our evaluation in turn will cover different volumes of traffic.

The duration of a traffic-light phase depends on the expected traffic volume. To determine the adequate duration of such a phase, we use the AKF Schema [6]. It considers the traffic flows from all incoming to outgoing lanes. The AKF Schema considers traffic flows which are in conflict with each other and therefore have to pass the intersection in sequence. For example, the vehicles driving on the left incoming lane turning left are in conflict with vehicles from the opposite direction going straight and cannot pass the intersection at the same time. But if vehicles can go straight on several lanes of a direction, the traffic lights of these lanes have to be synchronized.

The so-called AKF Matrix is based on the conflicting traffic flows. Each column contains the expected traffic volumes of traffic flows which are in conflict. The values in every column are added up, and the maximum column sum is determined. For the intersection evaluated, the maximum column sum and, consequently, the traffic volume of the critical traffic flows at a traffic density of 50 vehicles/hour on every lane is 400. These values let us compute the time of circulation and, consequently, the lengths of single phase durations. The *time of circulation* is the time between two green phases of the same direction. It depends on the volumes of the conflicting traffic flows, the saturation-traffic volume, the minimum duration of a green light phase and the time between the green light phases for two different directions, called buffer time t_z.

The buffer time combines intersection-crossing time t_{cr} (in seconds), intersection-clearance time t_{cl} (in s) and intersection-entering time t_e (in s): $t_z = t_{cr} + t_{cl} - t_e$ This equation shows that t_z is based on the crossing distance and that it depends on the driving direction. To determine the time of circulation of the traffic light the maximum value of t_z is chosen and decomposed into a yellow phase (typically 2-3 s), a yellow-red phase (typically 2-3 s) and a red phase for all directions (typically 1-2 s).

Using the value of t_z just determined, the time of circulation t_u is computed according to the following equation given in [6]:

$$t_u = \frac{\sum_i t_z + \sum_i t_{min}}{1 - \frac{Q_{max}}{Q_s}}$$

where i is the number of conflicting traffic flows, t_z is the time between the end of the green phase for one direction and the begin of the green phase for another direction, t_{min} is the minimum duration of a green phase (10 s per conflicting direction, according to [6]), Q_{max} is the traffic volume of the conflicting traffic flows, which pass the intersection (in vehicles/hour), and Q_s is the saturation-traffic volume, which describes the expected number of vehicles being able to pass the intersection in all directions in one hour of green phases (2000 vehicles/hour).

Using the prior values the circulation time for vehicles going straight is

$$t_z = 3\,s + 5.6\,s - 2.25\,s = 6.35\,s \approx 7\,s$$

$$t_u = \frac{4 \cdot 7\,s + 4 \cdot 10\,s}{1 - \frac{400}{2000}} = 85\,s$$

This results in a green phase duration of $10\,s + \frac{(85\,s - 4 \cdot 7\,s - 4 \cdot 10\,s)}{4} = 14\,s$ and a red phase duration of $85\,s - 14\,s = 71\,s$.

5.2 Time-Slot Request

[5] has proposed a mechanism which uses agent technology for intersection control. [16] describes an extension of it dubbed Time-Slot Request (TSR). With TSR, the intersection agent allocates the next free time slot to the first driver-assistance agent which requests such a slot. In other words, TSR uses a first-in first-out scheme to allocate slots. [5] has shown that a system which uses such a scheme can outperform traffic lights regarding average waiting time. Note that waiting time is different from standstill time because we define waiting time as the difference of travel time and minimal travel time [17]. [5] does not evaluate environmental measures. We will show that TSR reduces fuel consumption compared to TL.

5.3 ITSA Valuation

The main idea of ITSA Valuation is to allocate the next free time slot to the vehicle whose driver has the highest valuation of reduced waiting time [16]. ITSA stands for Initial Time-Slot Auction. It uses auctions to allocate the next free slot to vehicles. With ITSA, a vehicle, once it has received a slot, cannot trade it for another one.

ITSA Valuation executes two algorithms concurrently. Algorithm 1 describes how driver-assistance agents contact the intersection agent. Algorithm 2 shows how the intersection agent chooses the driver-assistance agent to assign the next slot.

Algorithm 1 (Contact Step).

1. Driver-assistance agents whose vehicles approach the intersection request time slot from intersection agent
2. Intersection agent adds vehicle to virtual queue which represents its incoming lane
3. Intersection agent confirms request but does not provide time slot immediately

The first vehicle in each queue which has not received a time slot so far is called *candidate*. Candidates (from different lanes) are the only vehicles which can receive the next free time slot. The intersection agent executes allocations rounds continuously, to allocate time slots to candidates (Algorithm 2). In each allocation round, one candidate receives a time slot.

Algorithm 2 (Allocation Round).

1. Intersection agent calls all vehicles currently queued for bids
2. Vehicles reveal their valuation per second of reduced waiting time, their current speed and distance to the intersection
3. Intersection agent computes the queue with maximal sum of valuations and assigns time slot to the candidate of the respective queue
4. Intersection agent removes candidate from the virtual queue

While ITSA Valuation has been designed with the purpose of reducing the average valuation-weighted waiting time [16] we will show that it also curbs fuel consumption.

5.4 ITSA Fuel Consumption

The main idea of the novel environment-aware mechanism ITSA Fuel Consumption is to consider the estimated fuel consumption of each vehicle. To do so, the mechanism chooses the vehicle whose intersection crossing results in the minimum additional fuel consumption for all vehicles close to the intersection. ITSA Fuel Consumption uses the same protocol as ITSA Valuation. In contrast to ITSA Valuation, vehicles do not have to report their valuation of reduced waiting time. Instead, the intersection agent considers the influence of the allocation of the next free time slot to each candidate in each allocation round. I. e., the intersection agent computes the increase of fuel consumption induced by each allocation possible.

An allocation of a time slot typically delays other vehicles. The delay d_j^k is the time Vehicle j has to wait longer if the intersection agent allocates the next free slot to Vehicle k. Thus, the delay d_j^k is the difference between the next free slot of Vehicle j after an allocation to Vehicle k and the next free slot of Vehicle j before the allocation.

Example 1. *Let the next free time slots of Vehicles j and k be $t_j = 20s$ and $t_k = 22s$. Suppose that the intersection agent allocates its next free time slot to Vehicle k. Further, suppose that this changes the next free time slot of Vehicle j to $t_j^* = 26s$. Then, the delay is $d_j^k = t_j^* - t_j = 26s - 20s = 6s$.*

Now suppose that Vehicle j and k can cross the intersection concurrently because the lanes used are non-conflicting, an allocation to Vehicle k does not change the next free slot of Vehicle j. Thus, the delay is $d_j^k = 0$.

Note that vehicles waiting behind a candidate are not delayed if the intersection agent allocates the next free time slot to 'their' candidate.

A delay of a vehicle increases its fuel consumption. In many cases it has to decelerate and accelerate. For each candidate, the intersection agent computes and accumulates the increase of fuel consumption of all other vehicles. To do so, it uses the estimation model from Section 4.3.

Finally, the intersection agent compares the increase of fuel consumption for all allocations possible and allocates the next free time slot in the best way. Like with ITSA Valuation, the vehicle waiting behind the former candidate becomes a new candidate, and the intersection agent initiates a new allocation round.

5.5 ITSA Delay

ITSA Fuel Consumption is rather complex because it needs detailed information about each vehicle approaching. Therefore, we propose ITSA Delay as a variant of ITSA Fuel Consumption. ITSA Delay needs less information because it does not compute the increase of total fuel consumption but the increase of total waiting time. It computes the increase of total waiting time for all allocations possible and allocates the next free time slot in the best way.

6. EVALUATION

To evaluate all intersection-control mechanisms discussed, we use a home-grown simulation framework. It allows the

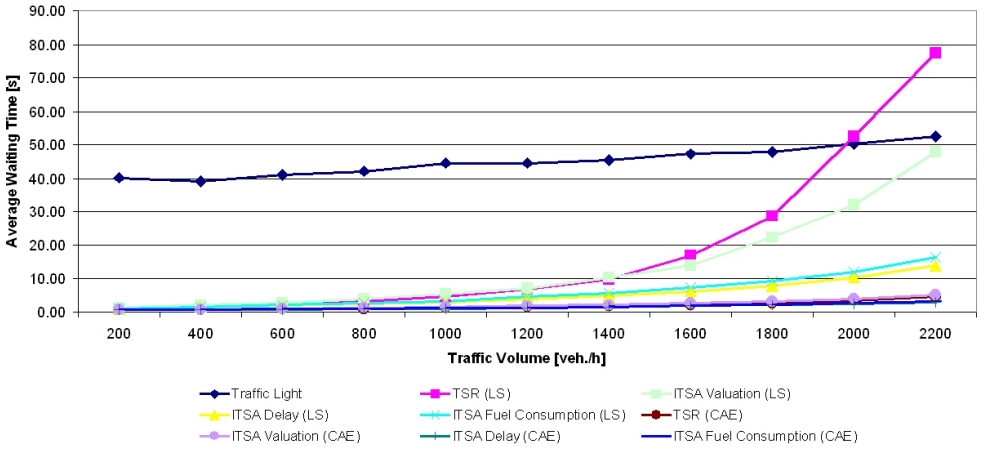


Figure 2: Average Waiting Time

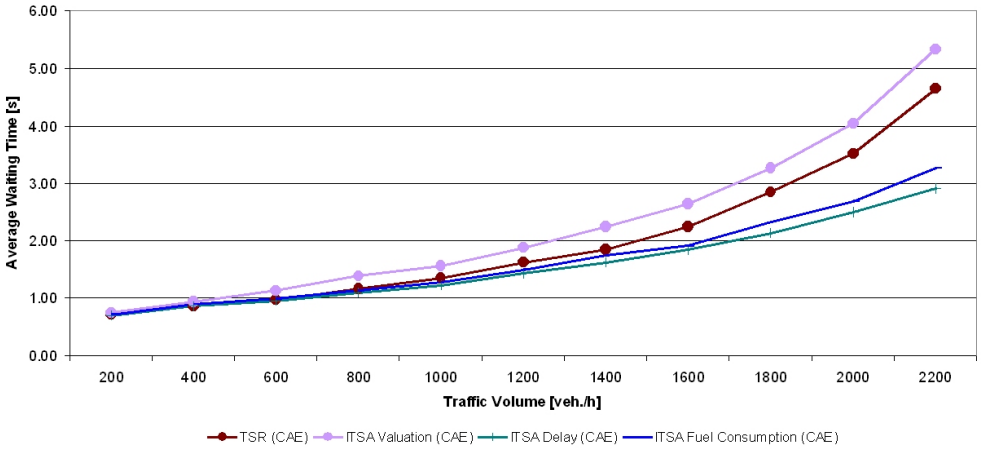


Figure 3: Average Waiting Time (CAE only)

simulation of traffic at an intersection. In the simulation, agent-based driver-assistance systems interact with agent-based intersection-control units. The behavior of vehicles and drivers is simulated.

6.1 Experimental Setup

For the evaluation we use a symmetric intersection consisting of four directions. Each direction has two incoming and two outgoing lanes. For each direction one incoming lane (right) allows to turn right and to go straight, and the other incoming lane (left) allows to turn left and to go straight.

To analyze the impact of traffic volume, every mechanism is evaluated with traffic volumes between 25 vehicles/hour and 275 vehicles/hour on every lane (in 25 vehicles/hour steps) respectively between 200 vehicles/hour and 2200 vehicles/hour in total. We assume the traffic volume to be exponentially distributed with the desired traffic volume as average. Each vehicle goes straight or turns right respectively left with equal probability. The maximum speed on the lanes is 50 km/h. The one on the intersection is 45 km/h.

Our simulation is space-continuous and time-discrete. We simulate 23 minutes in each simulation run. In the first three minutes, vehicles fill the intersection, and we only consider the vehicles of the last 20 minutes to avoid startup effects. The simulation consists of several stochastic components like interarrival times, valuations of reduced waiting time, or route choice. We use a seed which configures the stochastic components of a simulation run. To alleviate the influence of this seed, we always execute five simulation runs using the same five seeds (which of course are different) for each setting. While different seeds lead to a different simulation behavior, the average values remain the same for each setting. This allows us a pairwise comparison of simulation runs of different settings. We always compare simulations runs with the same seed. I. e., we compare only simulation runs with the same stochastic behavior.

6.2 Experiments

We use the same setting to evaluate the average waiting time and the average fuel consumption of the following intersection-control mechanisms. Next to Traffic Light we evaluate TSR, ITSA Valuation, ITSA Fuel Consumption and ITSA Delay for the two degrees of concurrency *lane shared* (LS) and *conflict-area exclusive* (CAE).

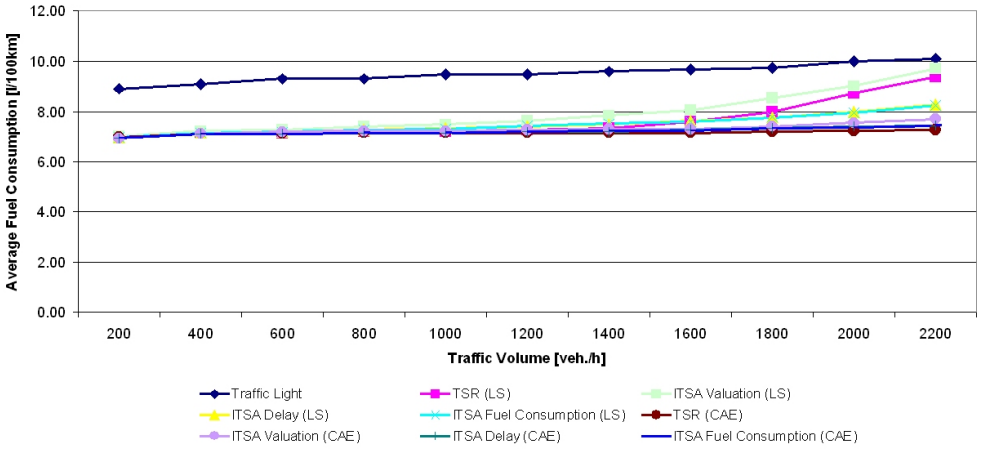


Figure 4: Average Fuel Consumption

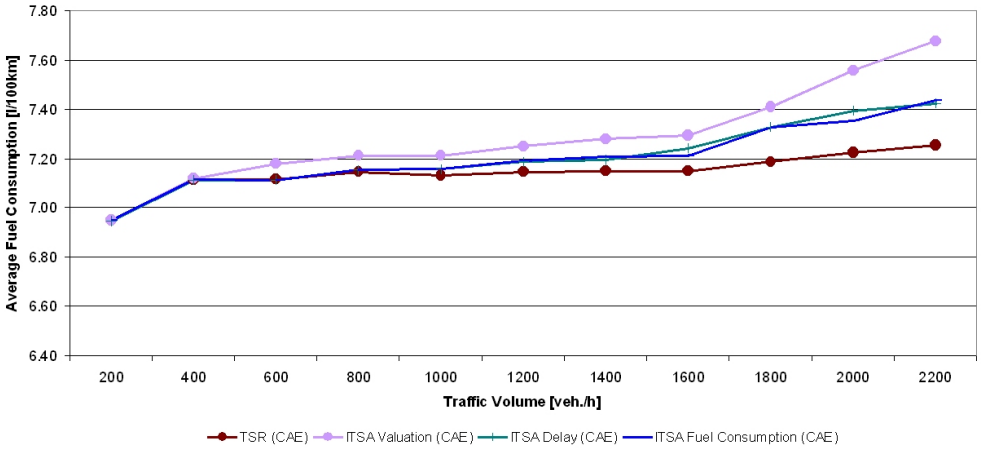


Figure 5: Average Fuel Consumption (CAE only)

6.2.1 *Waiting Time*

Figure 2 describes the average waiting time of all nine evaluated mechanisms: Traffic Light, TSR (LS), ITSA Valuation (LS), ITSA Delay (LS), ITSA Fuel Consumption (LS), TSR (CAE), ITSA Valuation (CAE), ITSA Delay (CAE), ITSA Fuel Consumption (CAE). The results of the mechanisms which use CAE are very similar. Thus, Figure 2 does not allow to distinguish the results of the mechanisms for CAE. Therefore, Figure 3 describes the average waiting time of these mechanisms separately. The results show that all mechanisms outperform Traffic Light for all traffic volumes evaluated, except for TSR (LS) and ITSA Valuation (LS) regarding average waiting time. TSR (LS) reduces the average waiting time up to 1800 vehicles/hour and ITSA Valuation (LS) up to 2000 vehicles/hour significantly.

As an example we list some average values and the 95% confidence intervals for 2000 vehicles/hour in detail: The average waiting time is 50.36 s [48.03, 52.69] for Traffic Light, 4.04 s [3.37, 4.70] for ITSA Valuation (CAE), 3.52 s [2.83, 4.21] for TSR (CAE), 2.69 s [2.28, 3.09] for ITSA Fuel Consumption (CAE), and 2.49 s [2.09, 2.90] for ITSA Delay (CAE). I. e., ITSA Delay (CAE) is slightly but not significantly better than ITSA Fuel Consumption (CAE). For 2000 vehicles/hour the relative reduction of the average waiting time compared to Traffic Light is 95% for ITSA Fuel Consumption (CAE) and ITSA Delay (CAE).

6.2.2 *Fuel Consumption*

Figure 4 describes the average fuel consumption of all mechanisms evaluated. It does not allow to distinguish the results of the mechanisms for CAE. Therefore, Figure 5 describes the average fuel consumption of these mechanisms separately. All mechanisms outperform TL significantly regarding fuel consumption.

For 2000 vehicles/hour, the average fuel consumption is 9.98 l/100 km [9.75, 10.20] for Traffic Light, 7.56 l/100 km [7.42, 7.69] for ITSA Valuation (CAE), 7.39 l/100 km [7.28, 7.51] ITSA Delay (CAE), 7.36 l/100 km [7.21, 7.50] for ITSA Fuel Consumption (CAE), and 7.22 l/100 km [7.13, 7.32] for TSR (CAE). I. e., TSR (CAE) is slightly better than ITSA Fuel Consumption (CAE) and ITSA Fuel Consumption (CAE) is slightly better than ITSA Delay (CAE). But in both cases the difference is not significant. For 2000 vehicles/hour the relative reduction of the average fuel consumption compared to Traffic Light is 26% for ITSA De-

lay (CAE), 26% for ITSA Fuel Consumption (CAE), and 28% for TSR (CAE).

6.2.3 Conclusion

Taking the results both for average waiting time and fuel consumption into account we come to the following conclusions: TL performs worse than any other evaluated mechanism in almost any case. The reduction of waiting time and fuel consumption is considerable, e. g., for 2000 vehicles/hour up to 95% respectively up to 28%.

As expected, all mechanisms for conflict-area exclusive outperform the ones for lane-shared significantly. ITSA Delay and ITSA Fuel Consumption lead to very similar results. ITSA Delay is slightly better regarding average waiting time, ITSA Fuel Consumption is slightly better regarding average fuel consumption. ITSA Valuation and TSR perform always worse than ITSA Delay and ITSA Fuel Consumption except for average fuel consumption using TSR (CAE). In this case TSR (CAE) leads to the best results.

Given our evaluation, we recommend to use ITSA Delay if one is interested in average waiting time and fuel consumption. ITSA Delay is always best regarding the average waiting time and nearly as good as ITSA Fuel Consumption. Further, ITSA Delay needs no detailed information about the actual vehicle type and can be computed more easily than ITSA Fuel Consumption.

7. SUMMARY

Intersections are a main bottleneck in vehicle traffic. Traffic causes pollution and fuel consumption. Existing mechanisms for intersection control optimize throughput and waiting time but not fuel consumption. To deal with this issue, we have designed a novel, agent-based mechanism for intersection control. We compare it both to traffic lights and to other mechanisms. For the comparison, we deploy a sophisticated estimation model for fuel consumption.

We show that agent-based intersection-control mechanisms outperform traffic lights both regarding waiting time and fuel consumption. This even holds for mechanisms which have not been designed with the explicit intention of reducing fuel consumption. Compared to traffic lights, ITSA Fuel Consumption (CAE) reduces fuel consumption by between 22% and 26%. ITSA Delay (CAE) reduces waiting time by between 94% and 98%. This is a substantial reduction.

Our mechanisms can be adapted to other objectives. Given appropriate estimation models, we can readily come up with mechanisms which aim to reduce other environmental target variables, e. g., CO^2 emissions or vehicle noise.

8. REFERENCES

[1] K. Ahn. Microscopic fuel consumption and emission modeling. Master's thesis, Faculty of the Virginia Polytechnic Institute and State University, 1998.

[2] R. Akcelik. Operating cost, fuel consumption and pollutant emission savings at a roundabout with metering signals. In *22nd ARRB conference: research into practice*, 2006.

[3] R. Akcelik and M. Besley. Operating cost, fuel consumption, and emission models in aasidra and aamotion. In *25th Conference of Australian Institutes of Transport Research (CAITR)*, 2004.

[4] D. P. Bowyer, R. Akçelik, and D. C. Biggs. Guide to fuel consumption analysis for urban traffic management. Special Report SR 32, ARRB Transport Research Ltd, Vermont South, Australia, 1985.

[5] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Artificial Intelligence Research*, 31:591–656, 2008.

[6] A. W. Gleue. A Simplified Method to Compute Signal-Controlled Intersections (In German). *Straßenbau und Straßenverkehrstechnik*, Volume 136, 1972.

[7] Green Vehicle Guide - An Australian Government Initiative. Fuel Consumption Label. `http://www.greenvehicleguide.gov.au`, 2010. Last accessed: 22/10/2010.

[8] D. Greenwood, B. Burdiliak, I. Trencansky, H. Armbruster, and C. Dannegger. Greenwave distributed traffic intersection control. In *AAMAS*, 2009.

[9] I. D. Greenwood. *A New Approach To Estimate Congestion Impacts For Highway Evaluation - Effects On Fuel Consumption And Vehicle Emissions*. PhD thesis, The University of Auckland, 2003.

[10] W.-T. Hung, H.-Y. Tong, and C.-S. Cheung. A modal approach to vehicular emissions and fuel consumption model development. *Journal of the Air & Waste Management Association*, 55(10), 2005.

[11] R. Ikeda, H. Kawashima, and T. Oda. Determination of traffic signal settings for minimizing fuel consumption. In *International Conference on Intelligent Transportation Systems*, 1999.

[12] D. Leung and D. Williams. Modelling of motor vehicle fuel consumption and emissions using a power-based model. *Environmental Monitoring and Assessment*, 65(1-2):21–29, 2000.

[13] P. Payeur and Y. Liu. Video traffic monitoring for flow optimization and pollution reduction. In *IEEE/AEI International Workshop on Advanced Environmental Monitoring Technologies*, 2003.

[14] K. Post, J. H. Kent, J. Tomlin, and N. Carruthers. Fuel consumption and emission modelling by power demand and a comparison with other models. *Transportation Research Part A*, 18(3):191–213, 1984.

[15] M. Ross. Automobile fuel consumption and emissions: Effects of vehicle and driving characteristics. *Annual Reviews of Energy and Environment*, 19:75–112, 1994.

[16] H. Schepperle and K. Böhm. Auction-Based Traffic Management: Towards Effective Concurrent Utilization of Road Intersections. In *IEEE Joint Conference on E-Commerce Technology (CEC'08) and Enterprise Computing, E-Commerce and E-Services (EEE'08)*, pages 105–112, 2008.

[17] H. Schepperle and K. Böhm. *Handbook of Research on Multi-Agent Systems for Traffic and Transportation Engineering*, chapter Valuation-Aware Traffic Control – the Notion and the Issues. Information Science Reference, 2009.

[18] A. Várhelyi. The effects of small roundabouts on emissions and fuel consumption: a case study. *Transportation Research Part D: Transport and Environment*, 7(1):65 – 71, 2002.

Decentralized coordination of plug-in hybrid vehicles for imbalance reduction in a Smart Grid

Stijn Vandael
Nelis Boucké, Tom Holvoet
DistriNet, Department of Computer Science
Katholieke Universiteit Leuven, Belgium
{firstname.lastname}@cs.kuleuven.be

Klaas De Craemer
Geert Deconinck
ELECTA, Department of Electrical Engineering
Katholieke Universiteit Leuven, Belgium
{firstname.lastname}@esat.kuleuven.be

ABSTRACT

Intelligent electricity grids, or 'Smart Grids', are being introduced at a rapid pace. Smart grids allow the management of new distributed power generators such as solar panels and wind turbines, and innovative power consumers such as plug-in hybrid vehicles. One challenge in Smart Grids is to fulfill consumer demands while avoiding infrastructure overloads. Another challenge is to reduce imbalance costs: after ahead scheduling of production and consumption (the so-called 'load schedule'), unpredictable changes in production and consumption yield a cost for repairing this balance.

To cope with these risks and costs, we propose a decentralized, multi-agent system solution for coordinated charging of PHEVs in a Smart Grid. Essentially, the MAS utilizes an "intention graph" for expressing the flexibility of a fleet of PHEVs. Based on this flexibility, charging of PHEVs can be rescheduled in real-time to reduce imbalances.

We discuss and evaluate two scheduling strategies for reducing imbalance costs: reactive scheduling and proactive scheduling. Simulations show that reactive scheduling is able to reduce imbalance costs by 14%, while proactive scheduling yields the highest imbalance cost reduction of 44%.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Coherence and coordination, Multiagent systems; J.7 [**Computer Applications**]: Industrial control

General Terms

Algorithms, Economics, Experimentation

Keywords

Multi-agent systems, plug-in hybrid vehicles, Smart Grids.

1. INTRODUCTION

In recent years, there is a global evolution in the way energy is generated and consumed due to climate change, energy independence and the impending decay of fossil fuels. In Europe, these changes are reflected in the 20-20-20 targets: 20% carbon reduction, 20% rise in energy efficiency and 20% production from renewables [1] by 2020. At the present, two major evolutions are already visible.

The first evolution is the explosive growth in the amount of small distributed generators (DG) connected to the local distribution grid (e.g solar panels). By nature, this type of renewable, dispersed electricity generation is unpredictable and uncontrollable.

The second evolution is the increasing amount of PHEVs, hybrid vehicles with a battery that can be charged through a regular power socket. Recent research predicts that in 2030, PHEVs will comprise 5% of the Belgian electricity consumption [2]. Because of this large impact of PHEVs on the electricity infrastructure, controlled charging of PHEVs is an important research topic. Apart from a challenge, PHEVs offer a tremendous opportunity for managing fluctuations caused by distributed generation.

Intelligent electricity grids or Smart Grids enable the management of such advanced production and consumption in the electricity grid. In a Smart Grid, it becomes possible to intelligently coordinate consumers to maintain the net balance and ensure an efficient, reliable and environmentally friendly production, transmission and distribution of electricity.

Multi-agent systems have been identified by the IEEE Power Engineering Society's Multi-Agent Systems (MAS) Working Group as a promising distributed control approach in power engineering [3, 4]. The working group identified the following key benefits of applying MAS in power engineering:
- *Flexibility*: the ability to respond to dynamic situations.
- *Extensibility*: the ability to easily add new functionality and augmenting or upgrading existing functionality.
- *Fault tolerance*: the ability of the system to meet its design objectives in case of failure.

In this paper, a decentralized solution based on MAS is proposed, discussed and evaluated for coordination of the charging of PHEVs to reduce imbalances caused by DG. The paper contributes to this research in three ways:

1. Assessment of the increasing imbalance costs due to renewables and the potential of PHEVs as a means to reduce these costs. (section 2)
2. Description of a multi-agent solution for large-scale coordination of PHEV charging and the explanation of different scheduling strategies to reduce imbalance costs. (section 3)
3. Evaluation through simulation of the multi-agent solution in scenarios with PHEVs and solar panels. (section 4)

Cite as: Decentralized coordination of plug-in hybrid vehicles for imbalance reduction in a Smart Grid, Stijn Vandael, Klaas De Craemer, Nelis Boucké, Tom Holvoet and Geert Deconinck, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 803-810.

2. BALANCE MANAGEMENT IN THE ELECTRICAL GRID

The unpredictability of renewable DG incurs a risk for traders on the electricity market, called the "imbalance cost". Especially day-ahead markets, where a load schedule has to be predicted 12-36 hours in advance, pose a serious problem. An example are wind farms: even with state of the art forecasting methods, the short-term electricity generation of wind farms cannot be predicted with a high degree of accuracy [5].

At the same time, recent research suggests that PHEVs will comprise 5% of the national electricity consumption [2]. Because cars are parked most of the day, opportunities arise for shifting the charging of PHEVs in time. This way the imbalance caused by unpredictable generation can be offset, while ensuring that PHEVs are charged in time, i.e. before their intended departure.

The management of the balance between production and consumption in electrical grids entails a complex engineering domain. In this section, we aim to identify the key elements and procedures in this domain that are required to clearly define the problem and motivate the solution.

2.1 TSO responsibilities

The electrical grid consists of a transmission grid and a distribution grid. The transmission grid transfers electricity from large power plants to the distribution grid, while the distribution grid distributes electricity to individual households, factories and street lighting. In each country, the transmission grid is maintained by a transmission system operator (TSO) and the distribution grid by one or more distribution system operators (DSO). While the responsibilities of the DSO are mostly infrastructural and administrative, one of the main tasks of the TSO is to constantly monitor and maintain the balance between supply and demand within its control area.

To balance between supply and demand, the TSO needs predictions of the energy that will be injected and withdrawn at each access points to its transmission grid. Each access point has a designated BRP (Balancing Responsible Party). This BRP provides the TSO with a predicted load schedule of the consumers and/or producers behind its respective access point. Based on these load schedules, the TSO manages electricity flows between the access points and the overall balance between production and consumption in its control area.

2.2 BRP responsibilities

The load schedule of a BRP is organized in fixed settlement periods. The length of a settlement period varies per country, but is typically 15 minutes (e.g Belgium and the Netherlands), 30 minutes (e.g England and Wales) or 1 hour (e.g Sweden and Norway). Load schedules submitted to the TSO must be balanced. This means that if a BRP has declared a scheduled supply to another BRP, the reverse transfer of energy must be found in the schedule of this other BRP [6] or in the import/export schedule to another control area.

BRPs need to provide their load schedule before a fixed deadline, called the "gate closure". Most European countries utilize a day-ahead gate closure. For example, in Italy, the gate closure is at 16h00 day-ahead for all settlement periods of the next day (from 00h00 until 24h00). After gate closure, the BRP's load schedule cannot be changed anymore.[1]

2.3 Imbalance cost

During a settlement period, the TSO continually balances supply and demand, taking into account finite network capacity. If there is insufficient supply to meet demand, the TSO dispatches extra supply reserves and vice versa. The costs (demand reserve) or revenues (supply reserve) for dispatching these reserves are settled with the BRPs causing the imbalance. An BRP with negative imbalance (more consumption or less production than planned) pays an imbalance tariff to the TSO, while an BRP with a positive imbalance (less consumption or more production than planned) gets paid an imbalance tariff.[2]

From an BRP's point of view, it is more profitable to sell its production and buy its consumption on the day-ahead market, because the imbalance tariffs for extra consumption are typically high and for extra production low. These lost revenues for an BRP are called the "imbalance price". This imbalance price is the difference between the before price (day-ahead tariff) and the after price (imbalance tariff). In economics, this is called an opportunity cost. The total imbalance cost in each settlement period is calculated as the difference between the metered energy volume with the contracted energy volume, multiplied by the imbalance price:

$$\text{Cost}_{imbalance} = (E_{measured} - E_{contracted}) \cdot \text{Price}_{imbalance}$$

Obviously, it is a challenge for BRPs to accurately predict their load schedules. A BRP responsible for an access point to a local distribution grid consisting of households, typically predicts its load schedule based on synthetic load profiles. For example, in Belgium, the local electricity regulator provides these profiles for every day of the year before the beginning of the year. Examples of a few different load profiles are depicted in figure 1.

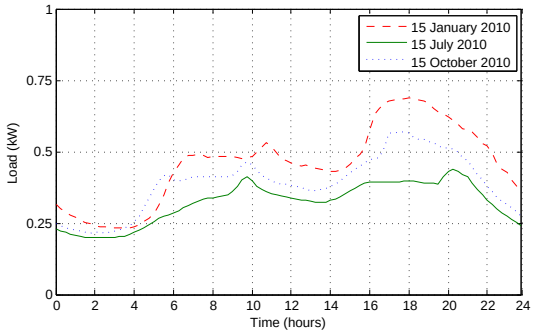


Figure 1: SLPs (synthetic load profiles)

[1]The time interval between the gate closure and the actual start of the corresponding period of operation varies between countries. The gate closure can be within the same day (intraday) or in the previous day (day-ahead) of the period of operation. For example, gate closure in Denmark is half an hour ahead (intraday), in the Netherlands one hour ahead (intraday) and in Italy at 16h00 day-ahead [7]. For intraday, the period of operation is one settlement period and for day-ahead, the period of operation is one day (from 00h00 until 24h00).

[2]In extreme cases, for example, when there is a huge overproduction from renewables, an BRP gets paid for consuming electricity.

3. A MULTI-AGENT SYSTEM SOLUTION FOR IMBALANCE REDUCTION

Supported by the conclusions of the IEEE Power Engineering Society's Multi-Agent Systems (MAS) Working Group [3, 4], as well as by our own experience [8], we target a decentralized, multi-agent system solution for the coordinated charging of PHEVs to reduce imbalances. This solution focuses on the actors and interactions aimed at mitigating the imbalance after gate closure. We assume that before gate closure, the load schedule with predictions of households and distributed generators was assembled by the BRP.

The schematic overview of the multi-agent system is depicted in figure 2. A PHEV agent represents the software managing the charging of a PHEV, a transformer agent controls a low-voltage transformer and the BRP agent manages the access point to the transmission grid. Each type of agents has the following primary goals:

- PHEV agent: *charge the battery of its PHEV in time.*
- Transformer agent: *prevent overloading of its transformer.*
- BRP agent: *minimize imbalance costs.*

These goals are not independent from each other. For example, a PHEV with an empty battery cannot be charged in an hour, because this would cause overloading the low voltage transformer and most likely cause imbalance; or a BRP cannot reduce a negative imbalance when PHEVs are about to leave and still need to be fully charged. The agents need to coordinate with each other to meet the individual goals of all agents.

3.1 Coordination mechanism

The agents are organized in a hierarchical structure (figure 3) and their basic coordination mechanism consist of four steps:

1. The PHEV agents send their charge intentions to the connecting transformer agents. Through aggregation of these charge intentions at each transformer agent, the BRP can assemble an intention graph of all PHEVs in the distribution grid.
2. The BRP agent decides how much energy will be charged in the next time step according to a suitable scheduling strategy (see section 3.2, "scheduling strategies").
3. The BRP agent informs the transformer agents about the energy that will be charged in the next time step. Accordingly, the transformer agents divide this energy between their underlying PHEVs.
4. The PHEV agents start charging the accepted amount of energy.

This coordination mechanism is executed at a frequency dependent on the required adaptiveness of the considered scenario. Initialization of the sequence is done by sending a global synchronization signal from the BRP down to all PHEVs.

The intention graph expresses the intentions of all PHEVs and enables the BRP to estimate the total flexibility of its PHEVs. In figure 4, the working of the intention graph is depicted:

(A) In this figure, an intention graph is depicted for two PHEVs at a given moment in time. The time-scale is divided into time intervals of a quarter hour, while the Y-axis indicates the amount of energy. As indicated in the figure, PHEV A will leave after the second quarter, while PHEV B will leave after the third quarter. Each of the PHEVs still needs 1 kWh of charging energy before they leave.

(B) In order to reduce imbalances (section 3.2), the BRP decides to fully charge PHEV A and half of PHEV B in the first quarter. Accordingly, PHEV A will charge for 1 kWh in the first quarter (= 4 kW), while PHEV B will charge for 0.5 kWh in the first quarter (= 2 kW).

(C) After the first quarter, PHEV A is fully charged and PHEV B still needs to be charged for 0.5 kWh.

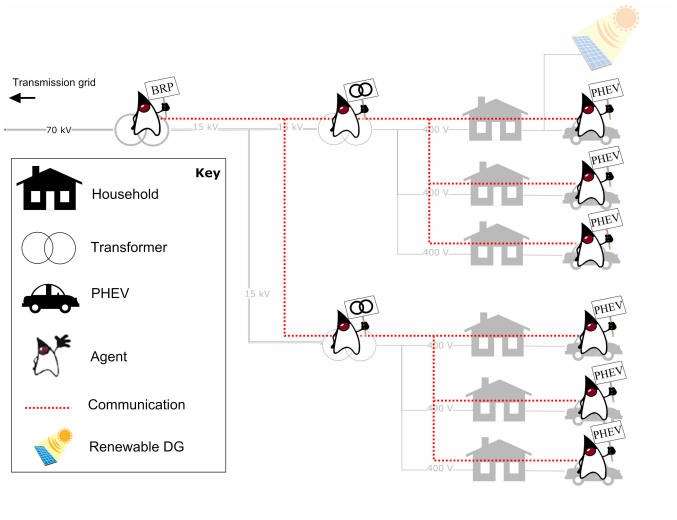


Figure 2: Schematic overview of the MAS.

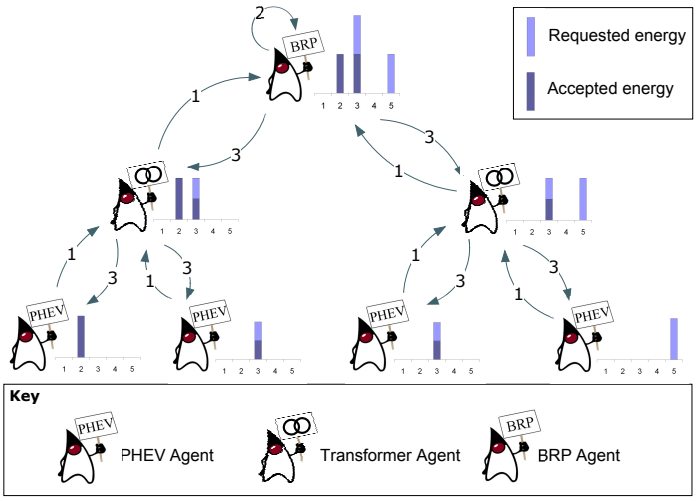


Figure 3: The MAS coordination mechanism.

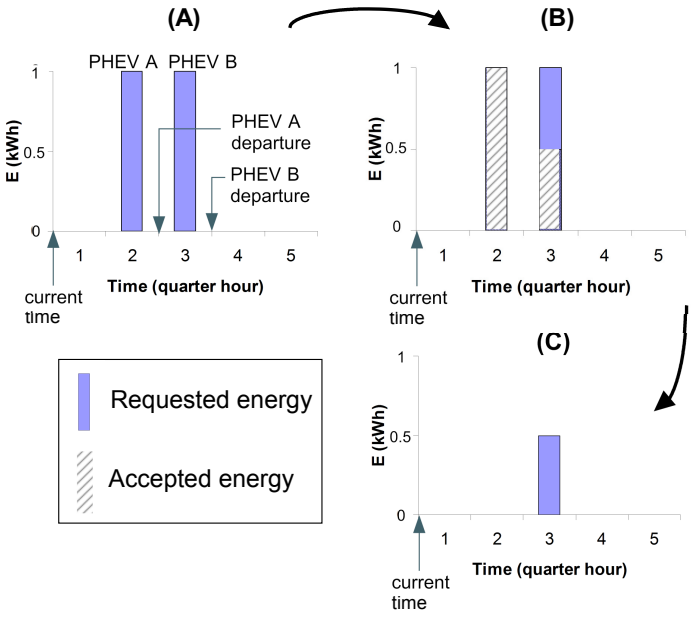


Figure 4: Representation of PHEV intentions.

3.2 Scheduling strategies

The BRP uses a scheduling strategy to achieve its goals. These goals have a strict order, which means that one goal cannot be achieved without achieving the previous goal. In order of importance:

1. Transformer and cable limits
   To avoid infrastructure damage, the transformer and cables have a power limit that cannot be overstepped. For that purpose, the agents send their current and maximum load towards the BRP agent (step 1 in the coordination mechanism). In each strategy, this constraint is integrated. In the rest of the explanation, this constraint is assumed, without repeated mentioning.

2. Charging of PHEVs
   To ensure that PHEVs' owners can fully benefit from their electric car, PHEVs are charged before they depart. The intention graph incorporates this goal.

3. Minimal imbalance costs
   When infrastructure limits are respected and PHEVs can be fully charged, load can be shifted in order to minimize imbalance costs in the BRPs perimeter. This will be the focus of the proposed strategies.

All scheduling strategies presented in this paper are explained with the small example depicted in figure 5. In this example, the BRP agent has to schedule the charging of 10 kWh in five settlement periods of 15 minutes. For this purpose, the BRP agent uses a day-ahead load schedule and a real-time schedule of the five settlement periods.

The **day-ahead schedule** consist of the sum of the predictions of non-PHEV load (households and DG) and PHEV load. This schedule was submitted to the TSO before gate closure (day-ahead) and doesn't change during the operation period.

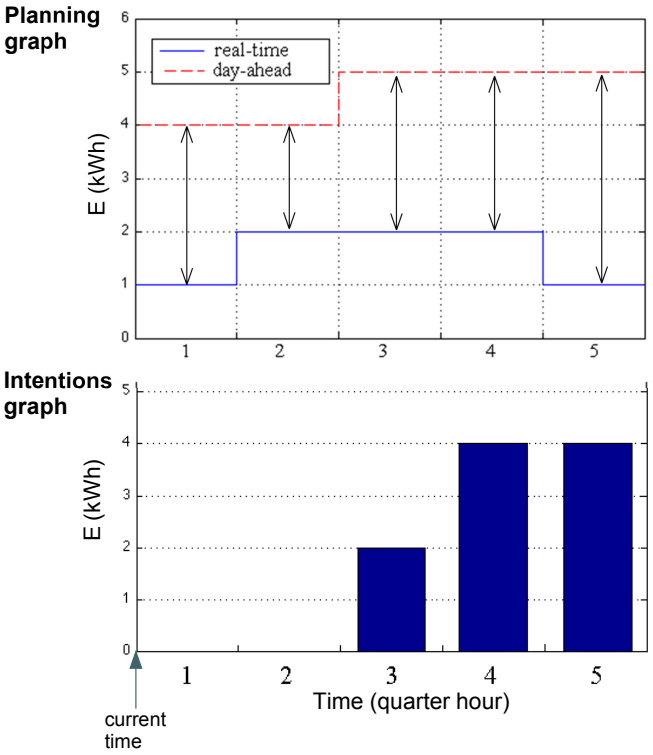


Figure 5: Scheduling example.

The **real-time schedule** only consists of the predictions of the non-PHEV load (households and DG). PHEV load is not included in this schedule, because the real-time schedule is used to online schedule the PHEV load on top.

The BRP schedules the charging of PHEVs onto the real-time schedule to approach the day-ahead schedule as closely as possible to reduce imbalance costs. While we assume that the real-time schedule doesn't change in this small example, this schedule can be updated with new information about non-PHEV loads that become available. For example, new weather information or load measurement data.

3.2.1 *Reactive strategy*

The reactive scheduling strategy is a strategy where imbalances are postponed as long as possible. Figure 6 shows the result of this strategy on the considered example. The amount of energy (10 kWh) is scheduled in order to meet the balancing requirements in the first three quarters. However, the imbalance is expected to increase from the fourth quarter due to a PHEV charging shortage. In case of a surplus of PHEV charging, reservations are made at the end of the scheduling to postpone any imbalances. Although the portfolio balancing strategy is reactive, PHEVs are ensured to fully charge their battery before departure, given that the transformer load constraints are respected. The PHEV intentions are always reserved in ascending order of departure time to ensure maximum utilization of flexibility.

Advantage: The portfolio is balanced as long as possible.

Disadvantage: The risk of a large future imbalance is great. When high imbalance costs coincide with this large imbalance, total imbalance costs will be high.

Algorithm 1: Reactive scheduling

```
PHEVEnergyLeft = sum(intentions)
for T:  1 to  endTime do
    while prediction(T) < dayahead(T)
         && energyLeft > 0 do
      PHEVEnergyLeft = reserve(T, PHEVEnergyLeft)
    end while
end for
for T in range(endTime, 1) do
      PHEVEnergyLeft = reserve(T, PHEVEnergyLeft)
end for
```

3.2.2 *Proactive strategy*

The proactive strategy is a strategy where imbalances are equally distributed among the schedule. Figure 7 shows the result of this strategy on the considered example. The amount of energy (10 kWh) is scheduled in order to minimize the average distance between the prediction and load schedule. Again, to ensure maximum flexibility in the future, the PHEVs were reserved in the order of their departure time. Note that the imbalance is the same as in the previous strategy, but the imbalance risk is divided over all timesteps. For example, in figure 7, when a large amount of PHEVs connects to the grid after quarter 3, it is possible that consumption becomes too high. In that case, the reactive strategy would be better.

Advantage: The risk for high imbalance costs is divided over the schedule.

Disadvantage: This strategy assumes a good prediction without constant changes.

Algorithm 2: Proactive scheduling

```
PHEVEnergyLeft = sum(intentions)
while PHEVEnergyLeft > 0  do
  if dayahead - prediction > 0 do
    T = timeOfLargestImbalance()
  else
    T = timeOfSmallestImbalance()
  end if
  PHEVEnergyLeft = reserve(T, PHEVEnergyLeft)
end while
```

4. SIMULATION EXPERIMENT: BALANCING SOLAR POWER

4.1 Experiment description

In this experiment, the proposed multi-agent system and its strategies are evaluated and compared for the reduction of imbalances caused by solar panels. The considered scenario is a future situation of a residential area with solar panels and PHEVs.

The scenario contains 200 households with consumption profiles obtained from the Belgian distribution grid provider Infrax [9]. These profiles contain actual measured household consumption on a 15 minute base.

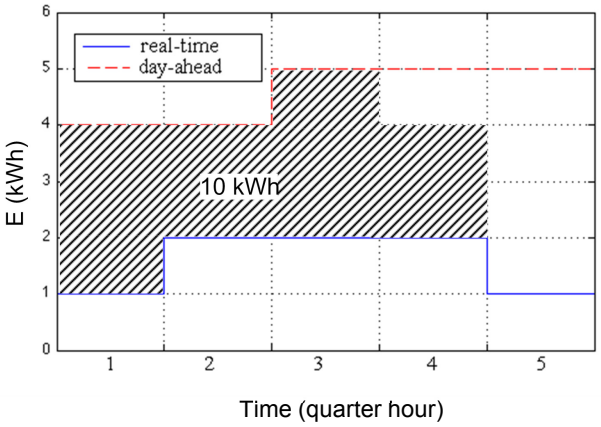


Figure 6: Reactive strategy.

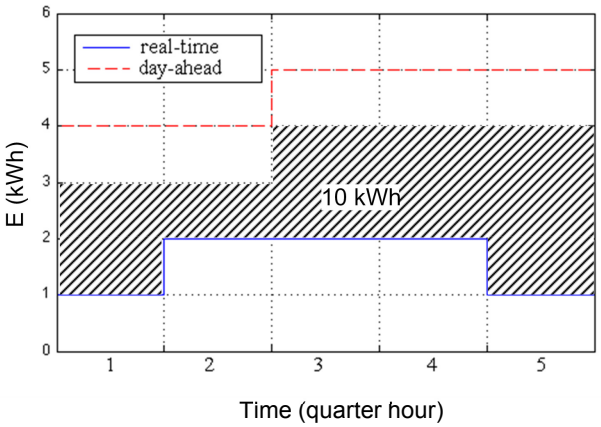


Figure 7: Proactive strategy.

From 200 households, 64 households have solar panels installed. Again, profiles were obtained from the Belgian distribution grid provider Infrax [9] from actual measured data.

For a true representation of the load caused by PHEVs, a realistic model of PHEV usage is utilized [10]. This model represents the state of a car (home, driving ...) on a per minute base. Furthermore, the Chevrolet Volt is chosen, which is expected to go in production at the end of 2010. In our simulations, we suppose that 50% of the vehicles are able to charge at a charging station during the day.

Day-ahead load schedule

The day-ahead load schedule consists of predictions for households, solar panels and PHEVs (figure 8). For household predictions, synthetic load profiles were used from the Flemish Regulation Entity for the Electricity and Gas market (VREG) [11].

For the production from PV (photovoltaic) panels, the solar output trend can be predicted, but not the short-term variations (due to moving clouds, shadow casting etc.). Accordingly, predictions for PV panels were made by applying a moving average filter (of 15 quarter hour samples) on the actual data (figure 9).

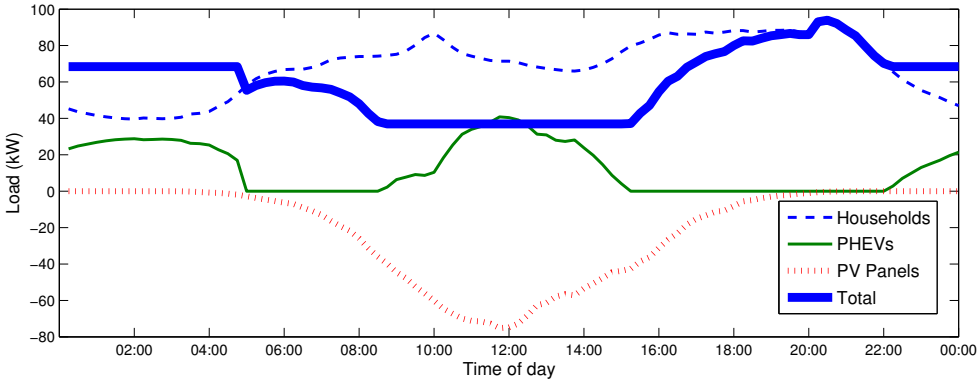


Figure 8: Day-ahead schedule

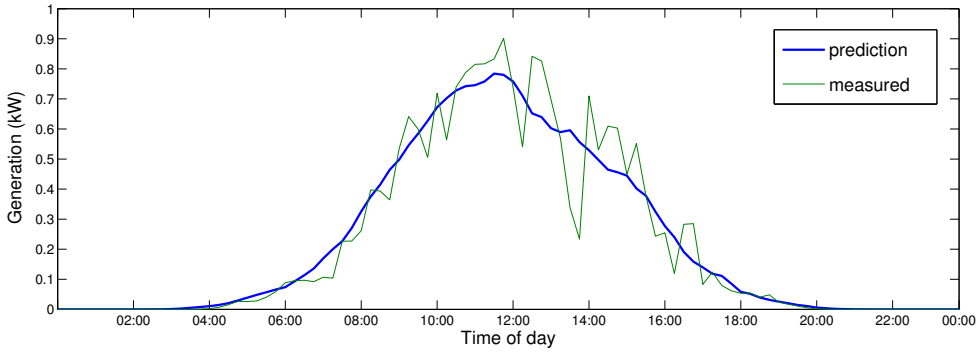


Figure 9: Power prediction of one PV panel

Although PHEVs will be coordinated, their expected load also has to be included in the day-ahead prediction. 50% of the vehicles are only able to charge at home, while 50% of the vehicles also have access to a daytime charging station. Therefore, half of the expected PHEV load (calculated by their battery content) is allocated during the night, while 50% of the PHEV load is allocated during the business hours. During the night, electricity on the Belpex day-ahead market[3] is generally cheaper, which amounts to cheaper electricity for the BRP. During business hours, solar production is highest, which makes charging PHEVs at those moments essential for balancing. Charging PHEVs during evening peak hours, when the household load is high, must be avoided at all costs to prevent overloading the infrastructure and paying high prices on the Belpex day-ahead market.

Imbalance cost

The imbalance cost is an opportunity cost, caused by buying or selling energy at an imbalance price instead of placing correct bids on the day-ahead market. The imbalance cost is calculated by using the day-ahead price (provided per hour by the Belgian day-ahead market Belpex) and the imbalance prices (provided per quarter hour by the Belgian TSO Elia).

[3]http://www.belpex.be

4.2 Simulation results

For simulating the described scenario, we built an open-source multi-agent simulator [12]. Simulations show that the reactive strategy is able to lower imbalance costs with 14%, while the proactive strategy is able to lower imbalance costs by 44%. The load imbalances for a typical simulation run using the active and proactive strategy show the reason for this difference (figure 10).

Between 10h00 and 13h30, a positive imbalance is visible for both strategies. This positive imbalance indicates a lower off-take than expected. The reason is that the solar panels are producing more than expected during these periods (figure 9), while the limited amount of PHEVs (figure 8) is unable to charge more to compensate for the overproduction.

The reactive strategy maintains the instantaneous balance, while ignoring possible balancing problems in the future. Accordingly, the active strategy immediately starts fully charging its PHEVs at 10h00 to compensate for the overproduction. The disadvantage is that the cars are fully charged by 12h30 and a high imbalance from 12h30 until 13h30 is unavoidable. During this high imbalance, the TSO was dispatching extra demand reserves, which leads to a high imbalance price for production. In contrary, the proactive strategy was able to avoid these high costs by spreading the risk over the total imbalance period.

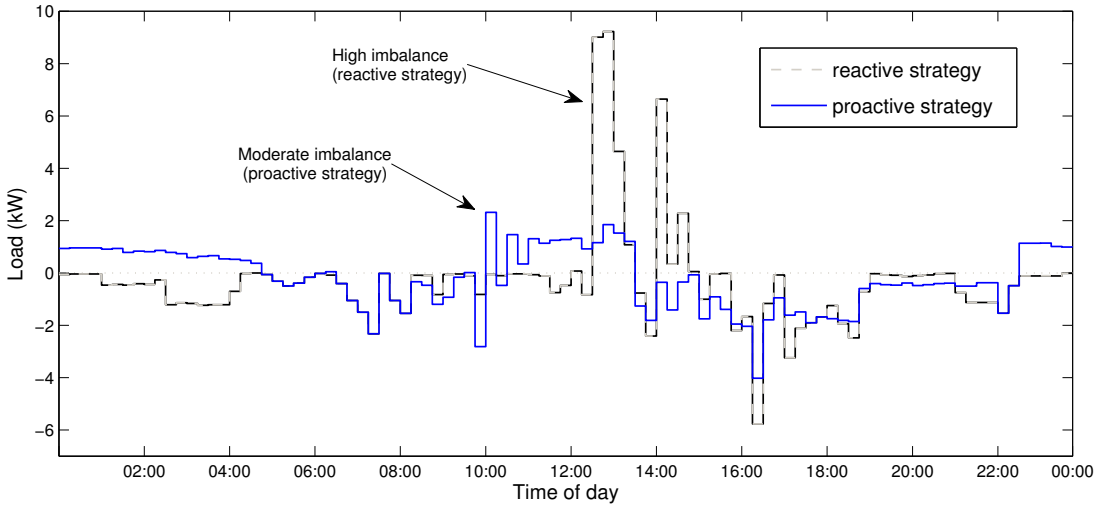


Figure 10: Load imbalance

5. RELATED WORK

In several research studies, multi-agent systems have been identified as the key technology in the future Smart Grid. Examples of MAS applications in Smart Grids range from island-mode control [13], micro-storage management [14] and micro grids [15] to market-based control [16, 17].

In [16], a novel market-based mechanism and trading strategies are proposed for a Smart Grid. In this mechanism, unforeseen demand or increased supply (not traded on the day-ahead market) are coped with by real-time trading between the actors (presented as agents) in the electricity market. The market mechanism proposed in this paper complements with our balancing mechanism in the sense that our balancing mechanism balances within the jurisdiction of one trader, while the mechanism of Vytelingum et. al balances between different traders through a market. Furthermore, because different traders are located on the transmission grid, the market mechanism includes congestion management by pricing the flow of electricity.

In [18], multi-agent coalitions for electrical vehicles are described for participation of these vehicles in the power regulation market. In the regulation market, electrical vehicles are used to provide both regulation-up power and regulation-down power. In this paper, regulation-up power was also provided by V2G (vehicle-to-grid), where vehicles are discharged onto the grid. Kamboj et. al modelled the coalition formation problem and presented various coalition formation strategies. The point of view of this paper is from the TSO's perspective. While vehicles in our paper are used for mitigating balancing cost of an BRP, Kamboj. et al actually deploys vehicles as reserve capacities for the TSO.

The PowerMatcher [19] is a market-based control concept for supply and demand matching in electricity networks. The basic MAS architecture of the PowerMatcher is a tree-structure similar to the one proposed in this paper. In the PowerMatcher, agents buy (consumers) and sell (producers) electricity on an electronic market by using a 'bid function'. This bid function expresses to what degree an agent is willing to pay (consumer) or be paid (producer) for a certain amount of electricity. By matching all these bid functions, the equilibrium price is determined to match demand and supply in a PowerMatcher cluster.

One of the field tests where the PowerMatcher was evaluated, is in the reduction of imbalance caused by trading of wind power on the APX (Amsterdam Power Exchange), by expanding an electricity trader's wind portfolio with flexible sources of demand and supply [20, 21]. For this purpose, a programme agent was included in the multi-agent system to push the market outcome to the programme value (the day-ahead load schedule). While our proposed MAS and the PowerMatcher are both used for reducing imbalance costs, the approaches are fundamentally different. While the PowerMatcher balances according to the degree an agent is willing to pay, our MAS balances according to the charging intentions of PHEVs. The contribution of the PowerMatcher is that a price component is explicitly integrated to incentivize consumers and producers, while our contribution is that PHEVs are assured to be charged before a certain time. Furthermore, while the PowerMatcher only represents short-term flexibility (expressed in Power), our mechanism is able to express long-term flexibility (expressed in Energy).

6. CONCLUSION AND FUTURE WORK

In the future, the coordinated charging of PHEVs will offer opportunities to mitigate imbalance costs. Due to the large scale and dynamic nature of the coordination problem, multi-agent systems are a promising technology in this area. The multi-agent system presented in this article uses an extendable, flexible and scalable technique for expressing PHEV intentions and controlling their charging behavior. Two scheduling strategies were proposed: reactive scheduling and proactive scheduling.

The presented simulation case shows that the MAS is capable of coordinating PHEVs to cope with unpredictable solar generation. Imbalance costs were decreased with 14-44%. Simulations showed that in most cases the reactive strategy was outperformed by the proactive strategy due to the great risks of a concentrated imbalance.

Future work will include the following aspects:

SCENARIOS. To more thoroughly evaluate solutions for balancing with PHEVs, more scenarios need to be tested. An important example is the integration of unpredictable wind power generation. Furthermore, the scenario considered in this paper does not necessarily hold for each region. For example, city regions will have different characteristics compared to rural regions.

SCALABILITY. The local communication and simple aggregation of intention graphs in the proposed MAS suggest a good scalability in terms of communication and execution time. However, this quality should be evaluated explicitly. In previous work [8], the demand-side management of PHEVs was evaluated against a reference solution based on quadratic programming. The same comparison techniques will be used for evaluation of the MAS in this article.

7. REFERENCES

[1] European Commission. Eu action against climate change, December 2008.

[2] K. Clement, E. Haesen, and J. Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on Power Systems*, 25(1):371–380, February 2010.

[3] S. D. J. Mcarthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi. Multi-agent systems for power engineering applications - part i: Concepts, approaches, and technical challenges. *IEEE Transactions on Power Systems*, 22(4):1743–1752, 2007.

[4] S. D. J. Mcarthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi. Multi-agent systems for power engineering applications - part ii: Technologies, standards, and tools for building multi-agent systems. *IEEE Transactions on Power Systems*, 22(4):1753–1759, 2007.

[5] Tilak Thakur, Sunita Goyal, Jaimala Gambhir, and Ishpreet Kaur. Optimisation of imbalance cost for wind power marketibility using hydrogen storage. In *2008 Joint International Conference on Power System Technology and IEEE Power India Conference*, pages 1–5. IEEE, October 2008.

[6] Leonardo Meeus, Konrad Purchala, and Ronnie Belmans. Development of the internal electricity market in europe. *The Electricity Journal*, 18(6):25–35, July 2005.

[7] SETSO Sub Group Balance Management. Current state of balance management in south east europe. Technical report, ETSO, June 2006.

[8] Stijn Vandael, Nelis Boucké, Tom Holvoet, and Geert Deconinck. Decentralized demand side management of plug-in hybrid vehicles in a smart grid. In *Proc. 1st Int. Workshop on Agent Technologies for Energy Systems (ATES-2010), co-located with 9th Int. Conf. on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 67–74, May 2010.

[9] Infrax. `http://www.infrax.be`, 2010.

[10] Eric De Caluwé. Potentieel van demand side management, piekvermogen en netondersteunende diensten geleverd door plug-in hybrid elektrische voertuigen op basis van een beschikbaarheidsanalyse. Master's thesis, KULeuven, 2008.

[11] Synthetic load profiles. [online available]: `http://www.vreg.be/nl/06_sector/02_leveranciers/03_voorschriften/03_elektriciteitsprofiel.asp`, 2010.

[12] Stijn Vandael. A simulator for the smart electric grid. `http://stijn.ulyssis.be/SmartGridSimulator/`, 2011.

[13] M. Pipattanasomporn, H. Feroze, and S. Rahman. Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pages 1–8, March 2009.

[14] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based micro-storage management for the smart grid. In *Autonomous Agents And MultiAgent Systems (AAMAS 2010), Toronto, Canada.*, 2010.

[15] Aris Dimeas and Nikos D. Hatziargyriou. A multi-agent system for microgrids. In *SETN*, pages 447–455, 2004.

[16] P. Vytelingum, S. D. Ramchurn, T. D. Voice, A. Rogers, and N. R. Jennings. Trading agents for the smart electricity grid. Autonomous Agents And MultiAgent Systems (AAMAS 2010), Toronto, Canad, 14th-18th May 2010.

[17] K. Kok, C. Warmer, R. Kamphuis, P. Mellstrand, and R. Gustavsson. Distributed control in the electricity infrastructure. In *Future Power Systems, 2005 International Conference on Future Power Systems*, pages 7 pp.–7, Nov. 2005.

[18] Sachin Kamboj, Keith Decker, Keith Trnka, Nathaniel Pearre, Colin Kern, and Willett Kempton. Exploring the formation of electric vehicle coalitions for vehicle-to-grid power regulation. In *Proc. 1st Int. Workshop on Agent Technologies for Energy Systems (ATES-2010), co-located with 9th Int. Conf. on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 67–74, May 2010.

[19] J. K. Kok, C. J. Warmer, and I. G. Kamphuis. Powermatcher: multiagent control in the electricity infrastructure. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 75–82, New York, NY, USA, 2005. ACM.

[20] R. Kamphuis, F. Kuijper, C. Warmer, M. Hommelberg, and Koen Kok. Software agents for matching of power supply and demand: a field-test with a real-time automated imbalance reduction system. pages 7 pp. –7, nov. 2005.

[21] M.P.F. Hommelberg, C.J. Warmer, I.G. Kamphuis, J.K. Kok, and G.J. Schaeffer. Distributed control concepts using multi-agent technology and automatic markets: An indispensable feature of smart power grids. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –7, 24-28 2007.

Online Mechanism Design for Electric Vehicle Charging

Enrico H. Gerding* eg@ecs.soton.ac.uk
Valentin Robu* vr2@ecs.soton.ac.uk
Sebastian Stein* ss2@ecs.soton.ac.uk
David C. Parkes† parkes@eecs.harvard.edu
Alex Rogers* acr@ecs.soton.ac.uk
Nicholas R. Jennings* nrj@ecs.soton.ac.uk

*University of Southampton, SO17 1BJ, Southampton, UK
†Harvard University, Cambridge, MA 02138, USA

ABSTRACT

Plug-in hybrid electric vehicles are expected to place a considerable strain on local electricity distribution networks, requiring charging to be coordinated in order to accommodate capacity constraints. We design a novel online auction protocol for this problem, wherein vehicle owners use agents to bid for power and also state time windows in which a vehicle is available for charging. This is a multi-dimensional mechanism design domain, with owners having non-increasing marginal valuations for each subsequent unit of electricity. In our design, we couple a greedy allocation algorithm with the occasional "burning" of allocated power, leaving it unallocated, in order to adjust an allocation and achieve monotonicity and thus truthfulness. We consider two variations: burning at each time step or on-departure. Both mechanisms are evaluated in depth, using data from a real-world trial of electric vehicles in the UK to simulate system dynamics and valuations. The mechanisms provide higher allocative efficiency than a fixed price system, are almost competitive with a standard scheduling heuristic which assumes non-strategic agents, and can sustain a substantially larger number of vehicles at the same per-owner fuel cost saving than a simple random scheme.

Categories and Subject Descriptors

I.2.11 [**AI**]: Distributed AI - multiagent systems

General Terms

Algorithms, Design, Economics

Keywords

electric vehicle, mechanism design, pricing

1. INTRODUCTION

Promoting the use of electric vehicles (EVs) is a key element in many countries' initiatives to transition to a low carbon economy [4]. Recent years have seen rapid innovation within the automotive industry [10], with designs such as plug-in hybrid vehicles (PHEVs, which have both an electric motor and an internal combustion engine) and range-extended electric vehicles (which have an electric motor and an on-board generator driven by an internal combustion engine) promising to overcome consumers' *range anxiety*[1] and thereby increasing mainstream EV use (the Toyota 'plug-in' Prius and the Chevrolet Volt are commercial examples of both, which will be on the road in 2011). However, there are significant concerns within the electricity distribution industries regarding the widespread use of such vehicles, since the high charging rates that these vehicles require (up to three times the maximum current demand of a typical home) could overload local electricity distribution networks at peak times [5]. Indeed, street-level transformers servicing between 10-200 homes may become significant bottlenecks in the widespread adoption of EVs [11].

To address these concerns, electricity distribution companies that are already seeing significant EV use (such as the Pacific Gas and Electric Company in California) have introduced time-of-use pricing plans for electric vehicle charging that attempt to dissuade owners from charging their vehicles at peak times, when the local electricity distribution network is already close to capacity[2]. While such approaches are easily understood by customers, they fail to fully account for the constraints on the local distribution networks, and they are necessarily static since they require that vehicle owners individually respond to this price signal and adapt their behaviour (i.e., manually changing the time at which they charge their vehicle). Looking further ahead, researchers have also begun to investigate the automatic scheduling of EV charging. Typically, this work allows individual vehicle owners to indicate the times at which the car will be available for charging, allowing automatic scheduling while satisfying the constraints of the distribution network [15, 2]. However, since these approaches separate the scheduling of the charging from the price paid for the electricity (typically assuming a fixed per unit price plan), they are unable to preclude the incentive to misreport (e.g., an owner may indicate an earlier departure time or further travel distances in order to receive preferential charging).

To address the above shortcomings, we turn to the field of *online mechanism design* [12]. Specifically, we focus on mechanisms that are *model-free* (which make no assumptions about future demand and supply of electricity), and that allocate resources as they become available (electricity is *perishable* since installing alternative storage capacity can be very costly). Now, existing mechanisms of this kind assume that the preferences of the agents (representing the vehicle owners) can be described by a single parameter, so-called *single-valued* domains. However, this assumption is not appropriate for our problem, where agents have multi-unit demand with marginal non-increasing valuations for incremental kilowatt hours (kWh) of electricity.[3] To this end, we extend the state of the art in

[1]Fear that a car will run out of electricity in the middle of nowhere.

Cite as: Online Mechanism Design for Electric Vehicle Charging, E.H. Gerding, V. Robu, S. Stein, D.C. Parkes, A. Rogers and N.R. Jennings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 811-818.

[2]See for example `www.pge.com/about/environment/pge/electricvehicles/fuelrates/`.

[3]Marginal valuations are non-increasing in our domain because distance and energy usage are uncertain, and therefore the first few units of electricity are more likely to be used, and (in the case of plug-in hybrid electric vehicles) any shortfall can be made up by using the vehicle's internal combustion engine.

dynamic mechanism design as follows:

- We develop a formal framework and solution for the EV charging problem, and show that it can be naturally modeled as an online mechanism design problem where agents have multi-unit demand with non-increasing marginal valuations.
- We develop the first model-free online mechanism for perishable goods, where agents have multi-unit demand with decreasing marginal valuations. To ensure truthfulness, we show that this mechanism occasionally requires units to remain unallocated (we say that these units are 'burned'), even if there is demand for these units. This burning can be done in two ways: at the time of allocation, or on departure of the agent. The latter results in higher allocative efficiency and allocations are easier to compute, but occasionally requires the battery to be discharged which may not always be feasible in practice. Both variants are (weakly) *dominant-strategy incentive compatible* (DSIC), which means that no agent has an incentive to misreport their demand vector and the vehicle availability, regardless of the others' reports.
- We evaluate our mechanism through numerical simulation of electric vehicle charging using vehicle use data taken from a recent trial of EVs in the UK. In doing so, we show how the agent valuations can be derived from real monetary costs to the vehicle owners, by considering factors such as fuel prices, the distance that the owner expects to travel, and the energy efficiency of the vehicle. Experiments conducted in this realistic setting show that the mechanism with on-departure burning is highly scalable (it can handle hundreds of agents), and both variants outperform any fixed price mechanism for this problem in terms of allocative efficiency, while performing only slightly worse than a well known scheduling heuristic, which assumes non-strategic agents.

Throughout this paper, we focus on measuring allocative efficiency rather than seller profit, since our main design goal is to assure that the capacity of the distribution network is not exceeded, and that agents that need electricity most are allocated, rather than on maximizing profits.

2. RELATED WORK

Online mechanism design is an important topic in the multi-agent and economics literature and there are several lines of research in this field. One of these aims to develop online variants of Vickrey-Clarke-Groves (VCG) mechanisms [13, 7]. While these frameworks are quite general, their focus is on (a slight strengthening of) Bayesian-Nash incentive compatibility, whereas in this paper we focus on the stronger concept of DSIC. Moreover, these works rely on a model of future availability, as well as future supply (e.g., Parkes and Singh [13] use an MDP-type framework for predicting future arrivals), while the mechanism proposed here is model-free. Such models require fewer assumptions, and make computing allocations more tractable than VCG-like approaches.

Model-free settings are considered by both Hajiaghayi et al. [8] and Porter [14], who study the problem of online scheduling of a single, re-usable resource over a finite time period. They characterise truthful allocation rules for this setting and derive lower bound competitive ratios. A limitation of this work [12, 8, 14] is that they consider single-valued domains and, as we show, these existing approaches are no longer incentive compatible for our setting where agents' preferences are described by a vector of values.

Another related direction of work concerns designing truthful multi-unit demand mechanisms for static settings. A seminal result in this area is the sufficient characterisation of DSIC in terms of weak monotonicity (WMON) [1]. Although this work is relevant to our model (we briefly discuss the relationship between our mechanism and WMON in Section 4.3), it does not propose any specific mechanism, and, more importantly, existing results do not immediately apply to online domains where agents arrive over time and report their arrival and departure times, as well as their demand.

A different approach for dynamic problems is proposed by Juda and Parkes [9]. They consider a mechanism in which agents are allocated options (a right to buy) for the goods, instead of the goods themselves, and agents can choose whether or not to exercise the options when they exit the market. The concept of options would need to be modified to our setting with perishable goods, with power allocated and then burned so that the final allocation reflects only those options that would be allocated. It is not clear how our online burning mechanism maps to their method.

In addition to theoretical results, several applications have been suggested for online mechanisms, including: the allocation of Wi-Fi bandwidth at Starbucks [6], scheduling of jobs on a server [14] and the reservation of display space in online advertising [3]. However, this is the first work that proposes an online mechanism for electric vehicle charging, and we show how our theoretical framework naturally maps into this domain.

3. EV CHARGING MODEL

In this section we present a model for our problem, formally defining it as an online allocation problem.

(Supply) We consider a model with discrete and possibly infinite time steps (e.g., hourly slots) $t \in T$. At each time step, a number of units of electricity are available for vehicle charging as described by the *supply function* $S : T \rightarrow \mathbb{N}_0^+$, where $S(t)$ describes the number of units available at time t. Supply can vary over time due to changes in electricity demand for purposes other than vehicle charging, as well as changeable supply from renewable energy sources, such as wind and solar.

Importantly, we assume that all vehicle batteries are charged at the same rate.[4] Thus, a unit of electricity corresponds to the total energy consumed for charging a single vehicle in a single time step. Note that, while there are multiple units of supply at each time step (and agents have demand for multiple units), each agent can be allocated *at most* a single unit per time step. These units are allocated using a periodic multi-unit *auction*, one per time step. Units of electricity are *perishable*, meaning that any unallocated units at each time step will be lost.

(Agents and Preferences) Each vehicle owner is represented by an agent. Let $I = \{1, \ldots, n\}$ denote the set of all agents. An agent i's (true) availability for charging is given by its *arrival time* $a_i \in T$ (i.e., the earliest possible time the vehicle can be plugged in), and *departure time* $d_i \geq a_i, d_i \in T$ (i.e., after which the vehicle is needed by the owner). We will sometimes use $T_i = \{a_i, \ldots, d_i\}$ to indicate agent i's availability and we say that agent i is *active* in the market during this period. An agent has a positive value for units allocated when the agent is active, and has zero value for any units allocated outside of its active period. Furthermore, agents have preferences which determine their value or utility for a certain number of units of electricity. These preferences can change from one agent to another, and depend on factors such as the *efficiency* of the vehicle, travel distance, uncertainty in usage, battery capacity and local fuel prices. Formally, preferences are described by a valuation vector $\mathbf{v}_i = \langle v_{i,1}, v_{i,2}, \ldots, v_{i,m_i} \rangle$, where $v_{i,k}$ denotes the *marginal value* for the k^{th} unit and m_i is the maximum demand from agent i. That is, $v_{i,k} = 0$ for $k > m_i$. We will often use $v_{i,k+1}$, which describes the value for the next unit when an agent

[4]We believe that our approach can be extended to address settings with variable charge rates, but leave this for future work.

already has k units of electricity. Note that the agent is indifferent w.r.t. the precise allocation times, and merely cares about the total number of units received over the entire active period. These components together describe agent i's *type* $\theta_i = \langle a_i, d_i, \mathbf{v}_i \rangle$. We let $\theta = \{\theta_1, \ldots, \theta_n\}$, and θ_{-i} is the types of all agents except i. We will often use the notation $(\theta_i, \theta_{-i}) = \theta$.

We assume that agents have *non-increasing marginal valuations*, i.e., $v_{i,k} \geq 0$ and $v_{i,k+1} \leq v_{i,k}$. As we will show in Section 5, this assumption is realistic in a setting with plug-in hybrid and range-extended EVs, where the more a vehicle battery is charged, the less it needs to rely on the fuel-consuming internal combustion engine.

(Reports and Mechanism) Importantly, we allow agents the opportunity to misreport their types. Let $\hat{\theta}_i = \{\hat{a}_i, \hat{d}_i, \hat{\mathbf{v}}_i\}$ denote an agent's report.[5] Given this, a *mechanism* takes the agents' reported (or observed) types as input as they enter the system, and based on these reports determines the allocation of resources, as well as the payments to the agents. Our goal is then to design a mechanism which incentivises truthful reporting. The *decision policy* then specifies an allocation $\pi_i^{\langle t \rangle}(\hat{\theta}; k^{\langle t \rangle})$ at each time point $t \in T$ and for each agent $i \in I$, where $\mathbf{k}^{\langle t \rangle} = (k_1^{\langle t \rangle}, \ldots, k_n^{\langle t \rangle})$ denotes the total *endowments* of the agents at time t before the start of the auction at time t. That is:

$$k_i^{\langle t \rangle} = \sum_{t'=\hat{a}_i}^{t-1} \pi_i^{\langle t' \rangle}\Big(\hat{\theta}_i, \hat{\theta}_{-i} | \mathbf{k}^{\langle t' \rangle}\Big).$$

The policy π is subject to the constraint that units can only be allocated to agents within their reported activation period. In what follows, we will use the abbreviated notation $\pi_i^{\langle t \rangle}(\hat{\theta})$, leaving any dependence on the current endowments implicit. Furthermore, let $\pi_i(\hat{\theta}_i, \hat{\theta}_{-i}) = \sum_{t=\hat{a}_i}^{\hat{d}_i} \pi_i^{\langle t \rangle}(\hat{\theta}_i, \hat{\theta}_{-i})$ denote the total number of units allocated to agent i in its (reported) active time period. We will sometimes omit the arguments when this is clear from the context. Furthermore, the *payment policy* specifies a payment function $x_i(\hat{\theta}_i, \hat{\theta}_{-i} | \pi_i)$ for each agent i. Importantly, while allocations occur at each time point $t \in T$ (since units are perishable), payments are calculated at the reported departure time $\hat{d}_i$ (i.e., when the owner physically unplugs the vehicle).

(Limited Misreports) As in [12], we assume that the agents cannot report an *earlier arrival*, nor a *later departure*. Formally, $\hat{a}_i \geq a_i$ and $\hat{d}_i \leq d_i$, and we say such a pair $\langle \hat{a}_i, \hat{d}_i \rangle$ is *admissible*. This is a valid assumption in our domain because the agent's vehicle has to be physically plugged into the system, and this cannot be done if the vehicle is not available. However, it can still report an earlier departure since the vehicle can be unplugged before the vehicle is truly needed. Similarly, it can delay its effective arrival (i.e., after having arrived, the vehicle owner can delay actually plugging in the vehicle).

(Agent Utility) Given its preferences, an agent's utility by the departure time is given by the valuation for its obtained units of electricity, minus the payments to the mechanism. Formally:

$$u_i(\hat{\theta}_i; \theta_i) = \sum_{k=1}^{\pi_i(\hat{\theta}_i, \hat{\theta}_{-i})} v_{i,k} - x_i\Big(\hat{\theta}_i, \hat{\theta}_{-i} | \pi_i\left(\hat{\theta}_i, \hat{\theta}_{-i}\right)\Big) \quad (1)$$

[5]In practice, reported arrival and departure correspond to times when the vehicle is physically plugged into, and, respectively, unplugged from the network (which could differ from when the vehicle is truly available), which can typically be observed by the system. This is because we use a greedy-like scheduling approach (see Section 4) which does not require agents to report their types, nor have knowledge of their true types, in advance. Consequently, it is straightforward to apply our approach to settings where agents do not know their exact availability or this changes due to unexpected events.

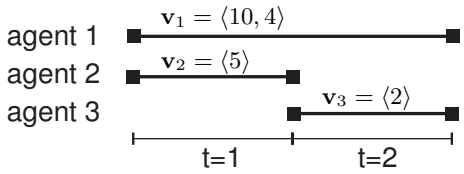


Figure 1: Example showing arrivals, departures, and valuation vectors of 3 agents.

4. THE ONLINE MECHANISM

In this section we consider the problem of designing a model-free mechanism for the above setting. Now, in the case of single-unit demand, a simple greedy mechanism with an appropriate payment policy is DSIC [12]. However, we will show through an example, that this is no longer the case in a multi-unit demand setting that we consider. A greedy allocation is formally defined as follows:

DEFINITION 1 (GREEDY ALLOCATION). *At each step t allocate the $S(t)$ units to the active agents with the highest marginal valuations, $v_{i,k_i^{\langle t \rangle}+1}$, where ties are broken randomly.*

Consider the example with 2 time steps and 3 agents in Figure 1, showing the agents' arrival, departure and valuations. Suppose furthermore that supply is $S(t) = 1$ at each time step. Greedy would then allocate both units to agent 1, because agent 1 has the highest marginal valuation in both auctions.

Now, consider the question of finding a payment scheme that makes greedy allocation truthful. How much should agent 1 pay? To answer this, note that the payment for the unit allocated at time $t = 1$ has to be at least 5. Otherwise, if agent 1 were present in the market only at time $t = 1$ and had a valuation $v_{1,1} \in (5 - \epsilon, 5)$, it would not be truthful, because it could report $\hat{v}_{1,1} > 5$ and still win. Similarly, the payment for the unit allocated at time $t = 2$ has to be at least 2. Thus, the *minimum* payment of agent 1 if allocated 2 units is $x_1(\hat{\theta} | \pi_1 = 2) = 7$.

On the other hand, how much should agent 1 pay if it were allocated only 1 unit instead? We argue no more than 2. If $x_1(\hat{\theta} | \pi_1 = 1) = 2 + \epsilon$ (where $\epsilon > 0$), then if the agent's first marginal value was instead $v_{1,1} \in (2, 2 + \epsilon)$, with remaining marginal values zero, then it would win in period 2, but it would pay $2 + \epsilon$ and hence have negative utility. However, if $x_1(\hat{\theta} | \pi_1 = 2) \geq 7$ and $x_1(\hat{\theta} | \pi_1 = 1) \leq 2$, then agent 1 wants only 1 unit, not 2, as allocated by the greedy mechanism (its utility for one unit is greater than for two, as $10 - 2 > 10 + 4 - 7$). Hence, online greedy allocation cannot be made truthful.[6]

In order to address this, in our mechanism we extend the Greedy decision policy by allowing the system to occasionally *"burn"* units of electricity when necessary, in order to maintain incentive compatibility. By burning we mean that this unit is not allocated to any agent, even when there is local demand. We consider two approaches: *immediate* burning, where the decisions to leave a unit unallocated is made at each time step before charging, and *on-departure* burning, where allocated units can be reclaimed by the system when the agent leaves the market (i.e., the corresponding amount of electricity is discharged from the battery on departure).

Each of these approaches has their own advantages and disadvantages. Burning on departure generally requires burning fewer units in some cases, and thus it leads to a higher efficiency. Moreover, the current method we use to determine payments for immediate burning can have a computational cost exponential in the

[6]Formally, this is because the decision policy violates a property called weak monotonicity [1]. In this paper, we omit a detailed discussion of this relationship, due to space restrictions.

number of the agents present, whereas for on-departure burning, the cost of determining payments is linear. However, in terms of the application domain, fast discharging of a vehicle's battery may not be practical.

Note that, for both approaches, the energy that is burnt is not necessarily wasted, but it is simply returned to the grid, to be used for other purposes than electric vehicle charging. For immediate burning, the unallocated electricity units are returned to the grid before it is actually charged by the agent. For the mechanism with on-departure burning, units may be charged first and then rapidly discharged when the agent leaves the market. While this may result in some loss, this is probably negligible w.r.t. the overall amount of electricity allocated.

4.1 The Mechanism

Before we introduce the decision policy, we show how we can compute a set of threshold values, which are used both to calculate the payments and to decide when to burn a unit of electricity. Let $k_{-i,j}^{\langle t \rangle} = \sum_{t'=a_j}^{t-1} \pi_j^{\langle t' \rangle}(\theta_{-i})$ denote the endowment of an active agent j at start time t, under the allocation we would have *in the absence of agent* i (note that calculating this value requires recomputing allocations without agent i in the market from a_i until the current time t). Then $v_{j,k_{-i,j}^{\langle t \rangle}+1}$ is the marginal valuation of agent j at time t in the absence of agent i. Given this, we define $v_{-i,t}^{(n)}$ to be the n^{th} highest of such valuations from all active agents $j \neq i$. Then $v_{-i,t}^{(S(t))}$, for supply $S(t)$, is the lowest value that is still allocated a unit at time t, if agent i were not present. Henceforth, we refer to $v_{-i,t}^{(S(t))}$ as the *marginal clearing value* for agent i in period t, and we will often use $v_{-i,t} = v_{-i,t}^{(S(t))}$ for brevity.

Now, let $\mathbf{p}_{-i}^{\langle t \rangle} = incr(v_{-i,a_i}, v_{-i,a_i+1}, \ldots v_{-i,t})$ denote agent i's price vector at time t, where a_i is the reported arrival time of agent i and $incr(.)$ is an operator which takes a vector of real values as input and returns it in increasing order. In addition, let $\mathbf{p}_{-i} = \mathbf{p}_{-i}^{\langle d_i \rangle}$ denote the value of this vector at time d_i when agent i leaves the market.

Intuitively, in any round t, the price $p_{-i,k}^t$ that agent i is charged for the k-th unit is the minimum valuation the agent could report for the k-th unit and win it by time t, given the greedy allocation policy with burning described below. Given this, the decision and payment policies of our mechanism are as follows.

- **Decision Policy** The decision consists of two stages.
  Stage 1 At each time point t, *pre-allocate* using Greedy (see Definition 1).
  Stage 2 We consider two variations in terms of when to decide to burn pre-allocated units:
  - **Immediate Burning.** Burn a unit whenever:
    $$v_{i,k_i^{\langle t \rangle}+1} < p^{\langle t \rangle}_{-i,k_i^{\langle t \rangle}+1}$$
  - **On-Departure Burning.** This type of burning occurs on reported departure. For each departing agent, burn any unit $k \leq \pi_i$ where $v_{i,k} < p_{-i,k}$.
- **Payment Policy** Payment occurs on reported departure. Given that π_i units are allocated to agent i at time $t = \hat{d}_i$, the payment collected from i is:
  $$x_i(\hat{\theta}_i, \hat{\theta}_{-i}|\pi_i) = \sum_{k=1}^{\pi_i} p_{-i,k} \quad (2)$$

Burning occurs whenever the marginal value for an additional unit is smaller than the marginal payment for that unit. Thus these values are effectively agent-specific threshold values, below which no unit is allocated to that agent. Moreover, it is important to note that the mechanism used for computing the prices mirrors the actual allocation mechanism. So, for example, if immediate burning is used in the decision policy, then for each agent i and for all times t, the values of the $\mathbf{p}_{-i}^{\langle t \rangle}$ vector are computed by re-running the market, in the absence of agent i using immediate burning, based on the reports of the other agents. Conversely, if on-departure burning is used for the decision policy, the same mechanism should be used in computing the $\mathbf{p}_{-i}$ prices.

	agent 1: $T_1 = \{1,2,3\}$ $\mathbf{v}_1 = \langle 10, 4 \rangle$	**agent 2:** $T_2 = \{1\}$ $\mathbf{v}_2 = \langle 5 \rangle$	**agent 3:** $T_3 = \{2,3\}$ $\mathbf{v}_3 = \langle 2 \rangle$
$t=1$	$k_1^{\langle 1 \rangle} = 0$ $v_{-1,1} = 5$ $\mathbf{p}_{-1}^{\langle 1 \rangle} = \langle 5 \rangle$ $\pi_1^{\langle 1 \rangle} = 1$	$k_2^{\langle 1 \rangle} = 0$ $v_{-2,1} = 10$ $\mathbf{p}_{-2}^{\langle 1 \rangle} = \langle 10 \rangle$ $\pi_2^{\langle 1 \rangle} = 0$	
$t=2$	$k_1^{\langle 2 \rangle} = 1$ $v_{-1,2} = 2$ $\mathbf{p}_{-1}^{\langle 2 \rangle} = \langle 2, 5 \rangle$ $\pi_1^{\langle 2 \rangle} = 0$ **(IM)** $\pi_1^{\langle 2 \rangle} = 1$ **(OD)**		$k_3^{\langle 2 \rangle} = 0$ $v_{-3,2} = 4$ $\mathbf{p}_{-3}^{\langle 2 \rangle} = \langle 4 \rangle$ $\pi_3^{\langle 2 \rangle} = 0$
$t=3$ IM	$k_1^{\langle 3 \rangle} = 1$ $v_{-1,3} = 0$ $\mathbf{p}_{-1}^{\langle 3 \rangle} = \langle 0, 2, 5 \rangle$ $\pi_1^{\langle 3 \rangle} = 1$		$k_3^{\langle 3 \rangle} = 0$ $v_{-3,3} = 4$ $\mathbf{p}_{-3}^{\langle 3 \rangle} = \langle 4, 4 \rangle$ $\pi_3^{\langle 3 \rangle} = 0$
$t=3$ OD	$k_1^{\langle 3 \rangle} = 2$ $v_{-1,3} = 0$ $\mathbf{p}_{-1}^{\langle 3 \rangle} = \langle 0, 2, 5 \rangle$ $\pi_1^{\langle 3 \rangle} = 0$		$k_3^{\langle 3 \rangle} = 0$ $v_{-3,3} = 0$ $\mathbf{p}_{-3}^{\langle 3 \rangle} = \langle 0, 4 \rangle$ $\pi_3^{\langle 3 \rangle} = 1$

Table 1: Example run of the mechanism with 3 agents and 3 time periods for immediate (IM) and on-departure (OD) burning. Grey cells indicate different values for IM and OD burning.

4.2 Example

To demonstrate how the mechanism works, we extend the previous example shown in Figure 1 to include a third time step, $t = 3$. Both agents 1 and 3 remain in the market at $t = 3$ (i.e., $d_1 = d_3 = 3$) and no new agents arrive. Furthermore, $S(t) = 1$ in $t \in \{1, 2, 3\}$, and so there are now 3 units to be allocated in total. Table 1 shows the endowments $k_i^{\langle t \rangle}$, the marginal clearing values $\mathbf{v}_{-i,t}$, the $\mathbf{p}_{-i}^{\langle t \rangle}$ vectors, and the allocation decisions $\pi_i^{\langle t \rangle}$ at different time periods.

We start by considering the allocations and payment using *immediate burning*. At time $t = 1$, Stage 1 of the mechanism allocates the unit to agent 1, and since $v_{1,1} = 10 \geq p_{-1,1}^{\langle 1 \rangle} = 5$, this unit is not burnt in the second stage. At time $t = 2$, the unit again gets pre-allocated to agent 1 since $v_{1,2} = 4 > v_{3,1} = 2$. However, the marginal clearing value $v_{-1,2}$ is inserted at the beginning of the $\mathbf{p}_{-1}^{\langle 2 \rangle}$ vector, and as a result $v_{1,2} = 4 < p_{-1,2}^{\langle 2 \rangle} = 5$. Consequently, this unit gets burnt and is allocated to neither of the agents. At time $t = 3$, therefore, the marginal value of agent 1 is still 4 (since its endowment is unchanged), and this value is added to agent 3's marginal clearing values. To calculate the marginal clearing value of agent 1, recall that the decision policy needs to be recomputed with agent 1 entirely removed from the market. In that case agent 3 would have been allocated a unit at time $t = 2$, and thus at time $t = 3$ the marginal value of this agent is 0. Thus, the value of 0 is inserted in the $\mathbf{p}_{-i}^{\langle 3 \rangle}$ vector. At $t = 3$, since agent 1 still has the highest marginal value, it is again pre-allocated the unit. However, now $v_{1,2} = 4 \geq \mathbf{p}_{-1,2}^{\langle 3 \rangle} = 2$, and therefore the unit is not burnt. So, in case of immediate burning, 2 out of 3 units are allocated to agent

1, and that agent pays $\mathbf{p}^{\langle 3\rangle}_{-1,1} + \mathbf{p}^{\langle 3\rangle}_{-1,2} = 2$.

Now consider the same setting but with *on-departure burning*. The first two time steps are as before, except that there is no burning at $t = 2$ (since this will be done on departure if needed). This changes the endowment state of agent 1 at $t = 3$, and therefore the marginal value of agent 1 at $t = 3$ is equal to $v_{1,3} = 0$. Therefore, the unit is allocated to agent 3, and the payment for this unit is $p_{-3,1} = 0$. The vector $\mathbf{p}^{\langle 3\rangle}_{-1}$ remains unchanged compared to the immediate burning case. At this point, there is no longer a need to burn one of the units of agent 1, since it has received $k = 2$ units, the same allocation as with immediate burning, and note that $v_{1,2} > p_{-1,2}$.

Still, it is possible to construct examples where, both with on-departure and immediate burning, half of the units need to be burnt. Furthermore, note that this unit cannot go to agent 3, because payment would have been $p^{\langle 3\rangle}_{-3,1} = 4$, which would result in a negative utility for agent 3.

4.3 Properties

In this section we prove that the above mechanism is DSIC. We will first establish DSIC with respect to valuations only, and prove truthful reporting of arrival and departure times separately. In more detail, we proceed in the following 3 stages: (i) We define the concept of a threshold policy, and show that, when coupled with an appropriate payment function, and given any admissible pair $\langle \hat{a}_i, \hat{d}_i\rangle$, if a decision policy is a threshold policy, then the mechanism is DSIC with respect to the valuations (Lemma 1). (ii) We show that our decision policy is a threshold policy (Lemma 2). (iii) Finally, we show that, if agents truthfully report their valuations, reporting $\hat{a}_i = a_i$, $\hat{d}_i = d_i$ is a weakly dominant strategy (Lemma 3). These results are combined in Theorem 1 to show that our policy is DSIC.

DEFINITION 2 (THRESHOLD POLICY). *A decision policy π is a threshold policy if, for a given agent i with fixed $\langle \hat{a}_i, \hat{d}_i\rangle$ and $\hat{\theta}_{-i}$, there exists a marginally non-decreasing threshold vector τ, independent from the report $\hat{\mathbf{v}}_i$ made by agent i, such that following holds: $\forall k, \hat{\mathbf{v}}_i$: $\pi_i(\hat{\theta}_i, \hat{\theta}_{-i}) \geq k$ if and only if $\hat{v}_{i,k} \geq \tau_k$.*

In other words, a threshold policy has a (potentially different) threshold τ_k for each k, such that agent i will receive at least k units if and only if its (reported) valuation for the k^{th} item is at least τ_k.[7]

Importantly, the vector τ has to be non-decreasing, i.e., $\tau_{k+1} \geq \tau_k$, and should be independent of the reported valuation vector $\hat{\mathbf{v}}_i$. Note that both of these properties are satisfied by the $\mathbf{p}_{-i}$ vector, and we will use this to show that our mechanism is a threshold policy. First, however, we show that a threshold policy with appropriate payments is DSIC with respect to the valuations.

LEMMA 1. *Fixing admissible $\langle \hat{a}, \hat{d}\rangle$ and $\hat{\theta}_{-i}$, if π is a threshold policy coupled with a payment policy:*

$$x_i(\hat{\theta}_i, \hat{\theta}_{-i}) = \textstyle\sum_{k=1}^{\pi_i(\hat{\theta}_i, \hat{\theta}_{-i})} \tau_k,$$

then if $\mathbf{v}_i$ is marginally non-increasing, reporting $\mathbf{v}_i$ truthfully is a weakly dominant strategy.

PROOF. Agent i's utility can be rewritten as:

$$u_i(\hat{\theta}_i; \theta_i) = \textstyle\sum_{k=1}^{\pi_i(\hat{\theta}_i, \hat{\theta}_{-i})} (v_{i,k} - \tau_k)$$

Since τ is independent of $\hat{\mathbf{v}}_i$, agent i can only potentially benefit by changing the allocation, $\pi_i(\hat{\theta}_i, \hat{\theta}_{-i})$. Since the values of $\tau_{k+1} \geq \tau_k$ (non-decreasing threshold vector) and $v_{i,k+1} \leq v_{i,k}$ (non-increasing marginal values), by definition 2 we have $v_{i,k} - \tau_k \geq 0$ for any $k \leq \pi_i(\theta_i)$ and $v_{i,k} - \tau_k \geq 0$ for any $k > \pi_i(\theta_i)$. Suppose that, by misreporting agent i is allocated $\pi_i(\hat{\theta}_i) > \pi_i(\theta_i)$, then $u_i(\hat{\theta}_i; \theta_i) < u_i(\theta_i; \theta_i)$ since:

$$\textstyle\sum_{k=\pi_i(\theta_i, \hat{\theta}_{-i})+1}^{\pi_i(\hat{\theta}_i, \hat{\theta}_{-i})} (v_{i,k} - \tau_k) < 0$$

Similarly, misreporting such that $\pi_i(\hat{\theta}_i, \hat{\theta}_{-i}) < \pi_i(\theta_i, \hat{\theta}_{-i})$ results in $u_i(\hat{\theta}_i; \theta_i) < u_i(\theta_i; \theta_i)$ since:

$$\textstyle\sum_{k=\pi_i(\hat{\theta}_i, \hat{\theta}_{-i})+1}^{\pi_i(\theta_i, \hat{\theta}_{-i})} (v_{i,k} - \tau_k) \geq 0$$

If misreporting has no effect on the allocation, the utility remains the same. Therefore, there is no incentive for agent i to misreport its valuations. □

Note that Greedy (as per Definition 1) is not a threshold policy. To see this, consider the example from Figure 1. As we saw earlier, Greedy allocates 2 units to agent 1, and the required threshold τ_2 for winning the second unit is 2 (below which Greedy would allocate 1 unit). However, if agent 1 had valuation $\mathbf{v}_1 = \langle 4, 4\rangle$, Greedy would allocate only 1 unit, even though $v_2 > \tau_2$, which conflicts with the requirement of a threshold policy.

The next lemma shows that the threshold condition holds if we include burning, and if we set the threshold values to $\tau_k = p_{-i,k}$.

LEMMA 2. *Given non-increasing marginal valuations, the decision policy π in Section 4.1 is (for either burning policy) a threshold policy where $\tau_k = p_{-i,k}$.*

PROOF. First, from the definition of vector $\mathbf{p}^{\langle t\rangle}_{-i}$ and $\mathbf{p}_{-i}$ from Section 4.1, the values of $\mathbf{p}^{\langle t\rangle}_{-i}$ are independent of the reports $\hat{\mathbf{v}}_i$ made by agent i. This is because each of its component values $v_{-i,a_i}, \ldots v_{-i,t}$ are computed based only on the reports of the other agents, by first removing agent i from the market.

Second, we need to show two inequalities, thus the proof is done in two parts. **Part 1:** Whenever $v_{i,k} \geq p_{-i,k}$, π_i allocates *at least* k units to agent i. **Part 2:** Whenever $v_{i,k} < p_{-i,k}$, π_i allocates *strictly less* than k units to agent i.

Part 1: Let $v_{i,k} \geq p_{-i,k}$. Suppose that agent i has the same marginal values, $v_{i,k}$, for the first k units (i.e., $v_{i,1} = v_{i,2} = \ldots = v_{i,k}$), then it will win exactly those auctions where $v_{i,k} \geq v_{-i,t}$, $t \in T_i$ in Stage 1 of the mechanism (ignoring tie breaking). Note that even when, by winning an auction, agent i displaces the losing marginal value to a future auction, since this value is less or equal to $v_{i,k}$, it will not affect the future auctions for agent i since it will still outbid that agent in the next auction. Now, because $p_{-i,j} \leq p_{-i,k}$ for $j \leq k$ (by definition), there must be at least k auctions where $p_{-i,k} \geq v_{-i,t}$ in the period $t \in T$, and since $v_{i,k} \geq p_{-i,k}$, agent i wins at least k auctions in Stage 1.

Furthermore, each time an auction is won, the clearing values appear as one of the j first elements of the $\mathbf{p}^t_{-i}$ vector, where j is the number of auctions so far (since these are the auctions with the lowest clearing values, and the clearing values are ordered ascendingly). Because agent i wins an auction in Stage 1 if and only if $v_{i,k} \geq v_{-i,t}$, it follows that $v_{i,k} = v_{i,j} \geq p_{-i,j}$ whenever it wins an auction in Stage 1. Therefore, there is no burning in Stage 2.

[7] A threshold policy satisfies weak-monotonicity (WMON) [1], and is therefore sufficient for truthfulness in this domain since we have bounded agent valuations and the domain is completely ordered, meaning that all payoff types agree on the same weak preference ordering on all allocations (i.e., more is always weakly better than less), and indifference to the way goods are allocated to other agents. We show that our decision policy has the threshold property, and thus the WMON, and that it also handles misreports of arrivals and departures.

The above holds if agent i has uniform marginal values of $v_{i,k}$ for the first k units. In fact, however, because of non-increasing valuations, we have $v_{i,j} \geq v_{i,k}$, for all $1 \leq j \leq k$, and thus the decision policy will allocate *at least* k units to agent i.

Part 2: Let $v_{i,k} < p_{-i,k}$. First consider the *on-departure burning* case. As per the definition of Stage 2 of the mechanism, unit k is burnt. However, we still need to show that any units $j > k$ are burnt as well. Since $p_{-i,j} \geq p_{-i,k}$ and $v_{-i,j} \leq v_{-i,k}$ for all $j > k$, it follows that $v_{i,j} < p_{-i,j}$ for all $j > k$. Therefore even if Stage 1 allocates k or more units, these will be burnt in Stage 2, and thus strictly less than k units remain.

Now consider the *immediate burning case*. Note that $p_{-i,k} \leq p_{-i,k}^{\langle t \rangle}$ for $(a_i + k - 1) \leq t \leq d_i$. That is, threshold values can only decrease over time. Thus it follows that $v_{-i,k} < p_{-i,k}^{\langle t \rangle}$ for any $(a_i + k - 1) \leq t \leq d_i$. Consider a case where, at time t_k, the k^{th} unit is allocated in Stage 1. Because $v_{-i,k} < p_{-i,k}^{\langle t_k \rangle}$, this unit will always be burnt in Stage 2 of the decision policy. Therefore, the final result is an allocation of strictly less than k units. □

By setting $\tau_k = p_{-i,k}$, the payment function in Equation 2 corresponds to the payment function in Lemma 1. Therefore the proposed mechanism is shown to be DSIC in valuations. We now complete the proof by showing that truthful reporting of the arrival and departure times are also DSIC (given limited misreports), given truthful reporting of $\mathbf{v}_i$.

LEMMA 3. *Given limited misreports, and assuming that truthfully reporting $\hat{\mathbf{v}}_i = \mathbf{v}_i$ is a dominant strategy for any given pair of arrival/departure reports $\langle \hat{a}_i, \hat{d}_i \rangle$, then it is a dominant strategy to report $\hat{a}_i = a_i$ and $\hat{d}_i = d_i$.*

PROOF. Let $\mathbf{p}_{-i}^{\langle \hat{a}_i, \hat{d}_i \rangle}$ denote the vector of increasingly ordered marginal clearing values (computed without i), given the agent reports $\hat{\theta}_i = \langle \hat{a}_i, \hat{d}_i, \mathbf{v}_i \rangle$. By reporting type $\hat{\theta}_i$, the agent is allocated $\pi_i(\hat{\theta}_i)$ items, and its total payment is: $\sum_{j=1}^{\pi_i(\hat{\theta}_i)} p_{-i,j}^{\langle \hat{a}_i, \hat{d}_i \rangle}$. For each agent i, misreporting from θ_i to $\hat{\theta}_i$ results in one of two cases:

$\boldsymbol{\pi_i(\hat{\theta}_i) = \pi_i(\theta_i)}$: Misreporting by agent i has no affect on the marginal clearing values $v_{-i,t}$, but can only decrease the size of the $\mathbf{p}_{-i}$ vector. In particular, due to limited misreports we have $\hat{a}_i \geq a_i$ and $\hat{d}_i \leq d_i$, and thus $\mathbf{p}_{-i}^{\langle \hat{a}_i, \hat{d}_i \rangle}$ contains a *subset* of the elements from $\mathbf{p}_{-i}^{\langle a_i, d_i \rangle}$. As these vectors are by definition increasingly ordered, it follows that $p_{-i,j}^{\langle \hat{a}_i, \hat{d}_i \rangle} \geq p_{-i,j}^{\langle a_i, d_i \rangle}, \forall j \leq (\hat{d}_i - \hat{a}_i + 1)$. Since the payment consists of the first $k_i = \hat{k}_i$ elements, this can only increase by misreporting.

$\boldsymbol{\pi_i(\hat{\theta}_i) \neq \pi_i(\theta_i)}$: First, we show that $\pi_i(\hat{\theta}_i) > \pi_i(\theta_i)$ could never occur. Since the marginal clearing values remain the same, but the number of auctions in which the agent participates decreases by misreporting, Stage 1 of the mechanism can only allocate fewer or equal items. Furthermore, since $p_{-i,j}^{\langle \hat{a}_i, \hat{d}_i \rangle} \geq p_{-i,j}^{\langle a_i, d_i \rangle}$, the possibility of burning can only increase in Stage 2. Thus, it always holds that $\pi_i(\hat{\theta}_i) \leq \pi_i(\theta_i)$.

Now, we consider the case $\pi_i(\hat{\theta}_i) < \pi_i(\theta_i)$. First, as shown for the case $\pi_i(\hat{\theta}_i) = \pi_i(\theta_i)$ above, we know that $\sum_{j=1}^{\pi_i(\hat{\theta}_i)} p_{-i,j}^{\langle a_i, d_i \rangle} \leq \sum_{j=1}^{\pi_i(\hat{\theta}_i)} p_{-i,k}^{\langle \hat{a}_i, \hat{d}_i \rangle}$ (i.e., the payment for those units won can only increase by misreporting arrival and/or departure). Furthermore, we know that the allocation $\pi_i(\theta_i)$ is preferable to any other allocation $\pi_i(\hat{\theta}_i) < \pi_i(\theta_i)$, otherwise reporting the true valuation vector $\mathbf{v}_i$ would not be a dominant strategy. Since the payment for these items is potentially even higher when misreporting, the agent cannot benefit by winning fewer items. □

We are now ready to present the main theoretical result:

THEOREM 1. *Given non-increasing marginal valuations and limited misreports, Greedy with on-departure and immediate burning and with payment function according to Equation 2 are DSIC.*

PROOF. The proof of this theorem follows directly from the above lemmas. Lemmas 1 and 2 show that, for any pair of arrival/departure (mis)-reports $\langle \hat{a}_i \hat{d}_i \rangle$ the decision policy is truthful in terms of the valuation vector $\mathbf{v}_i$, given an appropriate payment policy. Furthermore, the payments in Equation 2 correspond to those in Lemma 2, and therefore they truthfully implement the mechanism. Finally, Lemma 3 completes this reasoning, by showing that for a truthful report of valuation vector $\mathbf{v}_i$, agents cannot benefit by misreporting arrivals/departures. □

5. EXPERIMENTAL EVALUATION

In this section, we evaluate our proposed mechanism empirically. In doing so, we seek to answer a number of pertinent questions. First, since our greedy approach does not generally find the optimal allocation, we are interested in how close it comes to this in realistic settings. Second, we investigate the extent to which unit burning occurs in practice (i.e., how often units of electricity need to be burned by our decision policies, in order to ensure truthfulness). This is critical, as it may negatively affect efficiency. Finally, we compare our mechanism to a range of simpler truthful mechanisms that employ fixed pricing, as well as to a well-known online scheduling approach. These serve as benchmarks for our mechanism — fixed pricing is a common mechanism for selling goods in a wide range of settings, while the scheduling approach highlights what a non-truthful mechanism could achieve.

5.1 Experimental Setup

Our experimental setup is based on data collected during the first large-scale UK trial of EVs. In December 2009, 25 EVs were provided to members of the public as part of the CABLED (Coventry And Birmingham Low Emissions Demonstration) project.[8] The aim of this trial was to investigate real-world usage patterns of EVs. To this end, they were equipped with GPS and data loggers to record comprehensive usage information, such as trip durations and distances, home charging patterns and energy consumption.

We use the data published by this project for the first quarter of 2010 to realistically simulate typical behaviour patterns. More specifically, in each of our experiments, we simulate a single 24 hour day, where charging periods are divided into hourly time intervals. For the purpose of the experiments, a simulated day starts at 15:00, as vehicle owners begin to arrive back from work. To determine the arrival time of each agent, we randomly draw samples from the home charging start times reported by the project. These are highest after 18:00 and then quickly drop off during the night. Likewise, to simulate departures, we sample from data recording journey start times.

In order to simulate realistic marginal valuation vectors for the agents, we combine data from the project about journey distances with a principled approach for calculating the expected economic benefit of vehicle charging. In particular, we can calculate the expected utility of a given amount of charge (in kWh), c_e, given a price of fuel (in £/litre), p_p, an internal combustion engine efficiency (in miles/litre), e_p, an electric efficiency (in miles/kWh), e_e, and a probability density function, $p(m)$, that describes the distance to be driven the next day:

$$\mathbb{E}(u(c_e)) = \int_0^{\infty} \frac{p_p}{e_p} \cdot m \cdot p(m) \mathrm{d}m - \int_{c_e \cdot e_e}^{\infty} \frac{p_p}{e_p} \cdot m \cdot p(m) \mathrm{d}m, \quad (3)$$

[8]See `http://cabled.org.uk/`.

where the first term is the expected fuel cost without any charge, and the second term is the expected cost with a battery charge of c_e. Given this, and a charging rate (in kW), r_e, it is straight-forward to calculate the marginal valuation of the kth hour of charging time: $v_k = \mathbb{E}(u(k \cdot r_e)) - \mathbb{E}(u((k-1) \cdot r_e))$.

To generate a variety of marginal valuations, we note that e_e and e_p depend on the specific make and type of the EV and thus vary between households, while $p(m)$ depends on the driving behaviour of the car owner. We draw e_e uniformly at random from 2 – 4 miles/kWh and e_p is drawn from 9 – 18 miles/litre. Furthermore, we create $p(m)$ from daily driving distances presented in the CABLED report. These distances are typically short, with a daily mean of 23 miles, but the distribution has a long tail with a maximum of 101 miles. Next, we draw the capacity of a car battery from 15 – 25 kWh and set the charging rate to 3 kW. These and earlier specifications are all based on the Chevrolet Volt, the first mass-produced range-extended EV to be on the road in 2011. However, we include some variance to account for other vehicle types.

Finally, to derive the supply function S, we consider a realistic *neighbourhood-based* supply function using the average energy consumption of a UK household over time.[9] In this setting, the total energy available for charging depends on the number of households in the neighbourhood and the constraints of the local transformer. Hence, available supply during the night is significantly higher than during the day. Furthermore, we tested a range of other supply functions and valuation distributions, where we observed the same general trends as discussed in the remainder of this section. However, we omit the details here for brevity.

5.2 Benchmark Mechanisms

In addition to the two decision policies developed within this paper — Greedy with Immediate Burning *(Immediate)* and Greedy with On-Departure Burning *(On-Departure)* — we benchmark the following strategies that have been widely applied in similar settings:

Fixed Price allocates units to those agents that value them higher than a fixed price p. The price they pay for this unit is p. When demand is greater than supply, units are allocated randomly between all agents with a sufficiently high valuation. This mechanism is DSIC and so it constitutes a direct comparison to our mechanisms. However, to optimise the performance of the fixed price mechanism, p must be carefully chosen. Thus, we test all possible values (in steps of £0.01) and select the p that achieves the highest average efficiency (over 1000 trials) for a given setting. Thus, when showing the results of *Fixed Price*, this constitutes an upper bound of what could be achieved with this mechanism. We use the special case $p = 0$ as a baseline benchmark and denote this as ***Random***.

Heuristic allocates units such that a weighted combination of an agent's valuation and urgency (proximity to its departure time) is maximised. Here, an $\alpha \in [0, 1]$ parameter denotes the importance of the urgency, such that $\alpha = 1$ corresponds to the well-known earliest-deadline-first heuristic in scheduling, while $\alpha = 0$ indicates that units are always allocated to the agent with the highest valuation. This is not a truthful mechanism and we do not impose payments here, as its primary purpose is as a benchmark for our approach. Again, we always select the best α.

Optimal allocates units to agents to maximise the overall allocation efficiency, assuming complete knowledge of future arrivals and supply. Clearly, this mechanism is not practical and it is also not truthful (again we impose no payments), but it serves as an upper bound for the efficiency that could be achieved.

Having described the valuation calculation, the experimental setting, and the benchmarks, we now describe our results.

[9]We use the average evaluated during a work day in winter, available at `http://www.elexon.co.uk/`.

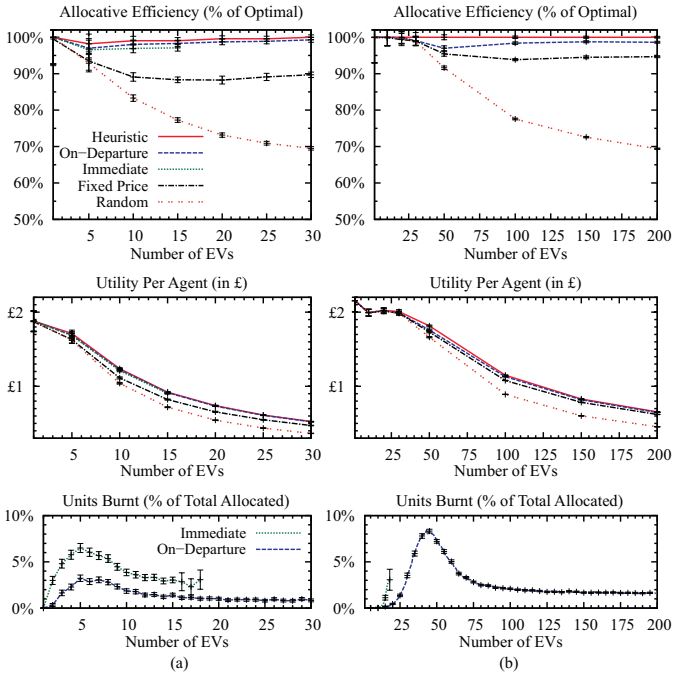


Figure 2: Results for a small neighbourhood with 30 houses (a) and a large one with 200 houses (b).

5.3 Results

For our experiments, we consider two possible neighbourhood sizes — one with 30 households and one with 200 households. In these settings, the capacity of the local transformer is constrained, so that only a couple of cars can charge at the same time in the 30 household case and up to 16 with 200 households. We choose such highly constrained settings here, because they are intrinsically more challenging and interesting than settings where all cars can be fully charged overnight. Across the experiments, we vary the number of these households that own an EV. Note here that we only show results for *Immediate* burning up to 15 agents, because our current implementation of this is computationally expensive. This is because the vector of marginal clearing values $\mathbf{p}_{-i}^{\langle t \rangle}$ at time t depends on which units are burned in i's absence (and as this vector is used to determine when burning takes place, it recursively depends on the corresponding vectors of all agents that are allocated in i's absence). Thus, we may potentially need to evaluate all subsets of agents, which grows exponentially with n. Although it may be possible to prune the search space efficiently in practice, we leave these computational aspects to future work. It is interesting that this does not apply to *On-Departure* burning, because here burning does not influence the agents' marginal clearing values.

The results for both settings are given in Figure 2. First, the top row shows the average[10] efficiency, normalised to the performance of *Optimal* (when there are more than 30 EV owners, *Optimal* becomes intractable and so we normalise results to the performance of *Heuristic* in those cases as a close approximation). Here, we note that our two burning policies consistently outperform (or match) all other truthful benchmarks. The improvement compared to *Random* is particularly pronounced, but our approach still achieves a significant improvement over the *Fixed Price* mechanism. For small neighbourhoods, this is almost 10%, while in larger neighbour-

[10]All results are averaged over 1000 trials. We plot 95% confidence intervals, and significant differences reported are at $t < 0.05$ level.

hoods, it is up to 5%. This is a promising result, because setting the optimal price for the fixed price strategy requires knowledge about the distributions of agents types, but our approach makes no such assumptions.

This improvement is due the ability of our mechanism to allocate the agents with the highest marginal valuations, while *Fixed Price* randomises over those that meet its price. Our approach is also responsive to changes in demand over time, consistently allocating units even when the highest valuations are low. In contrast, *Fixed Price* must be tuned to operate at any particular balance of supply and demand. Thus, it does not allocate when its price is unmet. It performs better in the larger setting because it is more likely that at least some of the agents meet the fixed price in this case.

Next, our mechanism also performs close to the *Optimal* and *Heuristic*, consistently achieving 95% or better, which indicates that our greedy approach performs well in realistic settings even without having access to complete information (such as departure times or even future arrivals). The lowest relative efficiency to the optimal is achieved when there are few EVs (about 20% of the neighbourhood). Here, scheduling constraints are most critical, as it may sometimes be optimal to prioritise an agent with lower valuations over one with higher valuations, but a longer deadline. This becomes less critical when there are more agents, as there are typically sufficiently many with high valuations. Finally, we see that *Immediate* burning achieves a slightly lower average efficiency than *On-Departure*. This is due to higher levels of burning, but the difference is small (and, in fact, not statistically significant).

In the second row of Figure 2, the average utility of each EV owner's allocation (not including the payments to the mechanism) is shown. This corresponds directly to the fuel costs that a single EV owner saves by using electricity instead of fuel. Initially, this is high (around £2), as there is little competition, but starts dropping as more EV owners compete for the same amount of electricity. Of key interest here is the horizontal separation between the different mechanisms. For a given fuel saving per agent, our mechanism can sustain a significantly larger number of agents than the other incentive-compatible mechanisms. For example, to save at least £1 per agent in the small neighbourhood, *Random* can support up to 10 EV owners, while *Immediate* and *On-Departure* achieve the same threshold for up to 14 EV owners (a 40% improvement). In the large neighbourhood, our mechanism can support around 60 additional vehicles in some cases (to achieve a £0.65 threshold).

Finally, the last row shows the average number of units that are burned by our two decision policies, as a percentage of the overall (tentatively) allocated units. Again, due to computational limitations, full results for the *Immediate* burning policy are only shown up to 15 agents. For up to 18 agents, results from only 100 trials are shown (resulting in larger confidence intervals). *On-Departure* burning clearly burns significantly fewer units than *Immediate*, as the latter sometimes unnecessarily burns units. There is also a clear maximum in the number of burned units when around 20% of households are EV owners. This is because there is a significant amount of competition, with many agents that have similar marginal valuations, and this induces burning. However, when the number of agents rises further, burning drops again. This is because agents are increasingly less likely to be allocated more than a single unit in these very competitive settings and so there is no need for burning. It should be noted that burning is generally low (for *On-Departure* burning), with typically only 1-2% of allocated units being burned (and always less than 10%).

6. CONCLUSIONS

This paper proposes a novel online allocation mechanism for a problem that is of great practical interest for the smart grid community, that of integrating EVs into the electricity grid. Our contribution to existing literature is two-fold. On the theoretical side, we extend model-free, online mechanism design with perishable goods to handle multi-unit demand with decreasing marginal valuations.

On the practical side, we empirically evaluate our mechanism in a real-world setting, and showed that the proposed mechanism is highly robust, and achieves better allocative efficiency than any fixed-price benchmark, while only being slightly suboptimal w.r.t. an established cooperative scheduling heuristic.

For future work we plan to look at several issues. First, in this paper we assumed all EVs have a uniform charging rate, but in the future we plan to extend the allocation model to deal with heterogeneous maximal charging rates (corresponding to different types of EVs). Second, it would be interesting to compare the performance of the model-free online mechanism proposed in this paper to a model-based approach, such as the one in [13]. Finally, this paper looked at performance in terms of a realistic application scenario, but we also plan to study the worst-case bounds on allocative efficiency and number of items our mechanism burns in future work.

7. REFERENCES

[1] S. Bikhchandani, S. Chatterji, R. Lavi, A. Mu'alem, N. Nisan, and A. Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.

[2] K. Clement-Nyns, E. Haesen, and J. Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on Power Systems*, 25(1):371–380, 2010.

[3] F. Constantin, J. Feldman, S. Muthukrishnan, and M. Pal. An online mechanism for ad slot reservations with cancellations. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA'09)*, pages 1265–1274, 2009.

[4] Department of Energy and Climate Change. *The UK low carbon transition plan: National strategy for climate and energy*. HM Government, 2009.

[5] P. Fairley. Speed bumps ahead for electric-vehicle charging. *IEEE Spectrum*, Jan. 2010. Avaliable at http://spectrum.ieee.org/green-tech/advanced-cars/speed-bumps-ahead-for-electricvehicle-charging/.

[6] E. Friedman and D.C. Parkes. Pricing WiFi at Starbucks– Issues in online mechanism design. In *Proc. of the 4th ACM Conf. on Electronic Commerce*, pages 240–241, 2003.

[7] A. Gershkov and B. Moldovanu. Efficient sequential assignment with incomplete information. *Games and Economic Behavior*, 68(1):144–154, 2010.

[8] M. Hajiaghayi, R. Kleinberg, M. Mahdian, and D.C. Parkes. Online auctions with re-usable goods. In *Proc. of the 6th ACM Conf. on Electronic Commerce (EC'05)*, pages 165–174, 2005.

[9] A.I. Juda and D.C. Parkes. An options-based solution to the sequential auction problem. *Artificial Intelligence*, 173:876–899, 2009.

[10] W.J. Mitchell, C.E. Borroni-Bird, and L.D. Burns. *Reinventing the automobile: Personal urban mobility for the 21st century*. MIT Press, 2010.

[11] Royal Academy of Engineering. *Electric Vehicles: Charged with potential*. Royal Academy of Engineering, 2010.

[12] D.C. Parkes. Online mechanisms. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, pages 411–439, 2007.

[13] D.C. Parkes and S. Singh. An MDP-Based approach to Online Mechanism Design. In *Proc. of NIPS'03*, 2003.

[14] R. Porter. Mechanism design for online real-time scheduling. In *Proc. the 5th ACM Conf. on Electronic Commerce (EC'04)*, pages 61–70, 2004.

[15] S. Vandael, N. Boucke, T. Holvoet, and G. Deconinck. Decentralized demand side management of plug-in hybrid vehicles in a smart grid. In *Proc. of 1st Int. Workshop on Agent Technologies for Energy Systems*, pages 67–74, 2010.

Voting Protocols

Homogeneity and Monotonicity of Distance-Rationalizable Voting Rules

Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
eelkind@ntu.edu.sg

Piotr Faliszewski
Department of Computer Science
AGH University of Science and Technology, Poland
faliszew@agh.edu.pl

Arkadii Slinko
Department of Mathematics
University of Auckland
New Zealand
slinko@math.auckland.ac.nz

ABSTRACT

Distance rationalizability is a framework for classifying voting rules by interpreting them in terms of distances and consensus classes. It can also be used to design new voting rules with desired properties. A particularly natural and versatile class of distances that can be used for this purpose is that of *votewise* distances [12], which "lift" distances over individual votes to distances over entire elections using a suitable norm. In this paper, we continue the investigation of the properties of votewise distance-rationalizable rules initiated in [12]. We describe a number of general conditions on distances and consensus classes that ensure that the resulting voting rule is homogeneous or monotone. This complements the results of [12], where the authors focus on anonymity, neutrality and consistency. We also introduce a new class of voting rules, that can be viewed as "majority variants" of classic scoring rules, and have a natural interpretation in the context of distance rationalizability.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

voting, distance rationalizability, monotonicity, homogeneity

1. INTRODUCTION

In collaborative environments, agents often need to make joint decisions based on their preferences over possible outcomes. Thus, social choice theory emerges as an important tool in the design and analysis of multiagent systems [13]. However, voting procedures that have been developed for human societies are not necessarily optimal for artificial agents and vice versa. For instance, there are voting rules that allow for polynomial-time winner determination (and thus are suitable for autonomous agents), yet have been deemed too complicated to be comprehended by an average voter in many countries; an example is provided by Single Transferable Vote. Further, unlike an electoral committee in a human society, the designer of a multi-agent voting system is usually unencumbered by legacy issues or the need to appeal to the general public, and can choose a voting rule that is most suitable for the application at hand, or, indeed, design a brand-new voting rule that satisfies the axioms that he deems important.

A recently proposed *distance rationalizability* framework [17, 10, 12, 11] is ideally suited for such settings. Under this framework, one can define a voting rule by a class of consensus elections and a distance over elections; the winners of an election are defined as the winners in the nearest consensus. In other words, for any election this rule seeks the most similar election with an obvious winner (where the similarity is measured by the given distance), and outputs its winner. Examples of natural consensus classes include *strong unanimity consensus*, where all voters agree on the ranking of all candidates, and *Condorcet consensus*, where there is a candidate that is preferred by a majority of voters to every other candidate. Combined with the *swap distance* (defined as the number of swaps of adjacent candidates that transforms one election into the other), these consensus classes produce, respectively, the Kemeny rule and the Dodgson rule.

The examples above illustrate that the distance rationalizability framework can be used to interpret (rationalize) existing voting rules in terms of a search for consensus (see [17] for a comprehensive list of results in this vein). It can also be applied to design new voting rules: for instance, in [10] the authors investigate the rule obtained by combining the Condorcet consensus with the Hamming distance. Further, by decomposing a voting rule into a consensus class and a distance we can hope to gain further insights into the structure of the rule. This decomposition is especially useful when the distance reflects changes in voters' opinions in a simple and transparent way. This is the case for the so-called *votewise* distances introduced in [12]. These are distances over elections that are obtained by aggregating distances between individual votes using a suitable norm, such as ℓ_1 or ℓ_∞. Indeed, paper [12] shows that one can derive conclusions about anonymity, neutrality and consistency of votewise rules (i.e., rules rationalized via votewise distances) from the basic properties of the underlying distances on votes, norms, and consensus classes.

In this paper we pick up this thread of research and study two important properties of voting rules not considered in [12], namely, monotonicity and homogeneity. Briefly put, monotonicity ensures that providing more support to a winning candidate cannot turn him into a loser, and homogeneity ensures that the result of an election depends on the proportions of particular votes and not on their absolute counts. Both properties are considered highly desirable for reasonable voting rules. We focus on the four standard consensus classes considered in the previous work (strong unanimity $\mathcal{S}$, una-

Cite as: Homogeneity and Monotonicity of Distance-Rationalizable Voting Rules, E. Elkind, P. Faliszewski, A. Slinko, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 821-828.

nimity $\mathcal{U}$, majority $\mathcal{M}$ and Condorcet $\mathcal{C}$) and ℓ_1- and ℓ_∞-norms. Our aim is to identify distances on votes that, combined with these norms and consensus classes, produce homogeneous and/or monotone rules.

Of the four consensus classes considered in this paper, the majority consensus $\mathcal{M}$ received relatively little attention in the existing literature. Thus, in order to study the homogeneity and monotonicity of the rules that are distance-rationalizable with respect to $\mathcal{M}$, we need to develop a better understanding of such rules. Our main result here is a characterization of all voting rules that are rationalizable with respect to $\mathcal{M}$ via a neutral distance on votes and the ℓ_1-norm. It turns out that such rules have a very natural interpretation: they are "majority variants" of classic scoring rules. This characterization enables us to analyze the homogeneity of the rules in this class, leading to a dichotomy result.

As argued above, a votewise distance-rationalizable rule can be characterized by three parameters: a distance on votes, a norm, and a consensus class. From this perspective, it is interesting to ask how much the voting rule changes if we vary one or two of these parameters. We provide two results that contribute to this agenda. First, we show that essentially any rule that is votewise-rationalizable with respect to $\mathcal{M}$ can also be rationalized with respect to $\mathcal{U}$, by modifying the norm accordingly. This enables us to answer a question left open in [11]. Second, we show that, for any consensus class and any distance on votes, replacing the ℓ_1-norm with the ℓ_∞-norm produces a voting rule that is an n-approximation of the original rule, where n is the number of voters. For the Dodgson rule, this transformation produces a rule that is polynomial-time computable and homogeneous. This line of work also emphasizes the constructive aspect of the distance rationalizability framework: we are able to derive new voting rules with attractive properties by combining a known consensus class with a known distance measure in a novel way.

Related work. The formal theory of distance rationalizability was put forward by Meskanen and Nurmi [17], though the idea, in one shape or another, appeared in earlier papers as well (see, e.g., [18, 2, 16, 15]). The goal of Meskanen and Nurmi was to seek best possible distance-rationalizations of classic voting rules. This research program was advanced by Elkind, Faliszewski, and Slinko [10, 12, 11], who, in addition to further classification work, also suggested studying general properties of distance-rationalizable voting rules. In particular, in [11] they identified an interesting and versatile class of distances—which they called votewise distances—that lead to rules whose properties can be meaningfully studied.

The study of distance rationalizability is naturally related to the study of another—much older—framework, which is based on interpreting voting rules as maximum likelihood estimators (the MLE framework). This framework could be dated back to Condorcet and has been pursued by Young [21], and, more recently, in [8], [7], and [19]. To date, most of the research on the MLE framework was concerned with determining which of the existing voting rules can be interpreted as maximum likelihood estimators; however, paper [19] also shows that the MLE approach can be used to deduce new useful voting rules.

This paper is loosely related to the work of Caragiannis et al. [6], where the authors give a monotone, homogeneous voting rule that calculates scores which approximate candidates' Dodgson scores up to an $O(m \log m)$ multiplicative factor, where m is the number of candidates. The relation to our work is twofold. First, we also focus on monotonicity and homogeneity, although our goal is to come up with a general method of constructing monotone and homogeneous rules and not to approximate particular rules. Second, in the course of our study we discover a homogeneous and polynomial-time computable voting rule that approximates the scores of candidates in Dodgson elections up to a multiplicative factor of n, where n is the number of voters. While the number of voters is usually much bigger than the number of candidates, and thus our algorithm is usually inferior to that of [6], it illustrates the power of the distance rationalizability framework.

The rest of the paper is organized as follows. Section 2 contains preliminary definitions regarding voting rules in general and the distance-rationalizability framework specifically. In Section 3 we provide a detailed study of rules that are votewise rationalizable with respect to the majority consensus. Sections 4 and 5 present our results on, respectively, homogeneity and monotonicity of votewise rules. We conclude in Section 6. We omit most proofs.

2. PRELIMINARIES

2.1. Basic notation. An *election* is a pair $E = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ is the set of *candidates* and $V = (v_1, \ldots, v_n)$ is the set of *voters*. Voter v_i is identified with a total order $\succ_i$ over C, which we will refer to as v_i's *preference order*, or *ranking*. We write $c_j \succ_i c_\ell$ to denote that voter v_i prefers c_j to c_ℓ. We denote by $\mathcal{P}(C)$ the set of all preference orders over C. For a voter v, we denote by $\mathrm{top}(v)$ the candidate ranked first by v, and set $\mathcal{P}(C, c) = \{v \in \mathcal{P}(C) \mid \mathrm{top}(v) = c\}$. For any voter $v_i \in V$ and a candidate $c \in C$, we denote by $\mathrm{rank}(v_i, c)$ the position of c in v_i's ranking. For example, if $\mathrm{top}(v_i) = c$ then $\mathrm{rank}(v_i, c) = 1$. A *voting rule* is a mapping $\mathcal{R}$ that for any election (C, V) outputs a non-empty subset of candidates $W \subseteq C$ called the *election winners*. Given an election $E = (C, V)$ and $s \in \mathbb{N}$, we denote by sE the election (C, sV), where sV is obtained by concatenating s copies of V.

Two important properties of voting rules that will be studied in this paper are homogeneity and monotonicity.

Homogeneity. A voting rule $\mathcal{R}$ is *homogeneous* if for each election $E = (C, V)$ and each positive $s \in \mathbb{N}$ we have $\mathcal{R}(E) = \mathcal{R}(sE)$.

Monotonicity. A voting rule $\mathcal{R}$ is *monotone* if for every election $E = (C, V)$, every $c \in \mathcal{R}(E)$ and every $E' = (C, V')$ obtained from E by moving c up in some voters' rankings (but not changing their rankings in any other way) we have $c \in \mathcal{R}(E')$.

2.2. Voting rules. We will now define the classic voting rules discussed in this paper, namely, scoring rules, (Simplified) Bucklin, and Dodgson.

Scoring rules In this paper, we will use a somewhat nonstandard definition of a scoring rule. Any vector $\alpha = (\alpha_1, \ldots, \alpha_m) \in (\mathbb{R}_+ \cup \{0\})^m$ defines a partial voting rule $\mathcal{R}_\alpha$ for elections with a fixed number m of candidates. Under this rule, for each preference order $u \in \mathcal{P}(C)$, $|C| = m$, a candidate $c \in C$ gets $\alpha_{\mathrm{rank}(u,c)}$ points (as is standard) and these values are summed up to obtain the score of c. However, we define the winners to be the candidates with the lowest score (rather than the highest, as is typical when discussing scoring rules). A sequence of scoring vectors $(\alpha^{(m)})_{m \in \mathbb{N}}$, where $\alpha^{(m)} \in (\mathbb{R}_+ \cup \{0\})^m$, defines a voting rule $\mathcal{R}_{(\alpha^{(m)})}$ which is applicable for any number of alternatives.

For example, in this notation the Borda rule is defined by a family of scoring vectors $\alpha^{(m)} = (0, 1, \ldots, m-1)$ and the k-approval is the family of scoring vectors given by $\alpha_i^{(m)} = 0$ for $i \leq k$, $\alpha_i^{(m)} = 1$ for $i > k$. The 1-approval rule is also known as Plurality. The traditional model, where the winners are the candidates with the highest score, can be converted to our notation by setting $\alpha'_i = \alpha_{\max} - \alpha_i$, where $\alpha_{\max} = \max_{i=1}^m \alpha_i$. The reason for this deviation is that in the context of this paper it will be much more convenient to speak of minimizing one's score. Note that, in gen-

eral, we do not require $\alpha_1 \leq \cdots \leq \alpha_m$, although this assumption is obviously required for monotonicity.

Note that vectors $(\alpha_1, \ldots, \alpha_m)$ and $(\beta\alpha_1, \ldots, \beta\alpha_m)$ define the same voting rule for any $\beta > 0$; the same is true for $(\alpha_1, \ldots, \alpha_m)$ and $(\alpha_1+\gamma, \ldots, \alpha_m+\gamma)$ for any $\gamma \geq 0$. Thus, in what follows, we normalize the scoring vectors by requiring their smallest coordinate to be 0, and the smallest non-zero coordinate to be 1.

Bucklin Under the *Bucklin rule*, we first determine the smallest value of k such that some candidate is ranked in top k positions by more than half of the voters. The winner(s) are the candidates that are ranked in the top k positions the maximum number of times. Under the *Simplified Bucklin rule* $\mathcal{R}_{sB}$, the winners are all candidates ranked in top k positions by a majority of voters.

Dodgson To define the Dodgson rule, we need to introduce the concept of a *Condorcet winner*. A Condorcet winner is a candidate that is preferred to any other candidate by a majority of voters. The *Dodgson score* of a candidate c is the smallest number of swaps of adjacent candidates that have to be performed on the votes to make c the Condorcet winner. The winner(s) under the Dodgson rule are the candidates with the lowest Dodgson score.

2.3. Norms and Metrics. A *norm* on $\mathbb{R}^n$ is a mapping $N : \mathbb{R}^n \to \mathbb{R}$ that has the following properties for all $x, y \in \mathbb{R}^n$: (1) $N(\alpha x) = |\alpha|N(x)$ for all $\alpha \in \mathbb{R}$; (2) $N(x) \geq 0$ and $N(x) = 0$ if and only if $x = (0, \ldots, 0)$; (3) $N(x + y) \leq N(x) + N(y)$.

Two important properties of norms that will be of interest to us are symmetry and monotonicity. We say that a norm N is *symmetric* if for each permutation $\sigma : [1, n] \to [1, n]$ it holds that $N(x_1, \ldots, x_n) = N(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$. For monotonicity, we make use of the definition proposed in [3]. Specifically, we say that a norm N is *monotone in the positive orthant*, or $\mathbb{R}^n_+$*-monotone*, if for any two vectors $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathbb{R}^n_+$ such that $x_i \leq y_i$ for all $i \leq n$ we have $N(x_1, \ldots, x_n) \leq N(y_1, \ldots, y_n)$.

A well-studied class of norms are the ℓ_p*-norms* given by

$$\ell_p(x_1, \ldots, x_n) = (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

for $p \in \mathbb{N}$. This definition can be extended to $p = +\infty$ by setting $\ell_\infty(x_1, \ldots, x_n) = \max\{x_1, \ldots, x_n\}$. Observe that for any $p \in \mathbb{N} \cup \{+\infty\}$ the ℓ_p norm is, in fact, a family of norms, i.e., it is well-defined on $\mathbb{R}^i$ for any $i \in \mathbb{N}$. Also, any such norm is clearly symmetric and monotone in the positive orthant.

A *metric*, or *distance*, on a set X is a mapping $d : X^2 \to \mathbb{R}$ that satisfies the following conditions for all $x, y, z \in X$: (1) $d(x, y) \geq 0$; (2) $d(x, y) = 0$ if and only if $x = y$; (3) $d(x, y) = d(y, x)$; (4) $d(x, z) \leq d(x, y) + d(y, z)$. A function that satisfies conditions (1), (3) and (4), but not (2), is called a *pseudodistance*.

Given a distance d on X and a norm N on $\mathbb{R}^n$, we can define a distance $N \circ d$ on X^n by setting

$$(N \circ d)(\mathbf{x}, \mathbf{y}) = N(d(x_1, y_1), \ldots, d(x_n, y_n))$$

for all vectors $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in X^n$. A distance defined in this manner is called a *product metric*.

In this paper, we will study distances over votes and their extensions to distances over elections via product metrics. Some examples of distances over votes are given by the *discrete distance* d_{discr}, the *swap distance* d_{swap}, and the *Sertel distance* d_{ser}, defined as follows. For any set of candidates C and any $u, v \in \mathcal{P}(C)$, we set $d_{\mathrm{discr}}(u, v) = 0$ if $u = v$ and $d_{\mathrm{discr}}(u, v) = 1$ otherwise. The swap distance d_{swap} is given by $d_{\mathrm{swap}}(u, v) = \frac{1}{2}|\{(c, c') \in C^2 \mid c \succ_u c', c' \succ_v c\}|$, where $\succ_u$ and $\succ_v$ are the preference orders associated with u and v, respectively. The Sertel distance between u and v is defined as the smallest value of i such that for all $j > i$ voters u and v rank the same candidate in position j.

A distance d on $\mathcal{P}(C)$ is called *neutral* if for any $u, v \in \mathcal{P}(C)$ and any permutation $\pi : C \to C$ we have $d(u, v) = d(\pi(u), \pi(v))$, where $\pi(x)$ denotes the vote obtained from x by moving candidate c_i into position $\mathrm{rank}(x, \pi(c_i))$, for $i = 1, \ldots, |C|$. Clearly, all distances listed above are neutral.

2.4. Distance Rationalizability. Intuitively, a consensus class is a collection of elections with an obvious winner. Formally, a *consensus class* is a pair $(\mathcal{E}, \mathcal{W})$ where $\mathcal{E}$ is a set of elections and $\mathcal{W}$: $\mathcal{E} \to C$ is a function that for each election $E \in \mathcal{E}$ outputs the alternative called the *consensus winner*. The following four consensus classes have been considered in the previous work on distance rationalizability:

Strong unanimity. Denoted $\mathcal{S}$, contains elections $E = (C, V)$ where all voters report the same preference order. The consensus winner is the candidate ranked first by all voters.

Unanimity. Denoted $\mathcal{U}$, contains all elections $E = (C, V)$ where all voters rank the same candidate first. The consensus winner is the candidate ranked first by all voters.

Majority. Denoted $\mathcal{M}$, contains all elections $E = (C, V)$ where more than half of the voters rank the same candidate first. The consensus winner is the candidate ranked first by the majority of voters.

Condorcet. Denoted $\mathcal{C}$, contains all elections $E = (C, V)$ with a Condorcet winner. The consensus winner is the Condorcet winner.

We say that a voting rule $\mathcal{R}$ is *compatible* with a consensus class $\mathcal{K}$ if for any consensus election $E \in \mathcal{K}$ it holds that $\mathcal{W}(E) = \mathcal{R}(E)$. Similarly, $\mathcal{R}$ is said to be *weakly compatible* with $\mathcal{K}$ if for any $E \in \mathcal{K}$ we have $\mathcal{W}(E) \in \mathcal{R}(E)$. Essentially all well-known voting rules are weakly compatible with $\mathcal{S}, \mathcal{U}$ and $\mathcal{M}$, but there are rules that are not compatible with any of these consensus classes (e.g., k-approval for $k > 1$). The rules that are compatible with $\mathcal{C}$ are also known as *Condorcet-consistent* rules; we use the term "compatibility" rather than "consistency" to avoid confusion with the consistency property of voting rules.

We are now ready to define the concept of distance rationalizability. Our definition below is taken from [12], which itself was inspired by [17, 10].

DEFINITION 2.1. *Let d be a distance over elections and let $\mathcal{K} = (\mathcal{E}, \mathcal{W})$ be a consensus class. The* $(\mathcal{K}, d)$-score *of a candidate c in an election E is the distance (according to d) between E and a closest election $E' \in \mathcal{E}$ such that $c \in \mathcal{W}(E')$. A voting rule $\mathcal{R}$ is* distance-rationalizable *via a consensus class $\mathcal{K}$ and a distance d over elections (is* $(\mathcal{K}, d)$-rationalizable*) if for each election E the set $\mathcal{R}(E)$ consists of all candidates with the smallest $(\mathcal{K}, d)$-score.*

A particularly useful class of distances to be used in distance rationalizability constructions is that of *votewise* distances, which are obtained by combining a distance over votes with a suitable norm. Formally, given a set of candidates C, consider a distance d over $\mathcal{P}(C)$ and a family of norms $\mathcal{N} = (N_i)_{i=1}^{\infty}$, where N_i is a norm over $\mathbb{R}^i$. We define a distance $\widehat{d^{\mathcal{N}}}$ over elections with the set of candidates C as follows: for any $E = (C, V)$, $E' = (C, V')$, we set $\widehat{d^{\mathcal{N}}}(E, E') = (N_i \circ d)(V, V')$ if $|V| = |V'| = i$, and $\widehat{d^{\mathcal{N}}}(E, E') = +\infty$ if $|V| \neq |V'|$. A voting rule $\mathcal{R}$ is said to be $\mathcal{N}$*-votewise distance-rationalizable* (or simply $\mathcal{N}$*-votewise*) with respect to a consensus class $\mathcal{K}$ if there exists a distance d over votes such that $\mathcal{R}$ is $(\mathcal{K}, \widehat{d^{\mathcal{N}}})$-rationalizable. When $\mathcal{N}$ is the ℓ_p-norm for some $p \in \mathbb{N} \cup \{+\infty\}$, we write $\widehat{d^p}$ instead of $\widehat{d^{\ell_p}}$, and when $\mathcal{N} = \ell_1$, we omit the index altogether and write $\widehat{d}$. It is known

that any voting rule is distance-rationalizable with respect to any consensus class that it is compatible with [12]. However, some voting rules are not $\mathcal{N}$-votewise distance-rationalizable with respect to standard consensus classes for any reasonable norm $\mathcal{N}$ [11].

Let us now consider some examples of distance-rationalizations of voting rules. Nitzan [18] was the first to show that Plurality is $(\mathcal{U}, \widehat{d}_{\text{discr}})$-rationalizable and Borda is $(\mathcal{U}, \widehat{d}_{\text{swap}})$-rationalizable. It is easy to see that Dodgson is $(\mathcal{C}, \widehat{d}_{\text{swap}})$-rationalizable and Kemeny is $(\mathcal{S}, \widehat{d}_{\text{swap}})$-rationalizable. The distance $\widehat{d^{\infty}_{\text{ser}}}$, combined with the majority consensus, yields the Simplified Bucklin rule [12].

For any set of candidates C with $|C| = m$ and a scoring vector $\alpha = (\alpha_1, \ldots, \alpha_m)$, paper [12] defines a (pseudo)distance $d_\alpha(u, v)$ on $\mathcal{P}(C)$ as as $d_\alpha(u, v) = \sum_{j=1}^{m} |\alpha_{\text{rank}(u,c_j)} - \alpha_{\text{rank}(v,c_j)}|$, and shows that if—in our notation—$\alpha_1 \leq \alpha_k$ for all $k > 1$ then $\mathcal{R}_\alpha$ is $(\mathcal{U}, \widehat{d}_\alpha)$-(pseudo)distance-rationalizable.

3. $\mathcal{M}$-SCORING RULES

The majority consensus is a very natural notion of agreement in the society. However, it has received little attention in the literature so far. Here we will show that it leads to a series of interesting rules with nice properties.

DEFINITION 3.1. *For any scoring vector $\alpha = (\alpha_1, \ldots, \alpha_m)$, let $\mathcal{M}$-$\mathcal{R}_\alpha$ be a partial voting rule defined on the profiles with m alternatives as follows. Given an election $E = (C, V)$ with $|C| = m$ and $V = (v_1, \ldots, v_n)$, for each candidate $c \in C$, we define the $\mathcal{M}$-score of c as the sum of $\lfloor \frac{n}{2} \rfloor + 1$ lowest values among $\alpha_{\text{rank}(v_1,c)}, \ldots, \alpha_{\text{rank}(v_n,c)}$. The winners are the candidates with the lowest $\mathcal{M}$-$\mathcal{R}_\alpha$ scores. As in the classic case, a family of scoring vectors $(\alpha^{(i)})_{i \in \mathbb{N}}$ defines an $\mathcal{M}$-scoring rule $\mathcal{M}$-$\mathcal{R}_{(\alpha^{(i)})}$.*

We will refer to voting rules from Definition 3.1 as *$\mathcal{M}$-scoring rules*. Such rules (or their slight modifications) are often used for score aggregation in real-life settings; for example, it is not unusual for a professor to grade the students on the basis of their five best assignments out of six or in some sport competitions to select winners on the basis of one or more of their best attempts.

It is not hard to see that $\mathcal{M}$-Plurality is equivalent to Plurality: under both rules, the winners are the candidates with the maximum number of first-place votes. However, essentially all other scoring rules differ from their $\mathcal{M}$-counterparts.

PROPOSITION 3.2. *Consider a normalized scoring vector $\alpha = (\alpha_1, \ldots, \alpha_m)$. The rule $\mathcal{M}$-$\mathcal{R}_\alpha$ coincides with $\mathcal{R}_\alpha$ if and only if (i) $\alpha_1 = \ldots = \alpha_m$ or (ii) $\alpha_i = 0$ for some $i \in \{1, \ldots, m\}$ and $\alpha_j = 1$ for all $j \neq i$.*

The $\mathcal{M}$-scoring rules tend to ignore extremely negative opinions. Therefore, intuitively, they are less susceptible to manipulation: if a voter v ranks a candidate c lower than the majority of other voters, v cannot manipulate against c by moving her to the bottom of their ranking. In this section we will show that these rules are also very interesting from the distance rationalizability point of view: it turns out that they essentially coincide with the class of rules that are ℓ_1-votewise rationalizable with respect to $\mathcal{M}$.

We will first need to generalize a result from [12] to pseudodistances and weak compatibility.

PROPOSITION 3.3. *Any voting rule that is pseudodistance-rationalizable with respect to a consensus class $\mathcal{K}$ is weakly compatible with $\mathcal{K}$.*

Now, we can characterize $\mathcal{M}$-scoring rules that are (pseudo)distance-rationalizable with respect to $\mathcal{M}$.

PROPOSITION 3.4. *Let $\alpha = (\alpha_1, \ldots, \alpha_m)$ be a normalized scoring vector. The rule $\mathcal{M}$-$\mathcal{R}_\alpha$ is ℓ_1-votewise distance-rationalizable with respect to $\mathcal{M}$ if and only if $\alpha_1 = 0$, $\alpha_j > 0$ for all $j \neq 1$. Further, $\mathcal{M}$-$\mathcal{R}_\alpha$ is ℓ_1-votewise pseudodistance-rationalizable with respect to $\mathcal{M}$ if and only if $\alpha_1 = 0$.*

We remark that our proof generalizes to scoring rules and $\mathcal{U}$, thus answering a question left open in [10], namely, whether scoring rules with $\alpha_i = \alpha_j$ for $i, j > 1$ can be distance-rationalized (rather than pseudodistance-rationalized). Further, in [10] the authors consider only monotone scoring rules, i.e., rules that satisfy—in our notation—$\alpha_1 \leq \cdots \leq \alpha_m$, while our result holds for all scoring vectors.

The following lemma explains how to find an $\mathcal{M}$-consensus that is nearest to a given election with respect to a given ℓ_1-votewise distance.

LEMMA 3.5. *Let $\mathcal{R}$ be a voting rule that is $(\mathcal{M}, \widehat{d})$-rationalized. Let $E = (C, V)$ be an arbitrary election where $V = (v_1, \ldots, v_n)$ and let $E' = (C, U)$ be an $\mathcal{M}$-consensus such that $\widehat{d}(E, E')$ is minimal among all n-voter $\mathcal{M}$-consensuses over C. Let $c \in C$ be the consensus winner of (C, U). Then, for each $i = 1, \ldots, n$, either $u_i \in \arg\min_{x \in \mathcal{P}(C,c)} d(x, v_i)$ or $u_i = v_i$.*

Combining Lemma 3.5 with the argument in the proof of Theorem 4.9 in [12], we can show that the converse of Proposition 3.4 is also true: any voting rule that can be pseudodistance-rationalized via $\mathcal{M}$ and a neutral ℓ_1-votewise pseudodistance is, in fact, an $\mathcal{M}$-scoring rule. Also, any $\mathcal{M}$-scoring rule is obviously neutral. We can summarize these observations in the following theorem.

THEOREM 3.6. *Let $\mathcal{R}$ be a voting rule. There exists a neutral ℓ_1-votewise pseudodistance $\widehat{d}$ such that $\mathcal{R}$ is $(\mathcal{M}, \widehat{d})$-rationalizable if and only if $\mathcal{R}$ can be defined as an $\mathcal{M}$-scoring rule $\mathcal{M}$-$\mathcal{R}_{(\alpha^{(i)})}$ such that $\alpha^{(i)}_1 \leq \alpha^{(i)}_j$ for all $j > 1$ and all $i \in \mathbb{N}$.*

The discussion above suggests that using the majority consensus to rationalize a voting rule is similar to using the unanimity consensus, except that we only take into account the best "half-plus-one" votes. In fact, it turns out that under very weak assumptions we can translate a votewise rationalization of a rule with respect to $\mathcal{M}$ to a votewise rationalization of that rule with respect to $\mathcal{U}$.

DEFINITION 3.7. *Let $\mathcal{N} = (N_i)_{i=1}^{\infty}$ be a family of functions where for each i, $i \geq 1$, N_i is a mapping from $\mathbb{R}^i$ to $\mathbb{R}$. We define a family $\mathcal{N}^{\mathcal{M}} = (N_i^{\mathcal{M}})_{i=1}^{\infty}$ as follows. For each $i \geq 1$, $N_i^{\mathcal{M}}$ is a mapping from $\mathbb{R}^i$ to $\mathbb{R}$ given by*

$$N_i^{\mathcal{M}}(x_1, \ldots, x_i) = N_{\lfloor \frac{i}{2} \rfloor + 1}(|x_{\pi(1)}|, \ldots, |x_{\pi(\lfloor \frac{i}{2} \rfloor + 1)}|),$$

where π is a permutation of $[1, i]$ such that $|x_{\pi(1)}| \geq |x_{\pi(2)}| \geq \cdots \geq |x_{\pi(i)}|$.

For a family of symmetric norms $\mathcal{N} = (N_i)_{i=1}^{\infty}$ that are monotone in the positive orthant, the family $\mathcal{N}^{\mathcal{M}}$ is also a family of norms, which we will call the *majority variant* of $\mathcal{N}$.

PROPOSITION 3.8. *Let $\mathcal{N} = (N_i)_{i=1}^{\infty}$ be a family of norms, where each N_i is a symmetric norm on $\mathbb{R}^i$ that is monotone in the positive orthant. Then the family $\mathcal{N}^{\mathcal{M}} = (N^{\mathcal{M}})_{i=1}^{\infty}$ is also a family of symmetric norms that are monotone in the positive orthant.*

As an immediate corollary we get the following result.

COROLLARY 3.9. *Let $\mathcal{N}$ be a family of symmetric norms that are monotone in the positive orthant and let d be a distance over votes. Let $\mathcal{R}$ be a voting rule that is $(\mathcal{M}, \widehat{d^{\mathcal{N}}})$-rationalizable. Then $\mathcal{R}$ is $(\mathcal{U}, \widehat{d^{\mathcal{N}\mathcal{M}}})$-rationalizable.*

This discussion illustrates that when a rule can be rationalized in several different ways, the right choice of a consensus class plays an important role, as it may greatly simplify the underlying norm and hence the distance. This is why it pays to keep a variety of consensus classes available and search for the best distance rationalizations possible. Corollary 3.9 also has a useful application: Paper [11] shows that STV[1] cannot be rationalized with respect to $\mathcal{S}, \mathcal{C}$ or $\mathcal{U}$ by any neutral $\mathcal{N}$-votewise distance, where $\mathcal{N}$ is a family of symmetric norms monotone in the positive orthant. Corollary 3.9 allows us to extend this result to $\mathcal{M}$, thus showing that STV cannot be rationalized by a "reasonable" votewise distance with respect to any of the standard consensus classes.

4. HOMOGENEITY

Homogeneity is a very natural property of voting rules. It can be interpreted as a weaker form of another appealing property, namely, consistency. Recall that a voting rule $\mathcal{R}$ is said to be *consistent* if for any two elections $E_1 = (C, V_1)$ and $E_2 = (C, V_2)$ with $\mathcal{R}(E_1) \cap \mathcal{R}(E_2) \neq \emptyset$ it holds that $\mathcal{R}(C, V_1 + V_2) = \mathcal{R}(E_1) \cap \mathcal{R}(E_2)$, where $V_1 + V_2$ denotes the concatenation of V_1 and V_2. Thus, loosely speaking, homogeneity imposes the same requirement as consistency, but only for the restricted case $V_1 = V_2$. Now, consistency is known to be hard to achieve: by Young's theorem [20], the only voting rules that are simultaneously anonymous, neutral and consistent are the scoring rules (or their compositions). In contrast, we will now argue that for many consensus classes and many values of $p \in \mathbb{N} \cup \{+\infty\}$, the rules that are ℓ_p-votewise rationalizable with respect to these classes are homogeneous. We start by showing that this is the case for ℓ_p, $p \in \mathbb{N}$, and consensus classes $\mathcal{S}$ and $\mathcal{U}$.

THEOREM 4.1. *For any distance d on votes, the voting rule $\mathcal{R}$ that is $(\mathcal{K}, \widehat{d^p})$-rationalizable for $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}\}$ and $p \in \mathbb{N}$ is homogeneous.*

For $\mathcal{M}$, the conclusion of Theorem 4.1 is no longer true. However, we can fully characterize homogeneous rules that can be rationalized via $\mathcal{M}$ and a neutral ℓ_1-votewise pseudodistance (recall that by Theorem 3.6 all such rules are necessarily $\mathcal{M}$-scoring rules). For convenience, we state the following theorem for scoring vectors that satisfy $\alpha_1 \leq \cdots \leq \alpha_m$; it is not hard to show that this can be done without loss of generality.

THEOREM 4.2. *A voting rule $\mathcal{M}$-$\mathcal{R}_\alpha$ with a normalized scoring vector $\alpha = (\alpha_1, \ldots, \alpha_m)$ that satisfies $\alpha_1 \leq \cdots \leq \alpha_m$ is homogeneous if and only if $\alpha_m = 1$ or $\alpha_{\lceil \frac{m}{2} \rceil} = 0$.*

PROOF SKETCH. Set $h = \lceil \frac{m}{2} \rceil$. We skip the easy proof of the case when $\alpha_m = 1$ (remember that the smallest non-zero coordinate is also 1). When $\alpha_h = 0$, then by the pigeonhole principle either there exists a candidate that is ranked in top h positions by a majority of voters (and its score is 0), or each candidate is ranked in top h positions by exactly half of the voters. In both cases, it is easy to show that the rule is homogeneous; we omit the details.

We will now show that if $\alpha_m > 1$ and $\alpha_h > 0$, the rule $\mathcal{M}$-$\mathcal{R}_\alpha$ is not homogeneous. We will only consider the case $\alpha_3 > 1$ (note that this implies $\alpha_2 = 1$); by careful padding, the construction in this proof can be modified to work for the general case.

Set $\alpha = \alpha_3$; we have $\alpha_1 = 0$, $\alpha_2 = 1$. We start by considering the case $m = 3$; later, we will generalize our construction to $m > 3$. Suppose first that $\alpha = \frac{p}{q}$ is a rational number written in its lowest terms. We construct an election $E = (C, V)$, where $C = \{a, b, c\}$ and V consists of the following votes:

1. $2p + q + 1$ votes $a \succ b \succ c$,
2. $2q + p + 1$ votes $b \succ c \succ a$, and
3. $p + q - 2$ votes $c \succ b \succ a$.

We observe that $|V| = 4(p + q)$, and the $\mathcal{M}$-scores of a and b are equal to p, and the $\mathcal{M}$-score of c is at least $p + q + 3$. Hence, both a and b are winners of E. On the other hand, in the election $2E = (C, 2V)$, the $\mathcal{M}$-scores of candidates a and b are, respectively, $(2q - 1)\alpha = 2p - \alpha$ and $2p - 1$. Since $\alpha > 1$, it cannot be the case that both a and b are winners of $2E$. Thus, in this case $\mathcal{M}$-$\mathcal{R}_\alpha$ is not homogeneous.

Now, if α is irrational, consider its continued fraction expansion $\alpha = (a_0, a_1, \ldots)$, and the successive convergents $\frac{h_i}{k_i}$, $i = 0, 1, \ldots$, where $h_0 = a_0$, $k_0 = 1$, $h_1 = a_1 h_0 + 1$, $k_1 = a_1$, and $h_i = a_i h_{i-1} + h_{i-2}$, $k_i = a_i k_{i-1} + k_{i-2}$ for $i \geq 2$. We know that for even values of i we have $\frac{h_i}{k_i} < \alpha$ and $|\alpha - \frac{h_i}{k_i}| < \frac{1}{k_i k_{i+1}}$. Also, it is not hard to show that for any $N > 0$ there exists an even value of i such that $k_{i+1} > N$. Thus, we pick an even i such that $k_{i+1} > \frac{2}{\alpha - 1}$ (recall that $\alpha > 1$). We obtain

$$0 < \alpha - \frac{h_i}{k_i} < \frac{1}{k_i k_{i+1}} < \frac{\alpha - 1}{2k_i}.$$

Now, set $p = h_i$, $q = k_i$, let $\varepsilon = \alpha - \frac{p}{q}$, and use the same construction as above. In E, the $\mathcal{M}$-score of a is $q\alpha$, the $\mathcal{M}$-score of b is $p < q\alpha$, and the $\mathcal{M}$-score of c exceeds that of a and b, so b is the unique winner. On the other hand, in $2E$ the $\mathcal{M}$-score of a is $(2q - 1)\alpha = 2p + 2q\varepsilon - \alpha$, while the $\mathcal{M}$-score of b is $2p - 1$. We have $\varepsilon < \frac{\alpha - 1}{2q}$, so a has a lower $\mathcal{M}$-score than b, and hence b cannot be the winner of $2E$. Thus, in this case, too, our rule is not homogeneous.

For $m > 3$, we modify this construction by adding $m - 3$ dummy candidates that each voter ranks last (in some arbitrary order). □

We have seen that many voting rules that are ℓ_1-votewise distance-rationalizable with respect to $\mathcal{M}$ are not homogeneous. However, homogeneity appears to be easier to achieve if we use the ℓ_∞-norm instead of ℓ_1. For example, Simplified Bucklin has been shown to be $(\mathcal{M}, \widehat{d_{\text{ser}}^{\infty}})$-rationalizable [12] and it can be shown to be homogeneous. Indeed, this follows from a more general result stating that ℓ_∞-votewise rules are homogeneous as long as they are rationalized via a consensus class that satisfies a fairly weak requirement.

DEFINITION 4.3. *A consensus class $\mathcal{K}$ is* split-homogeneous *if the following two conditions hold:*

(a) If U is a $\mathcal{K}$-consensus then for every positive integer s it holds that sU is a $\mathcal{K}$-consensus with the same winner;

(b) If U and W are two profiles, with n votes each, such that $U + W$ is a $\mathcal{K}$-consensus, then at least one of U and W is a $\mathcal{K}$-consensus with the same winner as $U + W$.

It turns out that combining a split-homogeneous consensus class with an ℓ_∞-votewise distance produces a homogeneous rule.

[1]We skip the description of STV due to space,but we mention that STV is one of the very few nontrivial voting rules used in real-life political systems.

THEOREM 4.4. *For any split-homogeneous consensus class $\mathcal{K}$ and any pseudodistance d on votes, the voting rule that is rationalized via $\mathcal{K}$ and $\widehat{d^\infty}$ is homogeneous.*

It is not hard to see that the consensus classes $\mathcal{S}, \mathcal{U}$ and $\mathcal{M}$ are split-homogeneous. Thus, we obtain the following corollary.

COROLLARY 4.5. *For any $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}\}$ and any pseudodistance d on votes, the voting rule that is rationalized via $\mathcal{K}$ and $\widehat{d^\infty}$ is homogeneous.*

In contrast, the Condorcet consensus is not split-homogeneous.

EXAMPLE 4.6. Consider the following election $E = (C, V)$ with $C = \{a, b, c, d, e\}$ and $V = (v_1, \ldots, v_{12})$:

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
a	b	c	d	e	c	e	a	b	c	d	c
b	c	d	e	a	a	d	e	a	b	c	a
c	d	e	a	b	b	c	d	e	a	b	b
d	e	a	b	c	d	b	c	d	e	a	d
e	a	b	c	d	e	a	b	c	d	e	e

Voters $v_1, \ldots, v_5$ form a Condorcet cycle, and voters $v_7, \ldots, v_{11}$ are obtained from voters $v_1, \ldots, v_5$ by reversing their preferences. Voters v_6 and v_{12} are identical and rank c first. It is not hard to verify that c is the Condorcet winner in E. On the other hand, in elections $E_1 = (C, V_1)$ and $E_2 = (C, V_2)$, where $V_1 = (v_1, \ldots, v_6)$ and $V_2 = (v_7, \ldots, v_{12})$, c is not a Condorcet winner both in E_1 and in E_2.

Indeed, we can construct an ℓ_∞-votewise distance that combined with $\mathcal{C}$ yields a nonhomogeneous rule.

PROPOSITION 4.7. *There exists a distance d on votes such that that rule rationalized by $\mathcal{C}$ and $\widehat{d^\infty}$ is not homogeneous.*

The combination of $\mathcal{C}$ and an ℓ_1-votewise distance does not necessarily lead to a homogeneous rule either: it is well known that the Dodgson rule is not homogeneous (see, e.g., [4] for a recent survey of Dodgson rule deficiencies), yet it is $(\mathcal{C}, \widehat{d}_{\text{swap}})$-rationalizable. In fact, we are not aware of any homogeneous voting rule that is ℓ_1-votewise distance-rationalizable with respect to $\mathcal{C}$. In contrast, we can construct a homogeneous rule that is ℓ_∞-votewise distance-rationalizable with respect to $\mathcal{C}$ by replacing ℓ_1 with ℓ_∞ in the rationalization of the Dodgson rule. We will call the resulting rule Dodgson$^\infty$; the next section will explain the name of the rule. To prove that Dodgson$^\infty$ is homogeneous, we will first explain how to determine the winners under this rule. It turns out that, in contrast to the Dodgson rule itself, Dodgson$^\infty$ admits a polynomial-time winner determination algorithm.

PROPOSITION 4.8. *The problem of computing the $(\mathcal{C}, \widehat{d^\infty}_{\text{swap}})$-score of a given candidate c in an election $E = (C, V)$ is in* P.

PROOF. It can be verified that the following algorithm runs in polynomial time and computes the $(\mathcal{C}, \widehat{d^\infty}_{\text{swap}})$-score of c.

1. Set $k = 0$.
2. If c is a Condorcet winner of E then return k.
3. For each vote where c is not ranked first, swap c and its predecessor.
4. Increase k by 1 and go to Step 2.

□

Using the algorithm given in the proof of Proposition 4.8, it is not hard to show that Dodgson$^\infty$ is homogeneous.

PROPOSITION 4.9. *Dodgson$^\infty$ is homogeneous.*

The Dodgson$^\infty$ rule has some desirable properties that the Dodgson rule itself is lacking. Thus, it is interesting to ask if the fomer can be used to approximate the latter, in the sense of Caragiannis et al. [5, 6]. It turns out that the answer is "yes": each ℓ_∞-votewise rule approximates the corresponding ℓ_1-votewise rule. However, the approximation ratio is often quite large.

THEOREM 4.10. *For any consensus class $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$ and any distance d on votes, let $\mathcal{R}$ and $\mathcal{R}^\infty$ be the voting rules rationalized via $\mathcal{K}$ and $\widehat{d}$ and $\widehat{d^\infty}$, respectively. Let $\text{score}_E^{\mathcal{R}}(c)$ (respectively, $\text{score}_E^{\mathcal{R}^\infty}(c)$) denote the $(\mathcal{K}, \widehat{d})$-score (respectively, $(\mathcal{K}, \widehat{d^\infty})$-score) of a candidate c in an election $E = (C, V)$. Then for each election $E = (C, V)$ and each candidate $c \in C$ we have*

$$\text{score}_E^{\mathcal{R}^\infty}(c) \leq \text{score}_E^{\mathcal{R}}(c) \leq |V| \cdot \text{score}_E^{\mathcal{R}^\infty}(c).$$

For the majority consensus we can strengthen the approximation guarantee from $|V|$ to $\lceil \frac{|V|}{2} + 1 \rceil$ using the fact that we only need the majority of the voters to rank a candidate first for him to be the $\mathcal{M}$-winner.

Of course, these approximations are very weak as they depend linearly on the number of voters; their appeal is in their generality. Further, since for the Dodgson rule its ℓ_∞-variant is homogeneous and polynomial-time computable, an appealing conjecture is that replacing ℓ_1 with ℓ_∞ in the rationalization of a voting rule is a general recipe for designing voting rules that are homogeneous and admit an efficient winner determination algorithm. It is unlikely that this conjecture holds unconditionally, but it would be very interesting to identify sufficient conditions for it to hold.

5. MONOTONICITY

Monotonicity is a very desirable property of voting rules: it stipulates that campaigning in favor of a candidate should not hurt him. While homogeneity seems to be essentially a function of the norm and the consensus class (as illustrated by Theorem 4.1 and Theorem 4.4, which hold for any distance d on votes), monotonicity seems to be most closely related to the properties of the distance on votes. Therefore, in this section we propose several notions of monotonicity for distances on votes that, combined with appropriate norms and consensus classes, produce a monotone rule. We do not consider the Condorcet consensus in this section: even a very well-behaved distance such as $\widehat{d}_{\text{swap}}$ may produce a non-monotone rule when combined with $\mathcal{C}$ (recall that the resulting rule is Dodgson, which is known to be non-monotone (see, e.g., [4]). Also, for simplicity, we focus on ℓ_1-votewise rules and ℓ_∞-votewise rules.

Let C be a set of candidates and let d be a distance on votes. How can we specify a condition on d so that voting rules rationalized using this distance are monotone? Consider an election with a winner c, a vote y, a vote $x \in \mathcal{P}(C, c)$ and a vote $z \in \mathcal{P}(C, a)$ for some $a \neq c$. It is tempting to require that for any vote y' obtained from y by pushing c forward it holds that $d(y', x) \leq d(y, x)$ and $d(y', z) \geq d(y, z)$. However, this condition turns out to be so strong that no reasonable distance can satisfy it. Indeed, suppose that y ranks c in position three or lower, and y' is obtained from y by shifting c by one position. Then y does not rank c in the first position, and our condition should hold for $z = y'$, implying $d(y, y') \leq 0$, which is clearly impossible.

Thus, we need to relax the condition above. There are two ways of doing so. First, we can require that when we move c forward in the vote, the distance to x declines faster than the distance to z. Alternatively, instead of imposing this condition for all $x \in \mathcal{P}(C, c)$

and $z \in \mathcal{P}(C, a)$, we can require that it holds for the closest vote that ranks c first, and the closest vote that ranks a first, respectively. We will now show that both relaxations, which we call, respectively, *relative monotonicity* and *min-monotonicity*, lead to meaningful conditions that are satisfied by some natural distances, and, combined with appropriate consensus classes, result in monotone voting rules. We consider relative monotonicity first.

DEFINITION 5.1. *Let C be a set of candidates. We say that a distance d on $\mathcal{P}(C)$ is* relatively monotone *if for each $c \in C$, every two preference orders y and y' such that y' is identical to y except that y' ranks c higher than y, and every two preference orders x and z such that x ranks c first and z does not, it holds that*

$$d(x, y) - d(x, y') \geq d(z, y) - d(z, y').$$

As a quick sanity check, we note that the swap distance, d_{swap}, satisfies the relative monotonicity condition. Indeed, let $d = d_{\text{swap}}$ and let C be a set of candidates, c be a candidate in C, and let y, y', x, and z be as in the definition of relative monotonicity. In addition, let k be a positive integer such that y' is identical to y except in y' candidate c is ranked k positions higher. We need k swaps to transform y into y' so $d(y, y') = k$. We first note that $d(x, y) - d(x, y') = k$. This is so because the swap distance measures the number of inverses between two preference orders. As x ranks c on top and y' ranks it k positions higher than y does (without any other changes), the number of inverses between x and y' is the same as that between x and y less k. By the triangle inequality $d(z, y) \leq d(z, y') + d(y', y) = d(z, y') + k$, hence $d(z, y) - d(z, y') \leq k$ and this completes the proof.

Relative monotonicity of a distance on votes naturally translates to the monotonicity of the resulting voting rule, provided we use ℓ_1 as a norm and either $\mathcal{S}$ or $\mathcal{U}$ as a consensus.

THEOREM 5.2. *Let $\mathcal{R}$ be a voting rule rationalized by $(\mathcal{K}, \widehat{d})$, where $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}\}$ and d is a relatively monotone distance on votes. Then $\mathcal{R}$ is monotone.*

However, relative monotonicity is a remarkably strong condition, not satisfied even by very natural distances that are, intuitively, monotone.

EXAMPLE 5.3. Consider a scoring vector $\alpha = (0, 1, 2, 3, 4, 5)$ that corresponds to the 6-candidate Borda rule and a candidate set $C = \{c, d, x_1, x_2, x_3, x_4\}$. Consider the following four votes:

$$\begin{aligned} x &: c > d > x_1 > x_2 > x_3 > x_4, \\ z &: x_1 > c > x_2 > x_3 > x_4 > d, \\ y &: x_1 > x_2 > d > c > x_3 > x_4, \\ y' &: x_1 > x_2 > c > d > x_3 > x_4. \end{aligned}$$

Note that y and y' are identical except that in y' candidate c is ranked one position higher, and that c is ranked on top in x and is not ranked on top in z. We verify that $d_\alpha(x, y) - d_\alpha(x, y') = 0$ but $d_\alpha(z, y) - d_\alpha(z, y') = 2$. Thus, d_α is not relatively monotone.

Our second approach to monotone distances, i.e., min-monotonicity, captures the intuition that d_α in the example above should be classified as monotone. We first define min-monotonicity formally.

DEFINITION 5.4. *Let C be a set of candidates. We say that a distance d on $\mathcal{P}(C)$ is* min-monotone *if for every candidate $c \in C$ and every two preference orders y and y' such that y' is the same as y except that it ranks c higher, for each $a \in C \setminus \{c\}$ we have:*

$$\begin{aligned} \min_{x \in \mathcal{P}(C,c)} d(x, y) &\geq \min_{x' \in \mathcal{P}(C,c)} d(x', y'), \\ \min_{z \in \mathcal{P}(C,a)} d(z, y) &\leq \min_{z' \in \mathcal{P}(C,a)} d(z', y'). \end{aligned}$$

We will now argue that for any non-decreasing scoring vector α the distance d_α is min-monotone.

PROPOSITION 5.5. *Let $\alpha = (\alpha_1, \ldots, \alpha_m)$ be a normalized scoring vector. (Pseudo)distance d_α is min-monotone if and only if α is nondecreasing.*

Proposition 5.5, combined with the proof of Theorem 4.9 of [12] gives the next corollary.

COROLLARY 5.6. *A voting rule $\mathcal{R}$ is $(U, \widehat{d})$-rationalizable for some min-monotone neutral pseudodistance d on votes if and only if $\mathcal{R}$ can be defined via a family of nondecreasing scoring vectors (one for each number of candidates).*

In essence, Proposition 5.5 ensures that for every nondecreasing scoring vector α, $\mathcal{R}_\alpha$ is ℓ_1-votewise rationalizable with respect to $\mathcal{U}$ via a min-monotone distance over votes, and the definition of min-monotonicity ensures that the scoring vector derived in the proof of Theorem 4.9 of [12] is nondecreasing.

Min-monotonicity is also useful in the context of the majority consensus: for $\mathcal{M}$, we can show an analogue of Theorem 5.2 both for ℓ_1-votewise rules and for ℓ_∞-votewise rules.

THEOREM 5.7. *Let d be a min-monotone distance on votes, and let $\mathcal{R}$ be the voting rule rationalized by $(\mathcal{M}, \widehat{d^{\mathcal{N}}})$, where $\mathcal{N} \in \{\ell_1, \ell_\infty\}$. Then $\mathcal{R}$ is monotone.*

However, it is not clear how to apply the notion of min-monotonicity in the context of the strong unanimity consensus. The reason is that given a profile V of voters over some candidate set C, finding an $\mathcal{S}$-consensus closest to V requires finding a single preference order u that minimizes the aggregated distance from V to this order. However, it need not be the case that u is a preference order that minimizes the distance from some vote $v \in V$ to a preference order that ranks $\text{top}(u)$ first.

Finally, we remark that we can combine both relaxations considered in this section, obtaining a class of distances that includes both relatively monotone distances and min-monotone distances.

DEFINITION 5.8. *Let C be a set of candidates. We say that a distance d on $\mathcal{P}(C)$ is* relatively min-monotone *if for each candidate $c \in C$ and each two preference orders y and y' such that y' is identical to y except that y' ranks c higher than y, for each candidate $a \in C \setminus \{c\}$ it holds that*

$$\begin{aligned} &\min_{x \in \mathcal{P}(C,c)} d(x, y) - \min_{x' \in \mathcal{P}(C,c)} d(x', y') \geq \\ &\quad \min_{z \in \mathcal{P}(C,a)} d(z, y) - \min_{z' \in \mathcal{P}(C,a)} d(z', y'). \end{aligned}$$

PROPOSITION 5.9. *Each distance on votes that is relatively monotone or min-monotone is relatively min-monotone.*

PROOF. Due to lack of space, we only give the proof for relatively monotone distances. Let C be a set of candidates, $c, a \in C$, and let $y, y' \in \mathcal{P}(C)$ be identical, except y' ranks c higher than y. Pick $\hat{x} \in \arg\min_{x' \in \mathcal{P}(C,c)} d(x', y)$, $\hat{z} \in \arg\min_{z' \in \mathcal{P}(C,a)} d(z', y')$. Then

$$\begin{aligned} &\min_{x \in \mathcal{P}(C,c)} d(x, y) - \min_{x' \in \mathcal{P}(C,c)} d(x', y') \geq d(\hat{x}, y) - d(\hat{x}, y') \geq \\ &\quad d(\hat{z}, y) - d(\hat{z}, y') \geq \min_{z \in \mathcal{P}(C,a)} d(z, y) - \min_{z' \in \mathcal{P}(C,a)} d(z', y'). \end{aligned}$$

Thus, d is relatively min-monotone. □

For $\mathcal{U}$ the proof of Theorem 5.2 extends to relatively min-monotone distances (and hence to min-monotone distances).

COROLLARY 5.10. *Any voting rule rationalized by $\mathcal{U}$ and $\widehat{d}$, where d is relatively min-monotone distance on votes, is monotone.*

6. CONCLUSIONS

We have discussed homogeneity and monotonicity of voting rules that are distance-rationalizable via votewise distances, focusing on ℓ_p-votewise rules, $p \in \mathbb{N} \cup \{+\infty\}$. A quick summary of our results is given in Tables 1 and 2.

	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{M}$	$\mathcal{C}$
ℓ_1	Y (Th. 4.1)	Y (Th. 4.1)	Y/N (Th. 4.2)	n (Dodgson)
ℓ_∞	Y (Th. 4.4)	Y (Th. 4.4)	Y (Th. 4.4)	y (Prop. 4.9)/ n (Prop. 4.7)

Table 1: (Homogeneity) Y at the intersection of column $\mathcal{K}$ and row $\mathcal{N}$ indicates that for any distance d on votes the $(\mathcal{K}, \widehat{d^{\mathcal{N}}})$-rationalizable rule is homogeneous. Y/N refers to a dichotomy result, and y/n refer to examples of homogeneous/non-homogeneous rules.

	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{M}$
ℓ_1	rel-mon (Th. 5.2)	rel-min-mon (Cor. 5.10)	min-mon (Th. 5.7)
ℓ_∞	?	?	min-mon (Th. 5.7)

Table 2: (Monotonicity) At the intersection of column $\mathcal{K}$ and row $\mathcal{N}$, we indicate a sufficient condition on d (relative monotonicity, min-monotonicity, relative min-monotonicity) for the $(\mathcal{K}, \widehat{d^{\mathcal{N}}})$-rationalizable rule to be monotone.

Motivated by our goal, we obtained a number of results, that, while not directly related to the primary topic of our study, contribute to the general understanding of votewise rationalizable rules. In particular, we identified a natural family of voting rules, which we called $\mathcal{M}$-scoring rules. These rules constitute a (provably distinct) variant of scoring rules that, when counting points for a given candidate, ignore the less favorable half of the votes. We have shown that $\mathcal{M}$-scoring rules have a natural interpretation in the context of distance rationalizability. By establishing a relationship between rules that are rationalizable with respect to $\mathcal{U}$ and $\mathcal{M}$, we resolved (in the negative) an open question about votewise rationalizability of STV posed in [11]. Also, our study of monotonicity allowed us to refine a result of [12] characterizing the class of scoring rules in terms of distance-rationalizability (our Corollary 5.6).

Our work leads to several open problems. First, we are far from having a complete understanding of homogeneity of the rules that are votewise distance-rationalizable with respect to the Condorcet consensus; even less is known about the monotonicity of such rules. Also, it would be interesting to know whether there are distances $d \neq d_{\text{swap}}$ for which the winner determination for the $(\mathcal{C}, \widehat{d^{\infty}})$-rationalizable rule is easier than for the $(\mathcal{C}, \widehat{d})$-rationalizable rule; the same question can be asked for the consensus class $\mathcal{S}$. We are also very much interested in finding less demanding, yet practically useful, conditions on distances that lead to monotone rules.

Acknowledgments This research was supported by AGH University of Science and Technology Grant no. 11.11.120.865, Foundation for Polish Science's program Homing/Powroty, Polish Ministry of Science and Higher Education grant N-N206-378637, NRF Research Fellowship (NRF-RF2009-08), NTU Start-Up Grant, and the Science Faculty of the University of Auckland FRDF grant 3624495/9844. The authors are grateful to the anonymous AAMAS rveiewers for their constructive feedback.

7. REFERENCES

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008.

[2] N. Baigent. Metric rationalisation of social choice functions according to principles of social choice. *Mathematical Social Sciences*, 13(1):59–65, 1987.

[3] F. Bauer, J. Stoer, and C. Witzgall. Absolute and monotonic norms. *Numerische Matematic*, 3:257–264, 1961.

[4] F. Brandt. Some remarks on Dodgson's voting rule. *Mathematical Logic Quarterly*, 55(4):460–463, 2009.

[5] I. Caragiannis, J. Covey, M. Feldman, C. Homan, C. Kaklamanis, N. Karanikolas, A. Procaccia, and J. Rosenschein. On the approximability of Dodgson and Young elections. In *SODA'09*, pp.1058–1067. 2009.

[6] I. Caragiannis, C. Kaklamanis, N. Karanikolas, and A. Procaccia. Socially desirable approximations for Dodgson's voting rule. In *ACM EC'10*, pp. 253–262, 2010.

[7] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *IJCAI'09*, pp. 109–115, 2009.

[8] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *UAI'05*, pp. 145–152, 2005.

[9] D. Coppersmith, L. Fleisher, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Transactions on Algorithms*, 6(3):Article 55, 2010.

[10] E. Elkind, P. Faliszewski, and A. Slinko. On distance rationalizability of some voting rules. In *TARK'09*, pp. 108–117, 2009.

[11] E. Elkind, P. Faliszewski, and A. Slinko. Good rationalizations of voting rules. In *AAAI'10*, pp. 774–779, 2010.

[12] E. Elkind, P. Faliszewski, and A. Slinko. On the role of distances in defining voting rules. In *AAMAS'10*, pp. 375–382, 2010.

[13] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[14] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *STOC'07*, pp. 95–103, 2007.

[15] C. Klamler. Borda and Condorcet: Some distance results. *Theory and Decision*, 59(2):97–109, 2005.

[16] C. Klamler. The Copeland rule and Condorcet's principle. *Economic Theory*, 25(3):745–749, 2005.

[17] T. Meskanen and H. Nurmi. Closeness counts in social choice. In M. Braham and F. Steffen, editors, *Power, Freedom, and Voting*. Springer-Verlag, 2008.

[18] S. Nitzan. Some measures of closeness to unanimity and their implications. *Theory and Decision*, 13(2):129–138, 1981.

[19] L. Xia, V. Conitzer, and J. Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *AAMAS'10*, pp. 399–406, 2010.

[20] H. Young. Social choice scoring functions. *SIAM Journal on Applied Mathematics*, 28(4):824–838, 1975.

[21] H. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16(2):335–353, 1977.

Possible Winners When New Alternatives Join: New Results Coming Up!

Lirong Xia
Dept. of Computer Science
Duke University
Durham, NC 27708, USA
lxia@cs.duke.edu

Jérôme Lang
LAMSADE
Université Paris-Dauphine
75775 Paris Cedex, France
{lang,jerome.monnot}@lamsade.dauphine.fr

Jérôme Monnot

ABSTRACT

In a voting system, sometimes multiple new alternatives will join the election after the voters' preferences over the initial alternatives have been revealed. Computing whether a given alternative can be a co-winner when multiple new alternatives join the election is called the *possible co-winner with new alternatives (PcWNA)* problem and was introduced by Chevaleyre et al. [6]. In this paper, we show that the PcWNA problems are NP-complete for the Bucklin, Copeland$_0$, and maximin (a.k.a. Simpson) rule, even when the number of new alternatives is no more than a constant. We also show that the PcWNA problem can be solved in polynomial time for plurality with runoff. For the approval rule, we examine three different ways to extend a linear order with new alternatives, and characterize the computational complexity of the PcWNA problem for each of them.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences–Economics; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Economics, Theory

Keywords

Computational social choice, possible co-winner with new alternatives

1. INTRODUCTION

In many real-life situations, multiple voters have to choose a common alternative out of a set that can grow during the process. For instance, when a committee wants to decide which proposal should be approved, some applications might arrive late (due to unexpected delay in the mailing system, etc). Suppose that we have already elicited the preference of the voters (members of the committee) on the initial proposals. It is important for the applicants to know whether they are already out (so that they can submit the same proposal to other founding sources right away without waiting for the committee members to make the final decision). A recent paper by Chevaleyre et al. [6] considers the following problem: *suppose that the voters' preferences about a set of initial alternatives have already been elicited, and we know that a given number k of new alternatives will join the election; we ask who among the initial alternatives can possibly win the election in the end*. This problem is a special case of the *possible winner problem* [18, 21, 20, 3, 4, 2], restricted to the case where the incomplete profile consists of a collection of full rankings over the initial alternatives (nothing being known about the voters' preferences about the new alternatives). It is somehow dual of another special case of the problem where the incomplete profile consists of a collection of full rankings over all alternatives for a subset of voters (nothing being known about the remaining voters' preferences), which itself is equivalent to the coalitional manipulation problem. The problem is also related to control by adding candidates [1, 11, 14, 12], as discussed in [6].

Ideally, given a voting rule, we would hope to find a polynomially-computable characterization which would allow us to quickly identify the possible (co)winners given an incomplete profile P and a number of new alternatives. Chevaleyre *et al.* [6] give such characterizations for plurality and Borda for an arbitrary number of new candidates, as well as for K-approval when there is a single new candidate. They show that these positive results do not extend to scoring rules in general, not even to K-approval, and show that computing possible (co)winners for 3-approval is NP-hard with three new candidates, as well as for some more sophisticated scoring rules, for a single new candidate. These results were further extended in [7], where a polynomial algorithm (but not an easy characterization) was proposed for 2-approval, as well as for K-approval for 2 new candidates.

The results given in [6] and [7] do not go beyond scoring rules. In this paper we go further by considering major voting rules that are outside the family of scoring rules, namely *approval*, *Bucklin*, *Copeland*, *maximin* and *plurality with runoff*. We will give two positive results, namely polynomially-computable characterization of possible (co)winners with new alternatives, for plurality with runoff and, with some specific assumptions we shall discuss later, for approval. However, for all other rules considered in this paper, we will show that finding such a characterization is hopeless, as we show that the possible (co)winner problem with new alternatives for these rules is NP-hard.

The reason why it is worth exploring the computational complexity of the possible (co)winner problem with new alternatives for various voting rules is threefold. First, it helps understanding the various possible (co)winner problems better, by comparing our results to complexity results of the possible (co)winner in the general case [21] as well as as in the specific case corresponding to the unweighted coalitional manipulation problem (see *e.g.*, [13, 15]). Second, these results help deciding which voting rules to apply

Cite as: Possible Winners When New Alternatives Join: New Results Coming Up!, Lirong Xia, Jérôme Lang and Jérôme Monnot, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 829-836.

in situations where we know beforehand that new candidates may come after the initial ones and where we want to know which of the initial ones can win the election. Voting rules for which we have an easy (polynomial) way to compute these possible winners are better suited to this class of situations. Third, on the other hand, hardness results can also be considered positive in settings where we want a voting rule to be hard to control by adding candidates *without the chair knowing a priori the voters' preferences on these candidates*. (We shall say more about this in Section 3).

We start by giving some background in Section 2. In Section 3 we recall the possible co-winner problem with respect to the addition of new alternatives (PcWNA). Each of the following sections is devoted to the PcWNA problem for a specific voting rule.

In Section 4 we focus on approval voting. Since the notion of a complete profile (including the new alternatives) extending a partial profile over the initial alternatives is not straightforward, we investigate three possible definitions, which we think are the three most reasonable definitions. To the best of our knowledge, two of these definitions are new. We show that PcWNA problems are trivial for two of these definitions, and NP-complete for the third one.

In Sections 5, 6 and 7 we show that the problem is NP-complete for, respectively, the Bucklin rule, the Copeland rule, and the maximin (a.k.a. Simpson) rule, and finally in Section 8 we focus on plurality with runoff, for which we give a polynomially computable characterization.

2. PRELIMINARIES

Let $\mathcal{C}$ be the set of *alternatives* (or candidates), with $|\mathcal{C}| = m$. Let $\mathcal{I}(\mathcal{C})$ denote the set of votes. Most often, the set of votes is the set of all linear orders over $\mathcal{C}$. An *n-profile* P is a collection of n votes for some $n \in \mathbb{N}$, that is, $P \in \mathcal{I}(\mathcal{C})^n$. A *voting rule* r is a mapping that assigns to each profile a set of winning alternatives[1], that is, r is a mapping from $\{\emptyset\} \cup \mathcal{I}(\mathcal{C}) \cup \mathcal{I}(\mathcal{C})^2 \cup \ldots$ to $2^{\mathcal{C}}$. For any profile P, the alternatives in $r(P)$ are called *co-winners* for P. If $r(P) = \{c\}$, then c is the *unique winner* for P.

Some common voting rules are listed below. For all of them (except the approval rule), $\mathcal{I}(\mathcal{C})$ is the set of all linear orders over $\mathcal{C}$; for the approval rule, the set of votes is the set of all subsets of $\mathcal{C}$, that is, $\mathcal{I}(\mathcal{C}) = \{S : S \subseteq \mathcal{C}\}$.

• *(Positional) scoring rules*: Given a *scoring vector* $\vec{v} = (v(1), \ldots, v(m))$, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $s(V, c) = v(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $s(P, c) = \sum_{i=1}^{n} s(V_i, c)$. The rule will select $c \in \mathcal{C}$ so that $s(P, c)$ is maximized. Some examples of positional scoring rules are *Borda*, for which the scoring vector is $(m-1, m-2, \ldots, 0)$; *l-approval* ($l \leq m$), for which the scoring vector is $v(1) = \ldots = v(l) = 1$ and $v_{l+1} = \ldots = v_m = 0$; and *plurality*, for which the scoring vector is $(1, 0, \ldots, 0)$.

• *Approval*: Each voter submits a set of alternatives (that is, the alternatives that are "approved" by the voter). The winner is the alternative approved by the largest number of voters. Note that the approval rule is different from the l-approval rule, in that for the l-approval rule, a voter must approve l alternatives, whereas for the approval rule, a voter can approve an arbitrary number of alternatives.

• *Bucklin*: The Bucklin score of an alternative c, denoted by $B_P(c)$ is the smallest number t such that more than half of the votes rank c among top t positions. A Bucklin winner has the lowest Bucklin score and is ranked within top $B_P(c)$ for most times.

[1]Such a function is often called a *voting correspondence* rather than a voting rule. We will however stick to the terminology "rule" throughout the paper.

• *Copeland$_\alpha$* ($0 \leq \alpha \leq 1$): For any two alternatives c_i and c_j, we can simulate a *pairwise election* between them, by seeing how many votes prefer c_i to c_j, and how many prefer c_j to c_i; the winner of the pairwise election is the one preferred more often. Then, an alternative receives one point for each win in a pairwise election, α points for each tie, and zero point for each loss. The alternatives that have the highest score win.

• *maximin (a.k.a. Simpson)*: Let $N_P(c_i, c_j)$ denote the number of votes that rank c_i ahead of c_j in P. The maximin score of alternative $c \in C$ in profile P is defined as $Sim_P(c) = \min\{N_P(c, c') : c' \in C \setminus \{c\}\}$. A maximin winner maximizes the maximin score.

• *Plurality with runoff*: The election has two rounds. In the first round, all alternatives are eliminated except the two with the highest plurality scores. In the second round (runoff), the winner is the alternative that wins the pairwise election between them. Here we use the *parallel-universe tie-breaking mechanism* [8], where an alternative c is a co-winner, if there exists a way to break ties in both rounds to make c win.

In this paper, all NP-hardness results are proved by reductions from the EXACT COVER BY 3-SETS problem (denoted by X3C) or the 3-DIMENSIONAL MATCHING problem (denoted by 3DM). An instance $I = (\mathcal{S}, \mathcal{V})$ of X3C consists of a set $\mathcal{V} = \{v_1, \ldots, v_{3q}\}$ of $3q$ elements and $t \geq q$ 3-sets $\mathcal{S} = \{S_1, \ldots, S_t\}$ of $\mathcal{V}$, i.e., for any $i \leq t$, $S_i \subseteq \mathcal{V}$ and $|S_i| = 3$. Without loss of generality, we assume that for each $v \in \mathcal{V}$, there exists $S \in \mathcal{S}$ such that $v \in S$. For any $v \in \mathcal{V}$, let $d_I(v)$ denote the number of 3-sets containing element v in instance I. Let $\Delta(I) = \max_{v \in \mathcal{V}} d_I(v)$. We are asked whether there exists a subset $J \subseteq \{1, \ldots, t\}$ such that $|J| = q$ and $\bigcup_{j \in J} S_j = \mathcal{V}$ (indeed, the sets S_j for $j \in J$ form a partition of $\mathcal{V}$). This problem is known to be NP-complete, even if $\Delta(I) \leq 3$ (problem [SP2] page 221 in [16]). In this paper, we will use a special case of 3DM that is also a special case of X3C, defined as follows.[2] Given A, B, X, where $A = \{a_1, \ldots, a_q\}$, $B = \{b_1, \ldots, b_q\}$, $X = \{x_1, \ldots, x_q\}$, $T \subseteq A \times B \times X$, $T = \{S_1, \ldots, S_t\}$ with $t \geq q$. We are asked whether there exists $M \subseteq T$ such that $|M| = q$ and for any $(a_1, b_1, x_1), (a_2, b_2, x_2) \in M$, we have $a_1 \neq a_2$, $b_1 \neq b_2$, and $x_1 \neq x_2$. That is, M corresponds to an exact cover of $\mathcal{V} = A \cup B \cup X$. This problem with the restriction where no element of $A \cup B \cup X$ occurs in more than 3 triples (i.e., $\Delta(I) \leq 3$) is known to be NP-complete (problem [SP1] page 221 in [16]).

To prove our NP-hardness results, we first prove that another useful special case of 3DM (as well as X3C) remains NP-complete.

Proposition 1 3DM *is* NP*-complete, even if q is even, $t = 3q/2$, and $\Delta(I) \leq 6$.*

PROOF. Let $I = (T, A \times B \times X)$ be an instance of 3DM with $A = \{a_1, \ldots, a_q\}$, $B = \{b_1, \ldots, b_q\}$, $X = \{x_1, \ldots, x_q\}$, $T \subseteq A \times B \times X$, $T = \{S_1, \ldots, S_t\}$ and $\Delta(I) \leq 3$. We next show how to build an instance $I' = (T', A' \times B' \times X')$ of 3DM in polynomial time, with $|A'| = |B'| = |X'| = q'$, $T' \subseteq A' \times B' \times X'$ and $|T'| = t'$ such that q' is even, $t' = 3q'/2$, and $\Delta(I') \leq 6$.

• If q is odd, then we add to the instance 3 new elements $\{a'_1, b'_1, x'_1\}$ with $A' = A \cup \{a'_1\}$, $B' = B \cup \{b'_1\}$, $X' = X \cup \{x'_1\}$ and one new triplet (a'_1, b'_1, x'_1).

• Suppose that q is even. If $t > 3q/2$, then we add $6(t - 3q/2)$ new elements $\{a'_1, \ldots, a'_{2(t-3q/2)}\}$ to A, $\{b'_1, \ldots, b'_{2(t-3q/2)}\}$ to B, $\{x'_1, \ldots, x'_{2(t-3q/2)}\}$ to X and $2(t - 3q/2)$ new triples $\{S'_1, \ldots, S'_{2(t-3q/2)}\}$, where for any $i \leq 2(t - 3q/2)$, $S'_i = (a'_i, b'_i, x'_i)$. If $t < 3q/2$, then we add $3q/2 - t$ dummy triples to T by duplicating $3q/2 - t$ triples of T once each. We note that $t \geq q$ implies that $t \geq 3q/2 - t$.

[2]Generally, 3DM is not a special case of X3C.

It is easy to check that in I', q' is even, $t' = 3q'/2$, and $\Delta(I') \leq 6$. The size of the input of the new instance is polynomial in the size of the input of the old instance. Moreover, I is a yes-instance if and only if I' is also a yes-instance. □

3. POSSIBLE (CO)WINNERS WITH NEW ALTERNATIVES

Let $\mathcal{C}$ denote the set of original alternatives, let Y denote the set of new alternatives. For any linear order V over $\mathcal{C}$, a linear order V' over $\mathcal{C} \cup Y$ *extends* V, if in V' the pairwise comparison between any pair of alternatives in $\mathcal{C}$ is the same as in V. That is, for any $c, d \in \mathcal{C}$, $c \succ_V d$ if and only if $c \succ_{V'} d$.

Given a voting rule r, an alternative c, and a profile P over $\mathcal{C}$, we are asked whether there exists a profile P' over $\mathcal{C} \cup Y$ such that P' is an extension of P and $c \in r(P')$. This problem is called the *possible co-winner with new alternatives (PcWNA)* problem [6, 7].

Similarly, we let *PWNA* denote the problem in which we are asked whether c is a possible (unique) winner, that is, $r(P') = \{c\}$. Up to now, the PcWNA and PWNA problems are well-defined for all voting rules studied in this paper (except the approval rule). For the approval rule, we will introduce three types of extension, and discuss the computational complexity of the PcWNA and PWNA problems under these extensions.

We denote by $\text{PWNA}_r(P, k)$ (respectively $\text{PcWNA}_r(P, k)$) the set of possible winners (respectively co-winners) for voting rule r and profile P with respect to the addition of k new alternatives.

It is straightforward to check that the PcWNA (respectively, PWNA) problems for all voting rules studied in this paper are in NP, because given an extension of a profile P, it takes polynomial time to verify if the given alternative c is a co-winner (respectively, the unique winner) for all rules studied in this paper. Therefore, in this paper, we do not show that PcWNA and PWNA are in NP for individual voting rules. (That is, we only show either polynomiality or NP-hardness proofs.)

Chevaleyre *et al.* [6, 7] discuss the relationship between the P(c)WNA problem and two related problems, namely control via adding candidates and candidate cloning. It is argued that the main difference between the three problems is that in the problem of control via adding candidates, the chair knows how the voters would rank the new candidates that can possibly be added by her; in the problem of candidate cloning, the chair only knows that every voter will order all the clones of a candidate contiguously in her vote, that is, every voter's preferences between a clone of c and another candidate d must be the same as her preferences between c and d; whereas in the P(c)WNA problem, the chair does not have any information about how the voters would rank the new candidates.

Even though it has been defined primarily as a problem dealing with voting with incomplete knowledge, the possible co-winner problem with new alternatives *can also be seen as a constructive control problem*, for the class of situations where the chair can add a number of new candidates without knowing how the voters will rank them: if the chair's preferred candidate x is not a co-winner for the current profile P, the chair has an incentive to add a number of new candidates for which x becomes a possible co-winner of the profile before the new alternatives are added. Of course the chair cannot guarantee that x must be a co-winner after the new alternatives are added[3], but at least x has some hope to win. The chair could find, even further, the number of new candidates k such that not only x becomes a possible co-winner, but also such that the number of possible co-winners is as low as possible.

[3]This actually corresponds to the *necessary co-winner problem*, to which the answer is trivial in the setting of this paper.

4. APPROVAL

Since the input of the approval rule is different from the input of other voting rules studied in this paper, we have to define the set of possible extensions of an approval profile over $\mathcal{C}$. Let $P_{\mathcal{C}} = (V_1, \ldots, V_n)$ be an approval profile over $\mathcal{C}$, where each V_i is a subset of $\mathcal{C}$. An extension of $P_{\mathcal{C}}$ over $\mathcal{C} \cup Y$ is a collection $(V_1', \ldots, V_n')$ where $V_i' \subseteq \mathcal{C} \cup Y$ is an extension of V_i. Now, we define what it means to say that $V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq \mathcal{C}$. We can think of three natural definitions as follows.

Definition 1 (extension of an approval vote, definition 1) *$V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq \mathcal{C}$ if $V' \cap \mathcal{C} = V$.*

In other words, under this definition, V' is an extension of V if $V' = V \cup Y'$, where $Y' \subseteq Y$. This definition coincides with the definition used in [19] (Definition 4.3) for the control of approval voting by adding candidates. The problem with Definition 1 is that it assumes that any alternative approved in V is still approved in V'. However, in some contexts, extending the choice with alternatives of Y may change the "approval threshold". Moreover, since we have more alternatives, this threshold should either stay the same or move upward: some alternatives that were approved initially may become disapproved. This leads to the following definition of extension.

Definition 2 (extension of an approval vote, definition 2) *$V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq X$ if one of the following conditions holds: (1) $V = V'$; (2) $V' \cap Y \neq \emptyset$ and $V' \cap \mathcal{C} \subseteq V$.*

Lastly, we may also allow the acceptance threshold to move downward, even though the set of alternatives grows, especially in the case where the new alternatives are particularly bad, thus rendering some alternatives in $\mathcal{C}$ acceptable after all. This leads to the third definition of extension.

Definition 3 (extension of an approval vote, definition 3) *$V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq \mathcal{C}$ if one of the following conditions holds: (1) $V' \cap \mathcal{C} \subset V$ and $V' \cap Y \neq \emptyset$; (2) $V \subset V' \cap \mathcal{C}$, and $Y \setminus V' \neq \emptyset$; (3) $V' \cap \mathcal{C} = V$.*

Under Definition 3, either the threshold moves upward, in which case all alternatives which were disapproved in V are still disapproved in V', and obviously, at least one alternative in Y should be approved; or the threshold moves downward, in which case all alternatives that were approved in V are still approved in V', and obviously not all alternatives in Y should be approved. Note that in the case where $V' \cap \mathcal{C} = V$, the threshold can move upward, or downward, or remain the same[4].

Let us give a brief summary of the three definitions of extension. Definition 1 assumes that the threshold cannot move; Definition 2 assumes that the threshold can stay the same or move upward (because the set of alternatives grows); and Definition 3 assumes that the threshold can stay the same, move upward, or move downward. Next, we show an example that illustrates these definitions. Let $\mathcal{C} = \{a, b, c, d\}$, $Y = \{y_1, y_2\}$, and $V = \{a, b\}$.

[4]The rationale behind Definition 3 is that the threshold may depend on the average quality of the alternatives, and therefore may go down after some bad new alternatives have been added. For instance, suppose a voter hates red meat, and has the preference relation `tofu` $\succ$ `fish` $\succ$ `chicken` $\succ$ `beef` $\succ$ `mutton`; if the initial set of alternatives is {`tofu, fish, chicken`}, it is perfectly reasonable that he should approve {`tofu, fish` }, while he would approve {`tofu, fish, chicken`} after `beef` and `mutton` have been added to the set of alternatives. This is perfectly in agreement with the notion of sincere ballot in approval voting (see, *e.g.*, [5, 10, 11] and references therein).

• $V_1' = \{a, b\}$ and $V_2' = \{a, b, y_1\}$ are extensions of V under all three definitions;

• $V' = \{a, y_1\}$ is an extension of V under definitions 2 and 3 but not under definition 1 (the threshold has moved upward, since b was approved in V and is no longer approved in V');

• $V' = \{a, b, c, y_1\}$ is an extension of V under definition 3 but neither under definitions 1 nor 2 (the threshold has moved downward, since c was not approved in V and becomes approved in V'—note that, intuitively, y_2 must be a very unfavorable alternative for this to happen);

• $V' = \{a, b, c\}$ is an extension of V under definitions 3 but neither under definitions 1 nor 2, for the same reason as above;

• $V' = \{a\}$ is not an extension of V under any of the definitions: to have b disapproved in V' and approved in V, the threshold has to move upward, which cannot be the case if no alternative of Y is approved;

• $V' = \{a, b, c, y_1, y_2\}$ is not an extension of V under any of the definitions: to have c disapproved in V and approved in V', the threshold has to move downward, which cannot be the case where all alternatives in Y are disapproved;

• $V' = \{a, c, y_1\}$ is not an extension of V under any of the definitions: the threshold cannot simultaneously move upward and downward.

It is straightforward to check that the PcWNA and PWNA problems are in P for approval under definition 1: an alternative $c \in \mathcal{C}$ is a possible (co-)winner in P if and only if it is a (co-)winner for approval in P (this is because for any $V \in P$, the scores of alternatives in $\mathcal{C}$ will not change from V to its extension V'). However, when we adopt definition 2 of extension, the problems become NP-complete.

Theorem 1 *Under Definition 2, the PcWNA and PWNA problems are* NP*-complete for the approval rule.*

PROOF. We first prove the hardness of the PcWNA problem by a reduction from X3C. For any X3C instance $I = (\mathcal{S}, \mathcal{V})$, we construct the following PcWNA instance.

Alternatives: $\mathcal{V} \cup \{c\} \cup Y$, where $Y = \{y_1, \ldots, y_{t-q}\}$.

Votes: for any $i \leq t$, we have a vote $V_i = S_i$; and we have an additional vote $V_{t+1} = \{c\}$. That is, $P = (V_1, \ldots, V_t, V_{t+1})$.

Suppose the X3C instance has a solution, denoted by $\{S_{i_1}, \ldots, S_{i_q}\}$. Then, take the following extension P' of P: for any $j \leq q$, let $V_{i_j}' = V_{i_j}$. For any $i \leq t$ such that $i \neq i_j$ for all $j \leq q$, we let V_i' be a singleton containing exactly one of the new alternatives. Let $V_{t+1}' = \{c\}$. For any $v \in \mathcal{V}$, because v appears exactly in one S_{i_j}, v is approved by exactly one voter. So is c. Now, there are exactly $t - q$ votes V_i where i is not equal to one of the i_j's. Therefore, the total approval score of the new alternatives is $t - q$, and it suffices to approve every new alternative exactly once. Therefore c is a co-winner in P', and thus a possible co-winner in P.

Conversely, suppose c is a possible co-winner for P and let P' be an extension of P for which c is a co-winner. We note that c is approved at most once in P'. Therefore, every alternative in $\mathcal{V} \cup Y$ must be approved at most once. Without loss of generality, assume that every vote V_i' in P' is either of the form V_i or of the form $\{y_j\}$ (if not, remove every alternative (except one y_j) from V_i'; c will still be a co-winner in the resulting profile). Since we have $t - q$ new alternatives, each being approved at most once in P', we have at least q votes V_i' in P' such that $V_i' = V_i$. If we had more than q votes V_i' such that $V_i' = V_i$, then more than $3q$ points would be distributed to $3q$ alternatives and one of them would get at least 2, which means that c would not be a co-winner in P'. Therefore we have exactly q votes V_i' such that $V_i' = V_i$, and $3q$ points distributed to $3q$ alternatives; since none of them gets more than one point, they get one point each, which implies that the collection of all S_i such that $V_i = V_i'$ forms an exact cover of C.

For the PWNA problem, we add one more vote $V_{t+2} = \{c\}$ to the profile P. □

Now, let us consider Definition 3. Notice that the profile P' where every voter adds c to her vote (if she was not already voting for c) is an extension of P, and obviously c is a co-winner in P'. Therefore, every alternative in $\mathcal{C}$ is a possible co-winner for P, which trivialize the problem.

5. BUCKLIN

Theorem 2 *The PWNA and PcWNA problems are* NP*-complete for Bucklin, even when there are three new alternatives.*

PROOF. We prove the NP-hardness of both PcWNA and PWNA by the same reduction from the special case of 3DM mentioned in Proposition 1. Given any 3DM instance where $|A| = |B| = |X| = q$, q is even, $t = 3q/2$, and no element in $A \cup B \cup X$ appears in more than 6 elements in T, we construct a PcWNA (PWNA) instance as follows. Without loss of generality, assume $q \geq 5$; otherwise the instance 3DM can be solved directly.

Alternatives: $A \cup B \cup X \cup Y \cup D \cup \{c\}$, where $Y = \{y_1, y_2, y_3\}$ is the set of new alternatives, and $D = \{d_1, \ldots, d_{9q^2}\}$ is the set of auxiliary alternatives.

Votes: For any $i \leq 2q + 1$, we define a vote V_i. Let $P = (V_1, \ldots, V_{2q+1})$. Instead of defining these votes explicitly, below we give the properties that P satisfies. The votes can be constructed in polynomial time.

(i) For any $i \leq q$, c is ranked in the first position. Suppose $S_i = (a, b, x)$. Then, let a, b, x be ranked in the $(3q + 1)$th, $(3q + 2)$th, and $(3q + 3)$th positions in V_i, respectively.

(ii) For any i such that $q < i \leq 3q/2 = t$, c is ranked in the $(3q + 4)$th position. Suppose $S_i = (a, b, x)$. Then, let a, b, x be ranked in the $(3q + 1)$th ,$(3q + 2)$th, and $(3q + 3)$th positions in V_i, respectively.

(iii) For any i such that $3q/2 < i \leq 2q + 1$, let c be ranked in the $(3q + 4)$th position, and no alternative in $A \cup B \cup X$ is ranked in the $(3q + 1)$th, $(3q + 2)$th, or $(3q + 3)$th position in V_i.

(iv) For any $c' \in A \cup B \cup X$, c' is ranked within top $3q + 3$ positions for exactly $q + 1$ times in P; and c' is never ranked in the $(3q + 4)$th position.

(v) For any $d \in D$, d is ranked within top $3q + 4$ positions at most once.

The existence of a profile P that satisfies (iv) is guaranteed by the assumption that in the 3DM instance, $q \geq 5$, no element is covered more than 6 times, and there are enough positions within top $3q+3$ positions in all votes to ensure that each alternatives in $\mathcal{C}$ appears exactly $q + 1$ times. We note that there are in total $9q^2$ auxiliary alternatives, and the total number of top $3q + 4$ positions in all votes is $(3q+4)(2q+1) < 9q^2$. Therefore, (v) can be satisfied. It follows that there exists a profile P that satisfies (i), (ii), (iii), (iv), and (v), and such a profile can be constructed in polynomial time (by first putting the alternatives to their positions defined in (i), (ii), and (iii), then filling out the positions using remaining alternatives to meet conditions (iv) and (v)). The Bucklin score of c is $3q+4$ in P. For any $j \leq q$, the Bucklin score of a_j (resp., b_j, x_j) is at most $3q+3$ in P, and for any $j \leq 9q^2$, the Bucklin score of $d_j \in D$ is at least $3q+4$ in P. Observe that the Bucklin score of any alternative cannot be decreased in any extension of P.

Suppose that the 3DM instance has a solution, denoted by $\{S_j : j \in J\}$, where $J \subseteq \{1, \ldots, t\}$. For any $j \in J$, we let V_j' be the extension of V_j in which y_1, y_2, y_3 are ranked in the $(3q + 1)$th, $(3q + 2)$th, and $(3q + 3)$th positions, respectively. For any

$j \in \{1, \ldots, 2q+1\} \setminus J$, we let V'_j be the extension of V_j where $\{y_1, y_2, y_3\}$ are ranked in the bottom positions. Let $P' = (V'_1, \ldots, V'_{2q+1})$. It follows that in P', the Bucklin score of c is $3q+4$ and c is ranked within top $3q+4$ for $3q/2$ times; the Bucklin score of any other alternative is at least $3q+4$, and none of them is ranked within top $3q+4$ for more than $q+1$ times. Therefore, c is the unique winner for Bucklin for P', which means that there is a solution to the PcWNA (PWNA) instance.

Conversely, suppose that there is a solution to the PcWNA (PWNA) instance, denoted by $P' = (V'_1, \ldots, V'_{2q+1})$. We recall that in order for c to be a co-winner, the Bucklin score of any alternative in $A \cup B \cup X$ must be at least $3q+4$ (since the Bucklin score of c cannot decrease in P'). Therefore, for every $a \in A$, there exists $i \leq t$ such that a is ranked within top $3q+3$ positions in V_i, and is ranked lower than the $(3q+3)$th position in V'_i. Consequently, in each of such V'_i, the new alternatives must be ranked within top $3q+3$ positions. Because $|A| = q$, each new alternative must be ranked within top $3q+3$ positions in $V_1, \ldots, V_t$ for q times. Because c is a co-winner, no alternative in Y is ranked within top $3q+3$ positions in P' for more than q times. Therefore, in exactly q votes in P', the alternatives in Y are ranked within top $3q+3$ positions. Let $\{V'_{i_1}, \ldots, V'_{i_q}\}$ denote these votes.

We claim that $\{S_{i_1}, \ldots, S_{i_q}\}$ is a solution to the 3DM instance. If not, then there exists $e \in B \cup X$ that does not appear in any S_{i_j}. However, it follows that e is ranked within top $3q+3$ positions for exactly q times, which means that the Bucklin score of e is at most $3q+3$. Therefore, the Bucklin score of e is lower than the Bucklin score of c. This contradicts the assumption that c is a co-winner for P'. Therefore, the PcWNA (PWNA) problem is NP-hard for Bucklin. □

6. COPELAND$_0$

For any profile P, the Copeland score of an alternative $c \in \mathcal{C}$ in profile P is denoted by $\mathrm{CS}_P(c) = |\{c' \in \mathcal{C} : N_P(c, c') > n/2\}|$ (recall that we focus on Copeland$_0$, which means that the tie in a pairwise election gives 0 point to both participating alternatives). We have the following straightforward observation.

Property 1 *For any profile P' over $\mathcal{C} \cup \{y\}$ that is an extension of profile P, the following inequalities hold:*

$$\forall c \in \mathcal{C},\ CS_P(c) \leq CS_{P'}(c) \leq CS_P(c) + 1 \qquad (1)$$

We prove that a useful restriction of X3C remains NP-complete.

Proposition 2 X3C *is* NP*-complete, even if $t = 2q-2$ and $\Delta(I) \leq 6$.*

PROOF. The proof is similar to the proof for Proposition 1. Let $I = (\mathcal{S}, \mathcal{V})$ be an instance of X3C, where $\mathcal{V} = \{v_1, \ldots, v_{3q}\}$ and $\mathcal{S} = \{S_1, \ldots, S_t\}$. We next show how to build an instance $I' = (\mathcal{S}', \mathcal{V}')$ of X3C in polynomial time, with $|\mathcal{V}'| = 3q'$ and $|\mathcal{S}'| \leq 6$ such that $t' = 2q' - 2$ and $\Delta(I') \leq 6$.

• If $t < 2q-2$, then we add $2q-2-t$ dummy 3-sets to $\mathcal{S}$ by duplicating $2q-2-t$ sets of $\mathcal{S}$ once each. It follows from $t \geq q$ that $2q-2-t \leq q-2 < t$.

• If $t > 2q-2$, then we add $3(t-2q+2)$ new elements $v'_1, \ldots, v'_{3(t-2q+2)}$ and $t-2q+2$ 3-sets $\{v'_1, v'_2, v'_3\}, \ldots, \{v'_{3(t-2q+2)-2}, v'_{3(t-2q+2)-1}, v'_{3(t-2q+2)}\}$.

The size of the input of the new instance is polynomial in the size of the input of the old instance. Moreover, I is a yes-instance if and only if I' is also a yes-instance. Finally, in the new instance I', we have: $|\mathcal{V}'| = |\mathcal{V}| = 3q$ and $t' = |\mathcal{S}'| = t + (2q-2-t) = 2q-2 = 2q'-2$ in the first case, while $3q' = |X'| = 3q + 3(t-2q+2) = 3(t-q+2)$ and $t' = |\mathcal{S}'| = t + (t-2q+2) = 2(t-q+1) = 2(q'-1)$ in the second case. Moreover, $d_{I'}(v) \leq 2d_I(v) \leq 6$ if $v \in \mathcal{V}$, and $d_{I'}(v) = 1$ if $v \in \mathcal{V}' \setminus \mathcal{V}$. □

Theorem 3 *The PcWNA problem is* NP*-complete for Copeland$_0$, even when there is one new alternative.*

PROOF. The proof is by a reduction from X3C. Let $I = (\mathcal{S}, \mathcal{V})$, where $t = 2q-2$ and $\Delta(I) \leq 6$ be an instance of X3C as described in Proposition 2. As previously, we can assume $q \geq 8$; hence $\Delta(I) \leq q-2$. For any X3C instance, we construct the following PcWNA instance for Copeland$_0$.

Alternatives: $\mathcal{V} \cup D \cup Y \cup \{c\}$, where $D = \{d_1, \ldots, d_t\}$ and $Y = \{y\}$ is the set of the new alternative.

Votes: For any $i \leq t$, we define the following $2t$ votes.

$$V_i = [d_i \succ (D \setminus \{d_i\}) \succ (\mathcal{V} \setminus S_i) \succ c \succ S_i]$$

$$V'_i = [\mathrm{rev}(S_i) \succ \mathrm{rev}(\mathcal{V} \setminus S_i) \succ \mathrm{rev}(D \setminus \{d_i\}) \succ c \succ d_i]$$

Here the elements in a set are ranked according to the order of their subscripts, i.e., if $S_i = \{v_2, v_5, v_7\}$, then the elements are ranked as $v_2 \succ v_5 \succ v_7$. For any set X such that $X \subset \mathcal{V}$ or $X \subset D$, let $\mathrm{rev}(X)$ denote the linear order where the elements in X are ranked according to the reversed order of their subscripts. For example, $\mathrm{rev}(\{v_2, v_5, v_7\}) = v_7 \succ v_5 \succ v_2$.

We also define the following $t = 2q-2$ votes.

$$W_1 = \ldots = W_{q-1} = [\mathcal{V} \succ D \succ c]$$

$$W'_1 = \ldots = W'_{q-1} = [\mathrm{rev}(D) \succ \mathrm{rev}(\mathcal{V}) \succ c]$$

Let $P = (V_1, V'_1, \ldots, V_t, V'_t, W_1, W'_1, \ldots, W_{q-1}, W'_{q-1})$.

We note that there are $3t$ votes in the instance. We recall that by assumption, $3t/2 = 3q-3$. We make the following observations on the function N_P.

• *For any $d \in D$, d beats c*: this holds because $N_P(c, d) = 1$.

• *For any $v \in \mathcal{V}$, v beats c*: this holds because $N_P(c, v) = d_I(v) \leq q-2 < 3q-3$.

• *For any $d \in D$ and $v \in \mathcal{V}$, d and v are tied*: this holds because $N_P(v, d) = t + q - 1 = 3q-3$.

• *For any $v, v' \in \mathcal{V}$ ($v' \neq v$), v and v' are tied.*

• *For any $d, d' \in D$ ($d' \neq d$), d and d' are tied.*

From these observations we have the following calculation on the Copeland scores:

• $\mathrm{CS}_P(c) = 0$.

• For any $v \in \mathcal{V}$, $\mathrm{CS}_P(v) = 1$.

• For any $d \in D$, $\mathrm{CS}_P(d) = 1$.

Now, assume that $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C; hence, there exists $J \subset \{1, \ldots, t\}$ with $|J| = q$ and $\bigcup_{j \in J} S_j = \mathcal{V}$. Next, we show how to make c a co-winner by introducing one new alternative y.

• For any $j \in J$, we let $\widetilde{V}_j = [d_j \succ D \setminus \{d_j\} \succ \mathcal{V} \setminus S_j \succ c \succ y \succ S_j]$ be the completion of V_j.

• For any $i \leq t$, we let $\widetilde{V}'_i = [\mathrm{rev}(S_i) \succ \mathrm{rev}(\mathcal{V} \setminus S_i) \succ \mathrm{rev}(D \setminus \{d_i\}) \succ c \succ y \succ d_i]$ be the completion of V'_i.

• For any vote not mentioned above, we put y in the top position.

• Finally, let P' denote the profile obtained in the above way.

It follows that y loses to c in their pairwise election, and for any other alternative $c' \in \mathcal{C}$ ($c' \neq y$ and $c' \neq c$), c' and y are tied in their pairwise election. Therefore, the Copeland score is 1 for c, any alternative in $\mathcal{V}$, and any alternative in D; the Copeland score of y is 0. It follows that c is a co-winner.

Next, we show how to convert a solution to the PcWNA instance to a solution to the X3C instance. Let $P' = (\widetilde{V}_1, \ldots, \widetilde{V}_t, \widetilde{V}'_1, \ldots, \widetilde{V}'_t, \widetilde{W}_1, \widetilde{W}'_1, \ldots, \widetilde{W}_{q-1}, \widetilde{W}'_{q-1})$ be a profile with the new alternative, such that c becomes a co-winner according to the Copeland$_0$ rule.

We denote $P_1' = (\widetilde{V}_1, \ldots, \widetilde{V}_t)$, $P_2' = (\widetilde{V}_1', \ldots, \widetilde{V}_t')$ and $P_3' = (\widetilde{W}_1, \widetilde{W}_1', \ldots, \widetilde{W}_{q-1}, \widetilde{W}_{q-1}')$. It follows from the above observations on Copeland scores of alternatives in profile P and inequalities (1) of Property 1, that $\mathrm{CS}_{P'}(c) = 1, \forall c' \in D \cup \mathcal{V}, \mathrm{CS}_{P'}(c) = 1$ and $\mathrm{CS}_{P'}(y) \leq 1$.

We now claim the following.

(a) $\forall v \in \mathcal{V}$, $N_{P'}(v, y) \leq 3q - 3$, $N_{P'}(y, c) = 3q - 2$ and $\forall d \in D, N_{P'}(d, y) = 3q - 3$. $N_{P_2'}(c, y) = t = 2q - 2$. Moreover, for any $i \leq t, c \succ y \succ d_i$ in $\widetilde{V}_i'$.

(b) $\forall v \in \mathcal{V}, N_{P_2' \cup P_3'}(v, y) \geq N_{P_2' \cup P_3'}(c, y)$.

For (a). Since c is a co-winner for P', c must beat y in their pairwise election. Meanwhile, any $c' \in \mathcal{V} \cup D$ cannot beat y in their pairwise elections. Therefore, we must have that $N_{P'}(c, y) \geq 3q - 2$, and for any $c' \in \mathcal{V} \cup D$, $N_{P'}(c', y) \leq 3q - 3$. For any $d_i \in D$, in profile P', we have that $d_i \succ c$ except in $\widetilde{V}_i'$, which means that $N_{P'}(d_i, y) \geq N_{P'}(c, y) - 1$ by transitivity in each vote. Hence, $3q - 3 \geq N_{P'}(d_i, y) \geq N_{P'}(c, y) - 1 \geq 3q - 3$, which means that $N_{P'}(d_i, y) = 3q - 3$ and $N_{P'}(c, y) = 3q - 2$. From these equalities, we deduce that $\forall d \in D$, $N_{P'}(d, y) = N_{P'}(c, y) - 1$ and then, for any $i \leq t$, we have that $c \succ y \succ d_i$ in $\widetilde{V}_i'$. It follows that $N_{P_2'}(c, y) = t = 2q - 2$.

For (b). For any $v \in \mathcal{V}$, because in any vote in $P_2' \cup P_3'$ $v \succ c$, by transitivity we have $N_{P_2' \cup P_3'}(v, y) \geq N_{P_2' \cup P_3'}(c, y)$.

Let $J = \{j \leq t : c \succ y \text{ in } \widetilde{V}_j\}$. We will prove that $|J| = q$ and $\cup_{j \in J} S_j = \mathcal{V}$. First, note that $|J| \leq q$ because $|J| = N_{P_1'}(c, y) \leq N_{P'}(c, y) - N_{P_2'}(c, y) = q$ from item (a).

Now, for any $v \in \mathcal{V}$ let $J_v = \{j \leq t : y \succ v \text{ in } \widetilde{V}_j\}$. We claim: $\forall v \in \mathcal{V}$, $J \cap J_v \neq \emptyset$. Otherwise, there exists $v^* \in \mathcal{V}$ with $J \cap J_{v^*} = \emptyset$. This means that $c \succ y$ implies $v^* \succ y$ in votes in P_1'. Hence, $N_{P_1'}(v^*, y) \geq N_{P_1'}(c, y)$. By adding this inequality with the inequality in item (b) (let $v = v^*$), we obtain that $N_{P'}(v^*, y) \geq N_{P'}(c, y)$. Now, combining the inequalities in item (a), we have that $3q - 3 \geq N_{P'}(v^*, y) \geq N_{P'}(c, y) = 3q - 2$, which is a contradiction. Therefore, for all $v \in \mathcal{V}$, $J \cap J_v \neq \emptyset$. Finally, since $|\mathcal{V}| = 3q$, $|S_i| = 3$ and $|J| \leq q$, we deduce that $|J| = q$ and $J = \{j \leq t : c \succ y \succ S_j \text{ in } \widetilde{V}_j\}$. Also, because for all $v \in \mathcal{V}$, $J \cap J_v \neq \emptyset$, we have $\bigcup_{j \in J} S_j = \mathcal{V}$. In conclusion, $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C. This completes the NP-hardness proof for the PcWNA problem for Copeland$_0$. □

7. MAXIMIN

To prove the NP-hardness of the PcWNA problem for Maximin, we first make the following observation, whose proof is straightforward.

Property 2 *Let P be a profile over $\mathcal{C}$, P' be a profile over $\mathcal{C} \cup \{y\}$ such that P' is an extension P. The following (in)equalities hold:*

(i) $\forall c \in \mathcal{C}$, $Sim_{P'}(c) = \min\{Sim_P(c), N_{P'}(c, y)\}$.

(ii) $\forall c \in \mathcal{C}$, $Sim_{P'}(c) \leq Sim_P(c)$.

Theorem 4 *PcWNA and PWNA problems are NP-complete for maximin, even when there is one new alternative.*

PROOF. We first prove the NP-hardness for the PcWNA problem by a reduction from X3C. Let $I = (\mathcal{S}, \mathcal{V})$ with $t = 2q - 2$ and $\Delta(I) \leq 6$ be an instance of X3C as described in Proposition 2. Without loss of generality, assume $q \geq 8$; in particular, we deduce $\Delta(I) \leq q - 2$. We define a PcWNA instance for maximin as follows:

Alternatives: $\mathcal{V} \cup \{c, d\} \cup \{y\}$, where y is the new alternative.

Votes: For any $i \leq t$, we define the following vote. $V_i = [(\mathcal{V} \setminus S_i) \succ d \succ c \succ S_i]$. Let $W_1 = \cdots = W_{q-1} = [c \succ \mathrm{rev}(\mathcal{V}) \succ d]$ and $W_q = [\mathrm{rev}(\mathcal{V}) \succ d \succ c]$. Let $P_1 = (V_1, \ldots, V_t)$, $P_2 = (W_1, \ldots, W_q)$, and $P = P_1 \cup P_2$.

We make the following observation on the maximin scores of the alternatives before y is added.

• $Sim_P(c) = q - 1$. Indeed, $N_P(c, d) = q - 1$ and $\forall v \in \mathcal{V}$, $N_P(c, v) = q - 1 + d_I(v) \geq q$.

• $Sim_P(d) \leq 6 \leq q - 2$. This is because for any $v \in \mathcal{V}$, v is covered by the 3-sets for no more than $q - 2$ times (the assumption of the input X3C instance), which means that in P_1, $d \succ v$ for at most $q - 2$ times, i.e., $N_P(d, v) = d_I(v) \leq 6 \leq q - 2$.

• For any $v \in \mathcal{V}$, $Sim_P(v) \geq q$. Actually, $N_P(v, d) > N_P(v, c) = t - d_I(v) + 1 \geq q$. For any $i < j \leq 3q$, $N_P(v_i, v_j) = N_{P_1}(v_i, v_j) \geq t - d_I(v) \geq 2q - 2 - (q - 2) = q$ and if $i > j$, $N_P(v_i, v_j) \geq N_{P_2}(v_i, v_j) = q$.

Now, suppose the X3C instance has a solution $J \subset \{1, \ldots, t\}$ with $|J| = q$ and $\bigcup_{j \in J} S_j = \mathcal{V}$. We show how to make c a co-winner by introducing one new alternative y.

• For any $j \in J$, we let $V_j' = [(\mathcal{V} \setminus S_j) \succ d \succ c \succ y \succ S_j]$.

• For any $j \in \{1, \ldots, t\} \setminus J$, we let $V_j' = [y \succ (\mathcal{V} \setminus S_j) \succ d \succ c \succ S_j]$.

• For any $j \leq q - 1$, we let $W_j' = [c \succ y \succ \mathrm{rev}(\mathcal{V}) \succ d]$.

• Let $W_q' = [y \succ \mathrm{rev}(\mathcal{V}) \succ d \succ c]$.

• Finally, let $P' = (V_1', \ldots, V_t', W_1', \ldots, W_q')$.

In P', the maximin score of y is $q - 1$ (via c), because $t = 2q - 2$, which means that $t - q + 1 = q - 1$; the maximin score of c is $q - 1$ (via d); the maximin score of d is no more than $q - 1$ (via any of $v \in \mathcal{V}$); and the maximin score of any $v \in \mathcal{V}$ is $q - 1$ (via y). Therefore, c is a co-winner for the maximin rule.

Next, we show how to convert a solution P' to the above PcWNA instance for the maximin rule to a solution to the X3C instance. Let $P' = (V_1', \ldots, V_t', W_1', \ldots, W_q')$ be an extension of P with one new alternative y, and c is the maximin winner for P'. Let $P_1' = (V_1', \ldots, V_t')$ and $P_2' = (W_1', \ldots, W_q')$.

We make the following observations.

(a) $\forall v \in \mathcal{V}$, $N_{P'}(v, y) \leq q - 1$,

(b) $N_{P'}(y, c) \leq q - 1$ and $N_{P'}(y, d) \geq q$,

(c) $y \succ c$ in W_q'.

For item (a): Because c is a co-winner, for any $v \in \mathcal{V}$, $Sim_{P'}(v) \leq Sim_{P'}(c)$. We recall that $Sim_P(c) = q - 1$ and $Sim_P(v) \geq q$. Thus, by Property 2 we have the following calculation.

$$\min\{N_{P'}(v, y), q\} \leq Sim_{P'}(v) \leq Sim_{P'}(c) \leq Sim_P(c) = q - 1$$

For item (b): First from (a), we deduce that for any $v \in \mathcal{V}$, $N_{P'}(y, v) \geq t + q - N_{P'}(v, y) > q$. Thus, we obtain:

$$\begin{aligned} Sim_{P'}(y) &= \min\{N_{P'}(y, c), N_{P'}(y, d)\} \\ &\leq Sim_{P'}(c) \leq Sim_P(c) = q - 1 \end{aligned} \tag{2}$$

Now, assume $N_{P'}(y, d) \leq q - 1$. Then, $N_{P_2'}(d, y) = q - N_{P_2'}(y, d) \geq q - N_{P'}(y, d) \geq 1$. Hence, there exists $i \leq q$ such that in W_i', we have that for any $v \in \mathcal{V}$, $v \succ d \succ y$. Moreover, $N_{P_1'}(d, y) = t - N_{P_1'}(y, d) \geq 2q - 2 - (q - 1) = q - 1$. Let $J_0 \subseteq \{1, \ldots, t\}$ (with $|J_0| = q - 1$) be the subscripts of arbitrary $q - 1$ votes in P_1', where $d \succ y$. Because $|\mathcal{V}| = 3q$ and $|S_j| = 3$, there exists $v^* \in \mathcal{V} \setminus \bigcup_{j \in J_0} S_j$. We deduce that for all $j \in J_0$, $v^* \succ y$ in V_j'. In conclusion, $N_{P'}(v^*, y) \geq |J_0| + 1 = q$, which contradicts item (a). Using inequality (2), item (b) follows.

For item (c): Otherwise, by the definition of W_q, we deduce:

$$\forall v \in \mathcal{V},\ N_{P_2'}(v, y) \geq 1 \tag{3}$$

On the other hand, using $N_{P_1'}(y, c) \leq N_{P'}(y, c)$ and item (b), we have $N_{P_1'}(c, y) = t - N_{P_1'}(y, c) \geq t - N_{P'}(y, c) \geq t - (q -$

$1) = q-1$. Let $J_0 \subseteq \{1,\ldots,t\}$ (with $|J_0| = q-1$) be the subscripts of arbitrary $q-1$ votes in P_1', where $c \succ y$. We have $\mathcal{V} \setminus \bigcup_{j\in J_0} S_j \neq \emptyset$ since $|\mathcal{V}| = 3q$ and $|S_i| = 3$. Hence, there exists $v^* \in \mathcal{V} \setminus \bigcup_{j\in J_0} S_j$ such that:

$$N_{P_1'}(v^*, y) \geq |J_0| = q-1 \tag{4}$$

Summing up inequalities (3) (let $v = v^*$) and (4), we reach a contradiction with item (a).

From items (b) and (c), we get

$$N_{P_1'}(y,c) = N_{P'}(y,c) - N_{P_2'}(y,c) \leq q-1-1 = q-2$$

Thus, $N_{P_1'}(c,y) = t - N_{P_1'}(y,c) \geq t-(q-2) = q$. Let J denote the subscripts of arbitrary q votes in P_1' where $c \succ y$. We claim $\bigcup_{j\in J} S_j = \mathcal{V}$. Otherwise, there exists $v^* \in \mathcal{V} \setminus \bigcup_{j\in J} S_j$. It follows that for any $j \in J$, $v^* \in (\mathcal{V} \setminus \bigcup_{j\in J} S_j) \subseteq \mathcal{V} \setminus S_j$, which means that $v^* \succ c \succ y$ in V_j. Hence, $N_{P'}(v^*, y) \geq N_{P_1'}(v^*, y) \geq |J| = q$, which contradicts item (a). In conclusion, $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C. Therefore, PcWNA is NP-complete for maximin.

For the PWNA problem, we make the following change. Let $W_q = [\text{rev}(\mathcal{V}) \succ c \succ d]$. Then, before the new alternative is introduced, the maximin score of c is q. Then, similarly we can prove the NP-hardness of the PWNA problem. □

8. PLURALITY WITH RUNOFF

In this section, we adopt the parallel-universe tie-breaking. If a tie occurs in the first round, then all possible compatible second rounds are considered: for instance, if the plurality scores, ranked in decreasing order, are $x_1 \mapsto 8, x_2 \mapsto 6, x_3 \mapsto 6, x_4 \mapsto 5\ldots$, then the set of co-winners contains the majority winner between x_1 and x_2 and the majority winner between x_1 and x_3. We show a necessary and sufficient condition for a given alternative c to be a possible co-winner with new alternatives for plurality with runoff. This condition can be easily converted to a polynomial-time algorithm that computes PcWNA for plurality with runoff. For any profile P and any alternative $x \in \mathcal{C}$, we let $S^P(x)$ denote the plurality score of x in P, that is, the number of times where x is ranked in the first position in votes in P. We let $X_P^-(c)$ denote the set of alternatives that lose to c in their pairwise elections, and let $X_P^+(c) = \mathcal{C} \setminus (X_P^-(c) \cup \{c\})$.

Proposition 3 *For any profile P and any alternative c, c is a possible co-winner with k new alternatives under P for plurality with runoff, if and only if one of the two following conditions holds:*

1. *there exists an alternative $d \in X_P^-(c)$ such that $\sum_{x\in\mathcal{C}\setminus\{c,d\}} \max(0, S^P(x)-\theta) \leq k\theta$, where $\theta = \min\left(S^P(d), S^P(c)\right)$.*
2. *$\sum_{x\in\mathcal{C}\setminus\{c\}} \max(0, S^P(x)-S^P(c)) \leq \lfloor n/2\rfloor + (k-1)S^P(c)$.*

PROOF. Let $P = (V_1,\ldots,V_n)$ be a profile over $\mathcal{C}$ and $P' = (V_1',\ldots,V_n')$ be a completion of P with k new alternatives. c is a co-winner in P' if one of the following conditions hold:

1. c and $d \in \mathcal{C}\setminus\{c\}$ are possible second round competitors, and c (weakly) beats d in their pairwise election under P'.
2. c and $y \in Y$ are possible second round competitors, and c (weakly) beat y in their pairwise election under P'.

Let $\succeq_M^P$ denote a weak majority relations under P, defined as follows. For any pair of alternatives a, b, $a \succeq_M^P b$ if at least half of the voters in P prefers a to b. $\succeq_M^{P'}$ is defined similarly. Let us first analyze the situations in which 1 occurs. First, in order to have $c \succeq_M^{P'} d$ we must have $c \succeq_M^P d$ (because the relative positions of c and d are the same in V_i and V_i'). Thus, 1 occurs if and only there exists an alternative d that loses to c in their pairwise elections and such that c and d can compete in the second round. Fix such d. In order for c and d to be possible second round competitors, we must have $\min(S^{P'}(c), S^{P'}(d)) \geq S^{P'}(x)$ for every $x \in \mathcal{C}\setminus\{c,d\}\cup Y$. Without loss of generality, we can assume that the scores of c and d are the same in P and P', and similarly for the scores of any $x \in \mathcal{C}$ such that $S^P(\mathcal{C}) \leq \min(S^P(c), S^P(d))$, since these alternatives do not need to lose any point to allow a possible second round between c and d. Let $\hat{\mathcal{C}}_{c,d}$ be the set of all candidates x in $\mathcal{C}\setminus\{c,d\}$ such that $S^P(x) > \min(S^P(c), S^P(d))$. Each candidate $x \in \hat{\mathcal{C}}_{c,d}$ has to lose at least $S^P(x) - \min(S^P(c), S^P(d))$ points, and for this we need $\sum_{x\in\hat{\mathcal{C}}_{c,d}} S^P(x) - \min(S^P(c), S^P(d))$ points to be given to the new candidates. Therefore, to have c and d (possibly) in the second round, the number of points we must distribute to new candidates is $\sigma = \sum_{x\in\mathcal{C}\setminus\{c,d\}} \max(0, S^P(x)-\theta)$, where $\theta = \min(S^P(c), S^P(d))$. Now, we also need the score of any new alternative y to be at most θ, therefore we need $\sigma \leq k\theta$. This leads to the condition 1 in the statement of the proposition.

Now, let us analyze the conditions allowing condition 2 to occur. In order to have c in the second round and none of the alternatives in $\mathcal{C}\setminus\{c\}$ enter the second round, we need to distribute $\kappa = \sum_{x\in\mathcal{C}\setminus\{c\}} \max(0, S^P(x)-\theta)$ points to the candidates in $\mathcal{C}$. Let y^* be the new alternative that enters the second round together with c. y^* can take at most $\lfloor n/2\rfloor$ points, otherwise y^* will beat c in their pairwise election. For any other new alternative y' can take at most $S^P(c)$ points. Therefore, we must have that

$$\kappa \leq \lfloor n/2\rfloor + (k-1)S^P(c)$$

It is straightforward that if the above equation holds, then there exists a way to extend P to P' with k new alternatives such that c is the winner for plurality with runoff. This leads to condition 2 in the statement of the proposition.

Therefore, c is a PcWNA if and only if one of the two conditions in the statement of the proposition holds. □

Example 1 *Let P be the following 4-candidate, 18-voter profile: 4 votes of $a \succ b \succ c \succ d$, 3 votes of $b \succ a \succ c \succ d$, 7 votes of $d \succ a \succ c \succ b$, 2 votes of $d \succ c \succ b \succ a$ and 2 votes of $c \succ a \succ b \succ d$. We want to determine if c is a possible co-winner with k new alternatives for plurality with runoff. Note that $X_P^-(c) = \{b,d\}$. For condition 1 to be satisfied, it suffices to consider d as the competitor for c. Then, $\theta = 2$ and condition 1 is satisfied if $3 \leq 2k$,* i.e.*, $k \geq 2$. For condition 2 to be satisfied, we have $\sum_{x\in\mathcal{C}\setminus\{c\}} \max(0, S^P(x) - S^P(c)) = 10$, $\lfloor n/2\rfloor = 9$. Therefore, condition 2 is satisfied if and only if $k \geq 2$. It follows that as soon as we have at least two new candidates, c is a possible co-winner.*

We also obtain a similar proposition for PWNA, whose proof is similar to the proof of Proposition 3, therefore is omitted.

Proposition 4 *For any profile P and any alternative c, c is a possible winner with k new alternatives under P for plurality with runoff, if and only if one of the three following conditions holds:*

1. there exists an alternative $d \in X_P^-(c)$ such that $S^P(d) \geq S^P(c)$ and $\sum_{x\in\mathcal{C}\setminus\{c,d\}} \max(0, S^P(x)-S^P(c)+1) \leq k(S^P(c)-1)$;

2. there exists an alternative $d \in X_P^-(c)$ such that $S^P(d) < S^P(c)$ and $\sum_{x\in X_P^-(c)\setminus\{d\}} \max(0, S^P(x) - S^P(d))$ $+ \sum_{x\in X_P^+(c)} \max(0, S^P(x) - S^P(d) + 1) \leq kS^P(d)$;

3. $\sum_{x \in \mathcal{C} \setminus \{c\}} \max(0, S^P(x) - S^P(c) + 1) \leq \lfloor (n+1)/2 \rfloor + (k-1)(S^P(c) - 1)$.

Corollary 1 *Determining whether $c \in \mathcal{C}$ is a possible (co-)winner for plurality with runoff is in* P.

9. CONCLUSION

In this paper we have gone beyond existing results on the complexity of the possible (co-)winner problem with new alternatives. While [6, 7] focused on scoring rules, we have identified three new rules for which the PcWNA problem is NP-complete (Bucklin, Copeland, and maximin). We also showed that the PcWNA problem has a polynomial time algorithm for plurality with runoff, and as far as approval voting is concerned, we examined three definitions of the extension of a profile to new alternatives and showed that depending on which definition we chose, the problem can be trivial or NP-complete. Our NP-completeness proofs and algorithms for the PcWNA problems, except for Copeland$_0$, can be extended to the PWNA problems for approval, Bucklin, maximin, and plurality with runoff. The results are summarized in the following table. These results can be compared with results for

Voting rule	PcWNA	PWNA
Borda	P [7]	
2-approval	P [7]	
l-approval, $l \geq 3$	NP-complete [2] [7]	
Approval	P (Def. 1) NP-complete (Def. 2) Trivial (Def. 3)	
Bucklin	NP-complete [2]	
Copeland$_0$	NP-complete [3]	?
maximin	NP-complete [3]	
Plurality with runoff	P	

Table 1: Complexity of PcWNA and PWNA problems for some common voting rules.

control by adding candidates and cloning. Control by adding candidates is NP-complete for most of voting rules considered here, namely Copeland [14], maximin [12], Borda, plurality with runoff and l-approval for $l \geq 2$ [9]; on the other hand, approval voting is immune to control by adding candidates [17]. Manipulability by cloning with positive probability (0-cloning) is polynomial for Borda, maximin and plurality with runoff, and NP-complete for Copeland and l-approval for $l \geq 2$ [9]. This shows that P(c)WNA, when viewed as a control problem, shows a resistance to strategic behaviour globally stronger than cloning and weaker than control by adding candidates.

An obvious and interesting direction for future research is studying the computational complexity of the PcWNA (PWNA) problems for more common voting rules, including STV, Copeland$_\alpha$ (for some $\alpha \neq 0$), ranked pairs, and voting trees. Even for Copeland$_0$, the complexity of the PWNA problem still remains open. Moreover, viewing P(c)WNA problem as a control problem where the chair can add new candidates but do not know the preferences of the voters over the new candidates, it is interesting to know which voting rules are more resistant to this type of control from a non-computational viewpoint.

Acknowledgements

Lirong Xia is supported by a James B. Duke Fellowship and NSF under award number IIS-0812113. Jérôme Lang thanks the ANR project ComSoc (ANR-09-BLAN-0305). We also thank the AAMAS-11 reviewers, as well as the COMSOC-10 reviewers who reviewed a preliminary version, for their helpful comments.

[2]Even with 3 new alternatives.

[3]Even with 1 new alternative.

10. REFERENCES

[1] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Social Choice and Welfare*, 16(8-9):27–40, 1992.

[2] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. In *Proc. of ECAI-10*, 2010.

[3] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. In *Proc. of MFCS-09*, 2009.

[4] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. of IJCAI-09*, pages 53–58, 2009.

[5] S. J. Brams and M. R. Sanver. Critical strategies under approval voting: Who gets ruled in and ruled out. *Electoral Studies*, 25(2):287–305, 2006.

[6] Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible winners when new candidates are added: the case of scoring rules. In *Proc. of AAAI-10*, 2010.

[7] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. Technical report, Cahiers du LAMSADE 302, Université Paris-Dauphine, 2010.

[8] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proc. of IJCAI-09*, pages 109–115, 2009.

[9] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proc. of AAAI-10*, 2010.

[10] U. Endriss. Vote manipulation in the presence of multiple sincere ballots. In *Proc. of TARK-07*, pages 125–134, 2007.

[11] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting broadly resists control. In *Proc. of MFCS-08*, pages 311–322, 2008.

[12] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Multimode control attacks on elections. In *Proc. of IJCAI-09*, pages 128–133, 2009.

[13] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53:74–82, 2010.

[14] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and copeland voting computationally resist bribery and constructive control. *JAIR*, 35(1):275–341, 2009.

[15] P. Faliszewski and A. D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31:53–64, 2010.

[16] M. Garey and D. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. Freeman, 1979.

[17] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5-6):255–285, 2007.

[18] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[19] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *JAIR*, 33:149–178, 2008.

[20] M. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proc. of IJCAI-07*, pages 1464–1469, 2007.

[21] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proc. of AAAI-08*, pages 196–201, 2008.

The Complexity of Voter Partition in Bucklin and Fallback Voting: Solving Three Open Problems

Gábor Erdélyi *
Nanyang Technological University
Singapore
erdelyi@cs.uni-duesseldorf.de

Lena Piras and Jörg Rothe
Heinrich-Heine-Universität Düsseldorf,
40225 Düsseldorf, Germany
{piras, rothe}@cs.uni-duesseldorf.de

ABSTRACT

Electoral control models ways of changing the outcome of an election via such actions as adding/deleting/partitioning either candidates or voters. These actions modify an election's participation structure and aim at either making a favorite candidate win ("constructive control") or prevent a despised candidate from winning ("destructive control"). To protect elections from such control attempts, computational complexity has been used to show that electoral control, though not impossible, is computationally prohibitive. Recently, Erdélyi and Rothe [10] proved that Brams and Sanver's fallback voting [5], a hybrid voting system that combines Bucklin with approval voting, is resistant to each of the standard types of control except five types of voter control. They proved that fallback voting is vulnerable to two of those control types, leaving the other three cases open.

We solve these three open problems, thus showing that fallback voting is resistant to all standard types of control by partition of voters—which is a particularly important and well-motivated control type, as it models "two-district gerrymandering." Hence, fallback voting is not only fully resistant to candidate control [10] but also fully resistant to constructive control, and it displays the broadest resistance to control currently known to hold among natural voting systems with a polynomial-time winner problem. We also show that Bucklin voting behaves almost as good in terms of control resistance. Each resistance for Bucklin voting strengthens the corresponding control resistance for fallback voting.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Economics, Theory

Keywords

Voting protocols, Social Choice Theory, Computational Social Choice, Bucklin voting, fallback voting

*Work done in part at HHU Düsseldorf.

Cite as: The Complexity of Voter Partition in Bucklin and Fallback Voting: Solving Three Open Problems, Gábor Erdélyi, Lena Piras, and Jörg Rothe, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011),* Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 837-844.

1. INTRODUCTION

Elections have been used for preference aggregation not only in the context of politics and human societies, but also in artificial intelligence, especially in multiagent systems, and other topics in computer science (see, e.g., [8, 14, 7]). That is why it is important to study the computational properties of voting systems. In particular, complexity can be used to protect elections against tampering attempts in control, manipulation, and bribery attacks by showing that such attacks, though not impossible in principle, can be computationally prohibitive.

Since the seminal paper of Bartholdi et al. [2], the complexity of *electoral control*—changing the outcome of an election via such actions as adding/deleting/partitioning either candidates or voters—has been studied for a variety of voting systems. Unlike *manipulation* [1, 6], which models attempts of strategic voters to influence the outcome of an election via casting insincere votes, control models ways of an external actor, the "chair," to tamper with an election's participation structure so as to alter its outcome. Another way of tampering with the outcome of elections is *bribery* [11, 12], which shares with manipulation the feature that votes are being changed, and with control the aspect that an external actor tries to change the outcome of the election. For more background on complexity results for control, manipulation, and bribery in approval voting and its variants, we refer to the survey of Baumeister et al. [3].

Regarding control, a central question is to find voting systems that are computationally resistant to as many of the common 22 control types as possible, where resistance means the corresponding control problem is NP-hard. Each control type is either constructive (the chair seeking to make some candidate win) or destructive (the chair seeking to make some candidate end up not winning). Erdélyi and Rothe [10] recently proved that fallback voting [5], a hybrid voting system combining Bucklin with approval voting, is resistant to each of these 22 standard control types except five types of voter control. They proved that fallback voting is vulnerable to two of those control types (i.e., these control problems are polynomial-time solvable), leaving the other three cases open. We solve these three open problems by showing that fallback voting is resistant to constructive and destructive control by partition of voters in the tie-handling model "ties promote" and to destructive control by partition of voters in the "ties eliminate" model. Partition of voters is a particularly important and well-motivated control type, as it models "two-district gerrymandering." Control-by-partition cases are the most difficult control types to deal with; their

resistance proofs require the most involved constructions.

Thus fallback voting is fully resistant not only to candidate control [10] but also to constructive control. In terms of the total number of proven resistances it even outnumbers "sincere-strategy preference-based approval voting" (SP-AV, a modification [9] of another hybrid system proposed by Brams and Sanver [4]): Fallback voting has the most (20 out of 22) proven resistances to control among natural voting systems with a polynomial-time winner problem. Among such systems, only SP-AV (with its 19 proven control resistances [9]) and plurality voting were previously known to be fully resistant to candidate control [2, 15], and only Copeland voting and SP-AV were previously known to be fully resistant to constructive control [12, 9]. However, plurality has fewer resistances to voter control, Copeland voting has fewer resistances to destructive control, and SP-AV is missing one destructive voter partition resistance and—perhaps more importantly—is arguably less natural a system than fallback voting, since in SP-AV (as modified by Erdélyi and Rothe [9]) it may happen that votes are rewritten to ensure admissibility (for further details see [3, 9]).

We also study the control complexity of Bucklin voting itself and show that it has (at least) 19 resistances to control, thus drawing level with SP-AV. In particular, also Bucklin voting is—like SP-AV and fallback voting—fully resistant to constructive control and to candidate control. Since Bucklin voting is a special case of fallback voting, each resistance result for Bucklin strengthens the corresponding resistance result for fallback voting.

2. PRELIMINARIES

Elections and Voting Systems.

An *election* (C, V) is given by a finite set C of candidates and a finite list V of votes over C. A *voting system* is a rule that specifies how to determine the winner(s) of any given election. The two voting systems considered in this paper are Bucklin voting and fallback voting.

In *Bucklin voting*, votes are represented as linear orders over C, i.e., each voter ranks all candidates according to his or her preferences. For example, if $C = \{a, b, c, d\}$ then a vote might look like $c \;\; d \;\; a \;\; b$, i.e., this voter (strictly) prefers c to d, d to a, and a to b. Given an election (C, V) and a candidate $c \in C$, define the *level i score of c in* (C, V) (denoted by $score^{i}_{(C,V)}(c)$) as the number of votes in V that rank c among their top i positions. Denoting the *strict majority threshold for a list V of voters* by $maj(V) = \lfloor \|V\|/2 \rfloor + 1$, the *Bucklin score of c in* (C, V) is the smallest i such that $score^{i}_{(C,V)}(c) \geq maj(V)$. All candidates with a smallest Bucklin score, say k, and a largest level k score are the *Bucklin winners (BV winners, for short) in* (C, V). If some candidate becomes a Bucklin winner on level k, we call him or her a *level k BV winner in* (C, V). Note that a level 1 BV winner must be unique, but there may be more level k BV winners than one for $k > 1$, i.e., an election may have more than one Bucklin winner in general.

Brams and Sanver [5] proposed fallback voting as a hybrid voting system that combines Bucklin with approval voting. In *approval voting*, votes are represented by approval vectors in $\{0, 1\}^{\|C\|}$ (with respect to a fixed order of the candidates in C), where 0 stands for disapproval and 1 stands for approval. Given an election (C, V) and a candidate $c \in C$, define the *approval score of c in* (C, V) (denoted by $score_{(C,V)}(c)$) as the number of c's approvals in (C, V), and all candidates with a largest approval score are the *approval winners in* (C, V). Note that an election may have more than one approval winner. *Fallback voting* combines Bucklin with approval voting as follows. Each voter provides both an approval vector and a linear ordering of all approved candidates. For simplicity, we will omit the disapproved candidates in each vote. For example, if $C = \{a, b, c, d\}$ and a voter approves of a, c, and d but disapproves of b, and prefers c to d and d to a, then this vote will be written as: $c \;\; d \;\; a$. We will always explicitly state the candidate set, so it will always be clear which candidates participate in an election and which of them are disapproved by which voter (namely those not occurring in his or her vote). Given an election (C, V) and a candidate $c \in C$, the notions of *level i score of c in* (C, V) and *level k fallback voting winner (level k FV winner, for short) in* (C, V) are defined analogously to the case of Bucklin voting, and if there exists a level k FV winner for some $k \leq \|C\|$, he or she is called a *fallback winner (FV winner, for short) in* (C, V). However, unlike in Bucklin voting, in fallback voting it may happen that no candidate reaches a strict majority for any level, due to voters being allowed to disapprove of (any number of) candidates, so it may happen that for no $k \leq \|C\|$ a level k FV winner exists. In such a case, every candidate with a largest (approval) score is an *FV winner in* (C, V). Note that Bucklin voting is the special case of fallback voting where each voter approves of all candidates. As a notation, when a vote contains a subset of the candidate set, such as $c \;\; D \;\; a$ for a subset $D \subseteq C$, this is a shorthand for $c \;\; d_1 \;\; \cdots \;\; d_\ell \;\; a$, where the elements of $D = \{d_1, \ldots, d_\ell\}$ are ranked with respect to some (tacitly assumed) fixed ordering of all candidates in C. For example, if $C = \{a, b, c, d\}$ is assumed to be ordered lexicographically and $D = \{b, d\}$ then "$c \;\; D \;\; a$" is a shorthand for $c \;\; b \;\; d \;\; a$.

Types of Electoral Control.

There are eleven types of electoral control, each coming in two variants. In *constructive control* [2], the chair tries to make his or her favorite candidate win; in *destructive control* [15], the chair tries to prevent a despised candidate's victory. We refrain from giving a detailed discussion of natural, real-life scenarios for each of these 22 standard control types that motivate them; these can be found in, e.g., [2, 15, 12, 16, 9, 3]. However, we stress that every control type is motivated by an appropriate real-life scenario.

When we define our 22 standard control types as decision problems, we assume that each election or subelection in these control problems will be conducted with the voting system at hand (i.e., either Bucklin or fallback voting) and that each vote will be represented as required by the corresponding voting system. We also assume that the chair has complete knowledge of the voters' preferences and/or approval strategies. This assumption may be considered to be unrealistic in certain settings, but is reasonable and natural in certain others, including small-scale elections among humans and even large-scale elections among software agents. More to the point, assuming the chair to have complete information makes sense for our results, as most of our results are NP-hardness lower bounds showing resistance of a voting system against specific control attempts and complexity lower bounds in the complete-information model are inherited by any natural partial-information model (see [15] for a more detailed discussion of this point).

All our decision problems are formally described in the standard Instance-Question format. As an explicit example, we define the decision problem corresponding to control by partition of voters with the tie-handling rule "ties promote" (TP), see [15]. This control type produces a two-stage election with two first-stage and one final-stage subelections. The constructive variant of this problem is defined as:

CONSTRUCTIVE CONTROL BY PARTITION OF VOTERS (TP)

Instance: A set C of candidates, a list V of votes over C, and a designated candidate $c \in C$.

Question: Can V be partitioned into V_1 and V_2 such that c is the unique winner of the two-stage election in which the winners of the two first-stage subelections, (C, V_1) and (C, V_2), run against each other in the final stage?

The destructive variant of this problem is defined analogously, except it asks whether c is *not* a unique winner of this two-stage election. In both variants, if one uses the tie-handling model TE ("ties eliminate," see [15]) instead of TP in the two first-stage subelections, a winner w of (C, V_1) or (C, V_2) proceeds to the final stage if and only if w is the only winner of his or her subelection. Each of the four problems just defined can be seen as a way of modeling "two-district gerrymandering."

There are many ways of introducing new voters into an election—think, e.g., of "get-out-the-vote" drives, or of lowering the age-limit for the right to vote, or of attracting new voters with certain promises or even small gifts), and such scenarios are modeled as CONSTRUCTIVE/DESTRUCTIVE CONTROL BY ADDING VOTERS: Given a set C of candidates, two disjoint lists of votes over C (one list, V, corresponding to the already registered voters and the other list, W, corresponding to the as yet unregistered voters whose votes may be added), a designated candidate $c \in C$, and a nonnegative integer k, is there a subset $W' \subseteq W$ such that $\|W'\| \leq k$ and c is (is not) the unique winner in $(C, V \cup W')$?

Disenfranchisement and other means of voter suppression is modeled as CONSTRUCTIVE/DESTRUCTIVE CONTROL BY DELETING VOTERS: Given a set C of candidates, a list V of votes over C, a designated candidate $c \in C$, and a nonnegative integer k, can one make c the unique winner (not a unique winner) of the election resulting from deleting at most k votes from V?

Having defined these eight standard types of voter control, we now turn to the 14 types of candidate control. Now, the control action seeks to influence the outcome of an election by either adding, deleting, or partitioning the candidates, again for both the constructive and the destructive variant.

In the adding candidates cases, we distinguish between adding, from a given pool of spoiler candidates, an *unlimited* number of such candidates (as originally defined by Bartholdi et al. [2]) and adding a *limited* number of spoiler candidates (as defined by Faliszewski et al. [12], to stay in sync with the problem format of control by deleting candidates and by adding/deleting voters). CONSTRUCTIVE/DESTRUCTIVE CONTROL BY ADDING (A LIMITED NUMBER OF) CANDIDATES, is defined as follows: Given two disjoint candidate sets, C and D, a list V of votes over $C \cup D$, a designated candidate $c \in C$, and a nonnegative integer k, can one find a subset $D' \subseteq D$ such that $\|D'\| \leq k$ and c is (is not) the unique winner in $(C \cup D', V)$? The "unlimited" version of the problem is the same, except that the addition limit k and the requirement "$\|D'\| \leq k$" are being dropped, so *any* subset of the spoiler candidates may be added.

CONSTRUCTIVE/DESTRUCTIVE CONTROL BY DELETING CANDIDATES is defined by: Given a set C of candidates, a list V of votes over C, a designated candidate $c \in C$, and a nonnegative integer k, can one make c the unique winner (not a unique winner) of the election resulting from deleting at most k candidates (other than c in the destructive case) from C?

Finally, we define the partition-of-candidate cases, again using either of the two tie-handling models, TP and TE, but now we define these scenarios with and without a run-off. The variant with run-off, CONSTRUCTIVE/DESTRUCTIVE CONTROL BY RUN-OFF PARTITION OF CANDIDATES, is analogous to the partition-of-voters control type: Given a set C of candidates, a list V of votes over C, and a designated candidate $c \in C$, can C be partitioned into C_1 and C_2 such that c is (is not) the unique winner of the two-stage election in which the winners of the two first-stage subelections, (C_1, V) and (C_2, V), who survive the tie-handling rule run against each other in the final stage? The variant without run-off is the same, except that the winners of first-stage subelection (C_1, V) who survive the tie-handling rule run against all members of C_2 in the final round (and not only against the winners of (C_2, V) surviving the tie-handling rule). As an example, think of a sports tournament in which certain teams (such as last year's champion and this year's hosting team) are given an exemption from qualification.

Immunity, Susceptibility, Resistance, Vulnerability.

Let $\mathfrak{CT}$ be a control type. We say a voting system is *immune to* $\mathfrak{CT}$ if it is impossible for the chair to make the given candidate the unique winner in the constructive case (not a unique winner in the destructive case) via exerting control of type $\mathfrak{CT}$. We say a voting system is *susceptible to* $\mathfrak{CT}$ if it is not immune to $\mathfrak{CT}$. A voting system that is susceptible to $\mathfrak{CT}$ is said to be *vulnerable to* $\mathfrak{CT}$ if the control problem corresponding to $\mathfrak{CT}$ can be solved in polynomial time, and is said to be *resistant to* $\mathfrak{CT}$ if the control problem corresponding to $\mathfrak{CT}$ is NP-hard. These notions are due to Bartholdi et al. [2] (except that we follow the now more common approach of Hemaspaandra et al. [16] who define *resistant* to mean "susceptible and NP-hard" rather than "susceptible and NP-complete").

Fallback voting is susceptible to each of our 22 control types [10]. It is easy to see that the same holds true for Bucklin voting. The proof is omitted.

LEMMA 2.1. *Bucklin voting is susceptible to each of the 22 control types defined in this section.*

3. PARTITION OF VOTERS IN BV AND FV

Table 1 shows in boldface our results on the control complexity of fallback voting for three cases of voter partition (the other results for fallback voting being due to Erdélyi and Rothe [10]) and of Bucklin voting for all 22 standard control types. For comparison, this table also shows the results for approval voting due to Hemaspaandra et al. [15], and for SP-AV due to Erdélyi et al. [9].

In this section, we solve the three questions left open in [10]. We start with the proof that fallback voting is resistant to constructive control by partition of voters in model TP (see Corollary 3.2). We do so by proving in Theorem 3.1 that even Bucklin voting is resistant to this type of

Control by	Fallback Voting		Bucklin Voting		SP-AV		Approval	
	Const.	Dest.	Const.	Dest.	Const.	Dest.	Const.	Dest.
Adding Candidates (unlimited)	R	R	**R**	**R**	R	R	I	V
Adding Candidates (limited)	R	R	**R**	**R**	R	R	I	V
Deleting Candidates	R	R	**R**	**R**	R	R	V	I
Partition of Candidates	TE: R TP: R	TE: R TP: R	**TE: R** **TP: R**	**TE: R** **TP: R**	TE: R TP: R	TE: R TP: R	TE: V TP: I	TE: I TP: I
Run-off Partition of Candidates	TE: R TP: R	TE: R TP: R	**TE: R** **TP: R**	**TE: R** **TP: R**	TE: R TP: R	TE: R TP: R	TE: V TP: I	TE: I TP: I
Adding Voters	R	V	**R**	**V**	R	V	R	V
Deleting Voters	R	V	**R**	**V**	R	V	R	V
Partition of Voters	TE: R **TP: R**	**TE: R** **TP: R**	**TE: R** **TP: R**	**TE: R** **TP: S**	TE: R TP: R	TE: V TP: R	TE: R TP: R	TE: V TP: V

Table 1: Overview of results. Key: I = immune, S = susceptible, R = resistant, V = vulnerable, TE = ties eliminate, and TP = ties promote. Results new to this paper are in boldface.

control. As our reduction works also for the TE tie-handling model, this strengthens the corresponding result for fallback voting from [10].

Our reductions in the proof of Theorem 3.1 are from the NP-complete problem Exact Cover by Three-Sets, which is defined as follows (see, e.g., [13]):

Exact Cover by Three-Sets (X3C)

Instance: A set $B = \{b_1, b_2, \ldots, b_{3m}\}$, $m \geq 1$, and a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$ with $\|S_i\| = 3$ for each i, $1 \leq i \leq n$.

Question: Is there a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that each element of B occurs in exactly one set in $\mathcal{S}'$?

Theorem 3.1. *Bucklin voting is resistant to constructive control by partition of voters in both model TE and model TP.*

Proof. Susceptibility holds by Lemma 2.1. To show NP-hardness we reduce X3C to our control problems. Let $(B, \mathcal{S})$ be an X3C instance with $B = \{b_1, b_2, \ldots, b_{3m}\}$, $m \geq 1$, and a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$ with $\|S_i\| = 3$ for each i, $1 \leq i \leq n$. We define the election (C, V), where $C = B \cup \{c, w, x\} \cup D \cup E \cup F \cup G$ is the set of candidates with $D = \{d_1, \ldots, d_{3nm}\}$, $E = \{e_1, \ldots, e_{(3m-1)(m+1)}\}$, $F = \{f_1, \ldots, f_{(3m+1)(m-1)}\}$, and $G = \{g_1, \ldots, g_{n(3m-3)}\}$, and where w is the distinguished candidate. Let V consist of the following $2n + 2m$ voters:

1. For each $i, 1 \leq i \leq n$, there is one voter of the form: $c \;\; S_i \;\; G_i \;\; (G - G_i) \;\; F \;\; D \;\; E \;\; (B - S_i) \;\; w \;\; x$, where $G_i = \{g_{(i-1)(3m-3)+1}, \ldots, g_{i(3m-3)}\}$ for each i, $1 \leq i \leq n$.

2. For each $i, 1 \leq i \leq n$, there is one voter of the form: $B_i \;\; D_i \;\; w \;\; G \;\; E \;\; (D - D_i) \;\; F \;\; (B - B_i) \;\; c \;\; x$, where, letting $\ell_j = \|\{S_i \in \mathcal{S} \mid b_j \in S_i\}\|$ for each j, $1 \leq j \leq 3m$, we define $B_i = \{b_j \in B \mid i \leq n - \ell_j\}$ and $D_i = \{d_{(i-1)3m+1}, \ldots, d_{3im-\|B_i\|}\}$.

3. For each $k, 1 \leq k \leq m + 1$, there is one voter of the form: $x \;\; c \;\; E_k \;\; F \;\; (E - E_k) \;\; G \;\; D \;\; B \;\; w$, where $E_k = \{e_{(3m-1)(k-1)+1}, \ldots, e_{(3m-1)k}\}$ for each k, $1 \leq k \leq m + 1$.

4. For each l, $1 \leq l \leq m - 1$, there is one voter of the form: $F_l \;\; c \;\; (F - F_l) \;\; G \;\; D \;\; E \;\; B \;\; w \;\; x$, where $F_l = \{f_{(3m+1)(l-1)+1}, \ldots, f_{(3m+1)l}\}$, for each l, $1 \leq l \leq m - 1$.

In this election, candidate c is the unique level 2 BV winner with a level 2 score of $n + m + 1$.

We claim that $\mathcal{S}$ has an exact cover $\mathcal{S}'$ for B if and only if w can be made the unique BV winner of the resulting election by partition of voters (regardless of the tie-handling model used).

From left to right: Suppose $\mathcal{S}$ has an exact cover $\mathcal{S}'$ for B. Partition V the following way. Let V_1 consist of:

- the m voters of the first group that correspond to the exact cover (i.e., those m voters of the form $c \;\; S_i \;\; G_i \;\; (G - G_i) \;\; F \;\; D \;\; E \;\; (B - S_i) \;\; w \;\; x$ for which $S_i \in \mathcal{S}'$) and

- the $m + 1$ voters of the third group (i.e., all voters of the form $x \;\; c \;\; E_k \;\; F \;\; (E - E_k) \;\; G \;\; D \;\; B \;\; w$.

Let $V_2 = V - V_1$. In subelection (C, V_1), candidate x is the unique level 1 BV winner. In subelection (C, V_2), candidate w is the first candidate who has a strict majority and moves on to the final round of the election. Thus there are w and x in the final run-off, which w wins with a strict majority on the first level. Since both subelections, (C, V_1) and (C, V_2), have unique BV winners, candidate w can be made the unique BV winner by partition of voters, regardless of the tie-handling model used.

From right to left: Suppose that w can be made the unique BV winner by exerting control by partition of voters (for concreteness, say in TP). Let (V_1, V_2) be such a successful partition. Since w wins the resulting two-stage election, w has to win at least one of the subelections (say, w wins (C, V_1)). If candidate c participates in the final round, he or she wins the election with a strict majority no later than on the second level, no matter which other candidates move forward to the final election. That means that in both subelections, (C, V_1) and (C, V_2), c must not be a BV winner. Only in the second voter group candidate w (who has to be a BV winner in (C, V_1)) gets points earlier than on the second-to-last level. So w has to be a level $3m + 1$ BV winner in (C, V_1) via votes from the second voter group in V_1. As c scores already on the first two levels in voter groups 1 and 3, only x and the candidates in B can prevent c from winning in (C, V_2). However, since voters from the second voter group have to be in V_1 (as stated above), in subelection (C, V_2) only candidate x can prevent c from moving forward to the final round. Since x is always placed behind c in all votes except those votes from the third voter group, x has to be a level 1 BV winner in (C, V_2). In (C, V_1) candidate w gains all the points on exactly the $(3m + 1)$st level, whereas the other candidates scoring more than one point up to this level receive their points on either earlier or later levels, so no candidate can tie with w on the $(3m + 1)$st level and w

is the unique level $3m+1$ BV winner in (C, V_1). As both subelections, (C, V_1) and (C, V_2), have unique BV winners other than c, the construction works in model TE as well.

It remains to show that $\mathcal{S}$ has an exact cover $\mathcal{S}'$ for B. Since w has to win (C, V_1) with the votes from the second voter group, not all voters from the first voter group can be in V_1 (otherwise c would have n points already on the first level). On the other hand, there can be at most m voters from the first voter group in V_2 because otherwise x would not be a level 1 BV winner in (C, V_2). To ensure that no candidate in B has the same score as w, namely n points, and gets these points on an earlier level than w in (C, V_1), there have to be exactly m voters from the first group in V_2 and these voters correspond to an exact cover for B. ❑

Since Bucklin voting is a special case of fallback voting, we can answer one of the questions raised in [10] as follows:

COROLLARY 3.2. *Fallback voting is resistant to constructive control by partition of voters in model TP.*

The following construction will be used to handle the destructive case of control by partition of voters in model TP for fallback voting (see Theorem 3.5 below). The construction starts from an instance of RESTRICTED HITTING SET, a restricted version of the NP-complete problem HITTING SET (see, e.g., [13]), which is defined as follows:

Name: RESTRICTED HITTING SET (RHS).
Instance: A set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of nonempty subsets $S_i \subseteq B$ such that $n > m$, and a positive integer k with $1 < k < m$.
Question: Does $\mathcal{S}$ have a hitting set of size at most k, i.e., is there a set $B' \subseteq B$ with $\|B'\| \leq k$ such that for each i, $S_i \cap B' \neq \emptyset$?

Note that by dropping the requirement "$n > m > k > 1$," we obtain the (unrestricted) HITTING SET problem. It is easy to see that RESTRICTED HITTING SET is NP-complete.

CONSTRUCTION 3.3. *Let $(B, \mathcal{S}, k)$ be a given instance of* RHS*, with a set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of nonempty subsets $S_i \subseteq B$, and an integer k with $1 < k < m < n$. Define election (C, V), where $C = B \cup D \cup E \cup \{c, w\}$ is the candidate set with $D = \{d_1, \ldots, d_{2(m+1)}\}$ and $E = \{e_1, \ldots, e_{2(m-1)}\}$ and where V consists of the following $2n(k+1) + 4m + 2mk$ voters:*[1]

1. *For each i, $1 \leq i \leq n$, $k+1$ voters approve of w S_i c.*
2. *For each j, $1 \leq j \leq m$, one voter approves of c b_j w.*
3. *For each j, $1 \leq j \leq m$, $k-1$ voters approve of b_j.*
4. *For each p, $1 \leq p \leq m+1$, one voter approves of $d_{2(p-1)+1}$ d_{2p} w.*
5. *For each r, $1 \leq r \leq 2(m-1)$, one voter approves of e_r.*
6. *$n(k+1) + m - k + 1$ voters approve of c.*
7. *$mk + k - 1$ voters approve of c w.*
8. *One voter approves of w c.*

[1]Recall: Disapproved candidates are omitted and approved candidates are ranked in the votes of a fallback election.

Note that $maj(V) = n(k+1) + 2m + mk + 1$. In election (C, V), only the two candidates c and w reach a strict majority, w on the third level and c on the second level (see Table 2). Thus c is the unique level 2 FV winner of election (C, V). Lemma 3.4 will be used in the proof of Theorem 3.5.

	c	$d_p \in D$	$e_r \in E$
$score^1$	$n(k+1) + 2m + mk$	≤ 1	1
$score^2$	$n(k+1) + 2m + mk + 1$	1	1
$score^{m+2}$	$2n(k+1) + 2m + mk + 1$	1	1
	w	$b_j \in B$	
$score^1$	$n(k+1) + 1$	$k-1$	
$score^2$	$n(k+1) + mk + k$	$\leq k + n(k+1)$	
$score^{m+2}$	$n(k+1) + 2m + mk + k + 1$	$\leq k + n(k+1)$	

Table 2: Level i scores in (C, V) for $i \in \{1, 2, m+2\}$.

LEMMA 3.4. *In election (C, V) from Construction 3.3, for every partition of V into V_1 and V_2, candidate c is an FV winner of (C, V_1) or (C, V_2).*

PROOF. For a contradiction, suppose that in both subelections, (C, V_1) and (C, V_2), candidate c is not an FV winner. Since $score^1_{(C,V)}(c) = \|V\|/2$, the two subelections satisfy that both $\|V_1\|$ and $\|V_2\|$ are even numbers, and that $score^1_{(C,V_1)}(c) = \|V_1\|/2$ and $score^1_{(C,V_2)}(c) = \|V_2\|/2$. Otherwise, c would have a strict majority already on the first level in one of the subelections and would win that subelection. For each $i \in \{1, 2\}$, c already on the first level has only one point less than the strict majority threshold $maj(V_i)$ in subelection (C, V_i), and c will get a strict majority in (C, V_i) no later than on the $(m+2)$nd level. Thus, for both $i = 1$ and $i = 2$, there must be candidates whose level $m+2$ scores in (C, V_i) are higher than the level $m+2$ score of c in (C, V_i). Table 2 shows the level $m+2$ scores of all candidates in (C, V). Only w and some $b_j \in B$ have a chance to beat c on that level in (C, V_i), $i \in \{1, 2\}$.

Suppose that c is defeated in both subelections by two distinct candidates from B (say, b_x defeats c in (C, V_1) and b_y defeats c in (C, V_2)). Thus the following must hold:[2]

$$score^{m+2}_{(C,V_1)}(b_x) + score^{m+2}_{(C,V_2)}(b_y) \geq score^{m+2}_{(C,V)}(c) + 2$$
$$2n(k+1) + 2k - n(k+1) \geq 2n(k+1) + mk + 2m + 3$$
$$2k \geq n(k+1) + mk + 2m + 3,$$

which contradicts our basic assumption $m > k > 1$. Thus the only possibility for c to not win any of the two subelections is that c is defeated in one subelection, say (C, V_1), by a candidate from B, say b_x, and in the other subelection, (C, V_2), by candidate w. Then it must hold that:[2]

$$score^{m+2}_{(C,V_1)}(b_x) + score^{m+2}_{(C,V_2)}(w) \geq score^{m+2}_{(C,V)}(c) + 2,$$

which is equivalent to

$$2n(k+1) + 2k + 2m + mk + 1 - n(k+1) - 1$$
$$\geq \quad 2n(k+1) + mk + 2m + 3,$$

i.e., $2k \geq n(k+1) + 3$. Since $n > 1$, this cannot hold, so c must be an FV winner in one of the subelections. ❑

[2]For the left-hand sides of the inequalities, note that each vote occurs in only one of the two subelections. To avoid double-counting those votes that give points to both candidates, we first sum up the overall number of points each candidate scores and then substract the double-counted points.

THEOREM 3.5. *Fallback voting is resistant to destructive control by partition of voters in model TP.*

PROOF. Susceptibility holds by [10, Lemma 3.4]. To prove NP-hardness in the TP case, we reduce RHS to our control problem. Consider the election (C, V) constructed according to Construction 3.3 from a given RHS instance $(B, \mathcal{S}, k)$, where $B = \{b_1, \ldots, b_m\}$ is a set, $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a collection of nonempty subsets $S_i \subseteq B$, and k is an integer with $1 < k < m < n$.

We claim that $\mathcal{S}$ has a hitting set $B' \subseteq B$ of size k if and only if c can be prevented from being the unique FV winner by partition of voters in model TP.

From left to right: Suppose $B' \subseteq B$ is a hitting set of size k for $\mathcal{S}$. Partition V into V_1 and V_2 as follows. Let V_1 consist of those voters of the second group where $b_j \in B'$ and of those voters of the third group where $b_j \in B'$. Let $V_2 = V - V_1$. In (C, V_1), no candidate reaches a strict majority (see Table 3), where $maj(V_1) = \lfloor k^2/2 \rfloor + 1$, and candidates c, w, and each $b_j \in B'$ win the election with an approval score of k.

	c	w	$b_j \in B'$	$b_j \notin B'$
score^1	k	0	$k-1$	0
score^2	k	0	k	0
score^3	k	k	k	0

Table 3: Level i scores in (C, V_1) for $i \in \{1, 2, 3\}$ and all candidates in $B \cup \{c, w\}$.

	c	$b_j \in B'$
score^1	$n(k+1) + 2m - k + mk$	0
score^2	$n(k+1) + 2m - k + mk + 1$	$\leq n(k+1)$
score^3	$\geq n(k+1) + 2m - k + mk + 1$	$\leq n(k+1)$
	w	$b_j \notin B'$
score^1	$n(k+1) + 1$	$k-1$
score^2	$n(k+1) + mk + k$	$\leq k + n(k+1)$
score^3	$n(k+1) + mk + 2m + 1$	$\leq k + n(k+1)$

Table 4: Level i scores in (C, V_2) for $i \in \{1, 2, 3\}$ and all candidates in $B \cup \{c, w\}$.

The level i scores in election (C, V_2) for $i \in \{1, 2, 3\}$ and all candidates in $B \cup \{c, w\}$ are shown in Table 4. Since in (C, V_2) no candidate from B wins, the candidates participating in the final round are $B' \cup \{c, w\}$. The scores in the final election $(B' \cup \{c, w\}, V)$ can be seen in Table 5. Since candidates c and w with the same level 2 scores are both level 2 FV winners, candidate c has been prevented from being the unique FV winner by partition of voters in model TP.

	c	w
score^1	$n(k+1) + 2m + mk$	$n(k+1) + m + 2$
score^2	$n(k+1) + 2m + mk + 1$	$n(k+1) + 2m + mk + 1$
	$b_j \in B'$	
score^1	$k-1$	
score^2	$\leq k + n(k+1)$	

Table 5: Level i scores in the final-stage election $(B' \cup \{c, w\}, V)$ for $i \in \{1, 2\}$.

From right to left: Suppose candidate c can be prevented from being a unique FV winner by partition of voters in model TP. From Lemma 3.4 it follows that candidate c participates in the final round. Since c has a strict majority of approvals, c has to be tied with or lose against another candidate by a strict majority at some level. Only candidate w has a strict majority of approvals, so w has to tie or beat c at some level in the final round. Because of the low scores of the candidates in D and E we may assume that only candidates from B are participating in the final round besides c and w. Let $B' \subseteq B$ be the set of candidates who also participate in the final round. Let ℓ be the number of sets in $\mathcal{S}$ not hit by B'. As w cannot reach a strict majority of approvals on the first level, we consider the level 2 scores of c and w: $score^2_{(B' \cup \{c,w\}, V)}(c) = n(k+1) + 2m + mk + 1 + \ell(k+1)$, and $score^2_{(B' \cup \{c,w\}, V)}(w) = n(k+1) + 2m + mk + k - \|B'\| + 1$. Since c has a strict majority already on the second level, w must tie or beat c on this level, so the following must hold:

$$\begin{aligned} score^2_{(B' \cup \{c,w\}, V)}(c) - score^2_{(B' \cup \{c,w\}, V)}(w) &\leq 0 \\ \|B'\| - k + \ell(k+1) &\leq 0. \end{aligned}$$

This is possible only if $\ell = 0$ (i.e., all sets in $\mathcal{S}$ are hit by B'), so $\|B'\| \leq k$. Thus $\mathcal{S}$ has a hitting set of size at most k. ❑

Finally, we turn to destructive control by partition of voters in model TE. The proof of Theorem 3.6 (which employs a reduction from DOMINATING SET) is omitted due to space.

THEOREM 3.6. *Bucklin voting (and thus fallback voting as well) is resistant to destructive control by partition of voters in model TE.*

4. CANDIDATE CONTROL IN BV

Theorem 4.1 strengthens the corresponding result for fallback voting [10].

THEOREM 4.1. *Bucklin voting is resistant to each of the* 14 *standard types of candidate control.*

For the hardness proofs showing Theorem 4.1, we again use the RHS problem defined in Section 3.

In this section, all reductions except one (namely that used to prove Lemma 4.2) will apply Construction 4.3 below. We first handle this one exception.

LEMMA 4.2. *Bucklin voting is resistant to constructive control by deleting candidates.*

PROOF. Susceptibility holds by Lemma 2.1. To prove NP-hardness of our control problem, we give a reduction from RHS. Let $(B, \mathcal{S}, k)$ be a RHS instance with a set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of nonempty subsets $S_i \subseteq B$, and a positive integer k satisfying $k < m < n$. Let $s_i = n + k - \|S_i\|$, $1 \leq i \leq n$, and $s = \sum_{i=1}^{n} s_i$. Note that all s_i are positive, since $m < n$.

Define election (C, V) with candidate set

$$C = B \cup C' \cup D \cup E \cup F \cup \{w\},$$

where $C' = \{c_1, c_2, \ldots, c_{k+1}\}, D = \{d_1, d_2, \ldots, d_s\}, E = \{e_1, e_2, \ldots, e_n\}, F = \{f_1, \ldots, f_{n+k}\}$, and let w be the distinguished candidate. Note that the number of candidates in D is $s = n^2 + kn - \sum_{i=1}^{n} \|S_i\|$. For each i, $1 \leq i \leq n$, let $D_i = \{d_{1+\sum_{j=1}^{i-1} s_j}, \ldots, d_{\sum_{j=1}^{i} s_j}\}$, so $\|D_i\| = s_i$.

Define V to consist of the following $2(n+k+1)+1$ voters:

1. For each i, $1 \leq i \leq n$, there is one voter of the form: $S_i \;\; D_i \;\; w \;\; C' \;\; E \;\; (D - D_i) \;\; (B - S_i) \;\; F$.

2. For each j, $1 \leq j \leq k+1$, there is one voter of the form: $E \;\; (C' - \{c_j\}) \;\; c_j \;\; B \;\; D \;\; w \;\; F$.

3. There are $k+1$ voters of the form: $w \;\; F \;\; C' \;\; E \;\; B \;\; D$.

4. There are n voters of the form: $C' \;\; D \;\; F \;\; B \;\; w \;\; E$.

5. There is one voter of the form: $C' \;\; w \;\; D \;\; F \;\; E \;\; B$.

There is no unique BV winner in election (C, V), since w and the candidates in C' are level $n+k+1$ BV winners.

We claim that $\mathcal{S}$ has a hitting set of size k if and only if w can be made the unique BV winner by deleting at most k candidates.

From left to right: Suppose $\mathcal{S}$ has a hitting set B' of size k. Delete the corresponding candidates. Now, w is the unique level $n+k$ BV winner of the resulting election.

From right to left: Suppose w can be made the unique BV winner by deleting at most k candidates. Since $k+1$ candidates other than w have a strict majority on level $n+k+1$ in election (C, V), after deleting at most k candidates, there is still at least one candidate other than w with a strict majority of approvals on level $n+k+1$. However, since w was made the unique BV winner by deleting at most k candidates, w must be the unique BV winner on a level lower than or equal to $n+k$. This is possible only if in all n votes of the first voter group w moves forward by at least one position. This, however, is possible only if $\mathcal{S}$ has a hitting set B' of size k. ❑

Construction 4.3 will be applied to prove the remaining 13 cases of candidate control stated in Theorem 4.1.

CONSTRUCTION 4.3. *Let $(B, \mathcal{S}, k)$ be a given instance of* RHS*, where $B = \{b_1, b_2, \ldots, b_m\}$ is a set, $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a collection of nonempty subsets $S_i \subseteq B$ such that $n > m$, and $k < m$ is a positive integer. (Thus, $n > m > k > 1$.) Define election (C, V), where $C = B \cup \{c, d, w\}$ is the candidate set and where V consists of the following $6n(k+1) + 4m + 11$ voters:*

1. *$2m+1$ voters: $c\, d\, B\, w$.*

2. *$2n + 2k(n-1) + 3$ voters: $c\, w\, d\, B$.*

3. *$2n(k+1) + 5$ voters: $w\, c\, d\, B$.*

4. *For each i, $1 \leq i \leq n$, $2(k+1)$ voters: $d\, S_i\, c\, w\, (B - S_i)$.*

5. *For each j, $1 \leq j \leq m$, two voters: $d\, b_j\, w\, c\, (B - \{b_j\})$.*

6. *$2(k+1)$ voters: $d\, w\, c\, B$.*

We now prove Theorem 4.1 (except for the case already handled separately in Lemma 4.2) via Construction 4.3, making use of the following lemma.

LEMMA 4.4. *Consider the election (C, V) constructed according to Construction 4.3 from a* RHS *instance $(B, \mathcal{S}, k)$.*

1. *c is the unique level 2 BV winner of $(\{c, d, w\}, V)$.*

2. *If $\mathcal{S}$ has a hitting set B' of size k, then w is the unique BV winner of election $(B' \cup \{c, d, w\}, V)$.*

3. *Let $D \subseteq B \cup \{d, w\}$. If c is not a unique BV winner of election $(D \cup \{c\}, V)$, then there exists a set $B' \subseteq B$ such that*

   (a) $D = B' \cup \{d, w\}$,

   (b) w is a level 2 BV winner of $(B' \cup \{c, d, w\}, V)$,

   (c) B' is a hitting set for $\mathcal{S}$ of size at most k.

PROOF. For the first part, note that there is no level 1 BV winner in election $(\{c, d, w\}, V)$ and we have the following level 2 scores in this election:

$$\begin{aligned} score^2_{(\{c,d,w\},V)}(c) &= 6n(k+1) + 2(m-k) + 9, \\ score^2_{(\{c,d,w\},V)}(d) &= 2n(k+1) + 4m + 2k + 3, \\ score^2_{(\{c,d,w\},V)}(w) &= 4n(k+1) + 2m + 10. \end{aligned}$$

Since $n > m$ (which implies $n > k$), we have:

$$\begin{aligned} &score^2_{(\{c,d,w\},V)}(c) - score^2_{(\{c,d,w\},V)}(d) \\ &\quad = 4n(k+1) - (2m + 4k) + 6 > 0, \\ &score^2_{(\{c,d,w\},V)}(c) - score^2_{(\{c,d,w\},V)}(w) \\ &\quad = 2n(k+1) - (2k+1) > 0. \end{aligned}$$

Thus, c is the unique level 2 BV winner of $(\{c, d, w\}, V)$.

For the second part, suppose that B' is a hitting set for $\mathcal{S}$ of size k. Then there is no level 1 BV winner in election $(B' \cup \{c, d, w\}, V)$, and we have the following level 2 scores:

$$\begin{aligned} score^2_{(B'\cup\{c,d,w\},V)}(c) &= 4n(k+1) + 2(m-k) + 9, \\ score^2_{(B'\cup\{c,d,w\},V)}(d) &= 2n(k+1) + 4m + 2k + 3, \\ score^2_{(B'\cup\{c,d,w\},V)}(w) &= 4n(k+1) + 2(m-k) + 10, \\ score^2_{(B'\cup\{c,d,w\},V)}(b_j) &\leq 2n(k+1) + 2 \quad \text{for all } b_j \in B'. \end{aligned}$$

It follows that w is the unique level 2 BV winner of election $(B' \cup \{c, d, w\}, V)$.

For the third part, let $D \subseteq B \cup \{d, w\}$. Suppose c is not a unique BV winner of election $(D \cup \{c\}, V)$.

(3a) Other than c, only w has a strict majority of votes on the second level and only w can tie or beat c in $(D \cup \{c\}, V)$. Thus, since c is not a unique BV winner of election $(D \cup \{c\}, V)$, w is clearly in D. In $(D \cup \{c\}, V)$, candidate w has no level 1 strict majority, and candidate c has already on level 2 a strict majority. Thus, w must tie or beat c on level 2. For a contradiction, suppose $d \notin D$. Then

$$\begin{aligned} score^2_{(D\cup\{c\},V)}(c) &\geq 4n(k+1) + 2m + 11; \\ score^2_{(D\cup\{c\},V)}(w) &= 4n(k+1) + 2m + 10, \end{aligned}$$

which contradicts the observation that w ties or beats c on level 2. Thus, $D = B' \cup \{d, w\}$, where $B' \subseteq B$.

(3b) This part follows immediately from the proof of (3a).

(3c) Let ℓ be the number of sets in $\mathcal{S}$ not hit by B'. We have that $score^2_{(B'\cup\{c,d,w\},V)}(w) = 4n(k+1) + 10 + 2(m - \|B'\|)$ and $score^2_{(B'\cup\{c,d,w\},V)}(c) = 2(m-k) + 4n(k+1) + 9 + 2(k+1)\ell$. From part (3b) we know that

$$score^2_{(B'\cup\{c,d,w\},V)}(w) \geq score^2_{(B'\cup\{c,d,w\},V)}(c),$$

so $4n(k+1) + 10 + 2(m - \|B'\|) \geq 2(m-k) + 4n(k+1) + 9 + 2(k+1)\ell$. This inequality implies $1 > \frac{1}{2} \geq \|B'\| - k + (k+1)\ell$. Since $T = \|B'\| - k + (k+1)\ell$ is an integer, we have $T \leq 0$. If $T = 0$ then $\ell = 0$ and $\|B'\| = k$. Now assume $T < 0$. If $\ell = 0$, B' is a hitting set with $\|B'\| < k$, and if $\ell > 0$ then $(k+1)\ell > k$, which contradicts $T = \|B'\| - k + (k+1)\ell < 0$. In each possible case, we have a hitting set (as $\ell = 0$) of size at most k. ❑

Proof of Theorem 4.1. In each case, susceptibility holds by Lemma 2.1. For the four adding-candidates cases, NP-hardness follows immediately from Lemma 4.4.

NP-hardness for constructive control by deleting candidates has been shown in Lemma 4.2. To show the problem NP-hard in the destructive case, let (C, V) be the election resulting from a RHS instance $(B, \mathcal{S}, k)$ according to Construction 4.3, and let c be the distinguished candidate. We claim that $\mathcal{S}$ has a hitting set of size at most k if and only if c can be prevented from being a unique BV winner by deleting at most $m - k$ candidates.

From left to right: Suppose $\mathcal{S}$ has a hitting set B' of size k. Delete the $m - k$ candidates $B - B'$. Now, both candidates c and w have a strict majority on level 2, but

$$\begin{aligned} score^2_{(\{c,d,w\}\cup B',V)}(c) &= 4n(k+1) + 2(m-k) + 9, \\ score^2_{(\{c,d,w\}\cup B',V)}(w) &= 4n(k+1) + 2(m-k) + 10, \end{aligned}$$

so w is the unique level 2 BV winner of this election.

From right to left: Suppose that c can be prevented from being a unique BV winner by deleting at most $m - k$ candidates. Let $D' \subseteq B \cup \{d, w\}$ be the set of deleted candidates (so $c \notin D'$) and $D = (C - D') - \{c\}$. It follows immediately from Lemma 4.4 that $D = B' \cup \{d, w\}$, where B' is a hitting set for $\mathcal{S}$ of size at most k.

To show that Bucklin voting is resistant to constructive (or destructive) control by partition/run-off partition of candidates in TE and TP, map the instance $(B, \mathcal{S}, k)$ to the instance $((C, V), w)$ (or $((C, V), c)$), where (C, V) is the election from Construction 4.3. NP-hardness now follows from Lemma 4.4; the detailed argument is omitted due to space limitations (note that, in particular, if $\mathcal{S}$ has a hitting set of size k, partitioning $C = (C_1, C_2)$ into $C_1 = B' \cup \{c, d, w\}$ and $C_2 = C - C_1$ will be successful). ❑ Theorem 4.1

5. ADDING/DELETING VOTERS IN BV

Finally, we turn to control by adding voters and by deleting voters for Bucklin voting. As with fallback voting [10], we have resistance in the constructive cases and vulnerability in the destructive cases. Since Bucklin voting is a special case of fallback voting, the two resistance results in Theorem 5.1 (which both are shown via a reduction from X3C) strengthen the corresponding results for fallback voting [10] and the two vulnerability results immediately follow from the corresponding results for fallback voting [10]. The proof of Theorem 5.1 is omitted due to space limitations.

THEOREM 5.1. *Bucklin voting is resistant to constructive control by adding voters and by deleting voters and is vulnerable to destructive control by adding voters and by deleting voters.*

6. CONCLUSIONS

Solving the three open questions of Erdélyi and Rothe [10], we have shown that fallback voting is fully resistant to control by partition of voters. Thus, among natural voting systems with a polynomial-time winner problem, fallback voting has the most proven resistances to control. SP-AV is known to have an almost as broad control resistance [9]; however, fallback voting is arguably more natural than SP-AV. We have also studied the control complexity of Bucklin voting, thus improving the corresponding resistance results for fallback voting. One case of control by partition of voters (namely, the destructive case in model TP) remains open for Bucklin voting. It would also be interesting and challenging to complement our worst-case hardness results by theoretical and empirical typical-case studies of these problems.

Acknowledgment

We thank the reviewers for the very helpful comments. This work was supported in part by DFG grants RO 1202/{11-1, 12-1} and the ESF EUROCORES program LogICCC.

7. REFERENCES

[1] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[2] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.

[3] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editors, *Handbook on Approval Voting*, chapter 10, pages 199–251. Springer, 2010.

[4] S. Brams and R. Sanver. Critical strategies under approval voting: Who gets ruled in and ruled out. *Electoral Studies*, 25(2):287–305, 2006.

[5] S. Brams and R. Sanver. Voting systems that combine approval and preference. In S. Brams, W. Gehrlein, and F. Roberts, editors, *The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn*, pages 215–237. Springer, 2009.

[6] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[7] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. WWW'01*, pages 613–622. ACM Press, 2001.

[8] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[9] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.

[10] G. Erdélyi and J. Rothe. Control complexity in fallback voting. In *Proc. CATS'09*, pages 39–48. Australian Computer Society *Conf. in Research and Practice in IT Series*, vol. 32, no. 8, January 2010.

[11] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[12] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[13] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[14] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.

[15] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[16] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

An Algorithm for the Coalitional Manipulation Problem under Maximin

Michael Zuckerman
michez@cs.huji.ac.il

Omer Lev
omerl@cs.huji.ac.il

Jeffrey S. Rosenschein
jeff@cs.huji.ac.il

The School of Computer Science and Engineering
The Hebrew University of Jerusalem

ABSTRACT

We introduce a new algorithm for the Unweighted Coalitional Manipulation problem under the Maximin voting rule. We prove that the algorithm gives an approximation ratio of $1\frac{2}{3}$ to the corresponding optimization problem. This is an improvement over the previously known algorithm that gave a 2-approximation. We also prove that its approximation ratio is no better than $1\frac{1}{2}$, i.e., there are instances on which a $1\frac{1}{2}$-approximation is the best the algorithm can achieve. Finally, we prove that no algorithm can approximate the problem better than to the factor of $1\frac{1}{2}$, unless $\mathcal{P} = \mathcal{NP}$.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms

Keywords

Social choice theory, Algorithms, Approximation

1. INTRODUCTION

In recent years, the importance of game-theoretic analysis as a formal foundation for multiagent systems has been widely recognized in the agent research community. As part of this research agenda, the field of *computational social choice* has arisen to explore ways in which multiple agents can effectively (and tractably) use elections to combine their individual, self-interested preferences into an overall choice for the group.

In an election, voters (agents) submit linear orders (rankings, or profiles) of the candidates (alternatives); a *voting rule* is then applied to the rankings in order to choose the winning candidate. In the prominent impossibility result proven by Gibbard and Satterthwaite [8, 11], it was shown that for any voting rule, a) which is not a dictatorship, b) which is onto the set of alternatives, and c) where there are at least three alternatives, there exist profiles where a voter can benefit by voting insincerely. Submitting insincere rankings in an attempt to benefit is called *manipulation*. Exploring the computational complexity of, and algorithms for, this *manipulation problem* is one of the most important research areas in computational social choice.

There are several ways to circumvent the Gibbard-Satterthwaite result, one of which is by using computational complexity as a barrier against manipulation. The idea behind this technique is as follows: although there may exist a successful manipulation, the voter must *discover* it before it can be used—but for certain voting rules, discovering a successful manipulation might be computationally hard. This argument was used already in 1989 by Bartholdi et al. [2], and in 1991 by Bartholdi and Orlin [1], where they proved, respectively, that second-order Copeland and Single Transferable Vote are both $\mathcal{NP}$-hard to manipulate.

Later, the complexity of coalitional manipulation was studied by Conitzer et al. [3]. In the coalitional manipulation problem, a coalition of potentially untruthful voters try to coordinate their ballots so as to make some preferred candidate win the election. Conitzer et al. studied the problem where manipulators are weighted: a voter with weight l counts as l voters, each of weight 1. This problem was shown to be $\mathcal{NP}$-hard, for many voting rules, even for a constant number of candidates. However, it has been argued that a more natural setting is the unweighted coalitional manipulation (UCM) problem, where all voters have equal power. In a recent paper [13], Xia et al. established as one of their main results that UCM is $\mathcal{NP}$-hard under the Maximin voting rule, even for 2 untruthful voters.

In 2009, Zuckerman et al. [14] defined a natural optimization problem for the unweighted setting (i.e., Unweighted Coalitional Optimization, UCO), namely finding the minimal number of manipulators sufficient to make some predefined candidate win. It is proven, as a corollary of their results, that the heuristic greedy algorithm proposed in the paper gives a 2-approximation to the UCO problem under Maximin. Here, we further study the UCO problem under Maximin, proposing a new greedy algorithm that gives a $1\frac{2}{3}$-approximation to the problem.[1] We then provide an example showing that the approximation ratio of the algorithm is no better than $1\frac{1}{2}$. Furthermore, since this gap (between $1\frac{2}{3}$ and $1\frac{1}{2}$) is due to the fact that the size of the manipulating coalition is rounded upwards, the actual bound on the ratio between the size of the coalition returned by the algorithm, and the minimum size of manipulating coalition, tends to $1\frac{1}{2}$ as the number of voters tends to infinity.

2. RELATED WORK

Behavior designed to alter outcomes in the Maximin voting rule has been widely studied. Perhaps the closest work to the UCM

[1]Strictly speaking, our algorithm is for the *decision* problem, but since the conversion of our algorithm to one for the optimization problem is straightforward, we consider it an approximation algorithm for the optimization problem.

Cite as: An Algorithm for the Coalitional Manipulation Problem under Maximin, Michael Zuckerman, Omer Lev and Jeffrey S. Rosenschein, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 845-852.

problem is control by adding voters (AV), which has been studied by Faliszewski et al. [6]. The difference between AV control and UCM is that in the latter, manipulative voters can vote whatever they like in order to make their preferred candidate win, whereas in the former, the votes in the additional set are fixed. Faliszewski et al. proved that AV control in Maximin (as well as DV [Delete Voters] control and constructive AC [Add Candidates] control) is $\mathcal{NP}$-complete. In contrast, they showed polynomial-time algorithms for a combination of AC_u (a variant of Adding Candidates) and DC (Delete Candidates), and for a combination of destructive AC and DC control.

In another paper, Elkind et al. studied control of elections by cloning candidates [5]. For prominent voting rules (including Maximin) they characterized preference profiles for which there exist a successful cloning manipulation. For Maximin, a profile is manipulable by cloning if and only if the preferred candidate does not win, but is Pareto optimal. The authors also provided a simple linear-time algorithm for solving the cloning manipulation problem under Maximin.

Yet another topic that involves outcome-altering behavior in elections is bribery. In their paper [4], Elkind et al. investigated a model of bribery where the price of each vote depends on the amount of change that the voter is asked to implement. They showed that for their model, bribery is $\mathcal{NP}$-complete for Maximin, as well as for some other voting rules.

3. MAXIMIN VOTING, MANIPULATION

An election consists of a set $C = \{c_1, \ldots, c_m\}$ of candidates, and a set $S = \{v_1, \ldots, v_{|S|}\}$ of voters. Each voter provides a total order on the candidates (i.e., each voter submits a linear ranking of all the candidates). The setting also includes a *voting rule*, which is a function from the set of all possible combinations of votes to C.

The Maximin voting rule is defined as follows. For any two distinct candidates x and y, let $N(x, y)$ be the number of voters who prefer x over y. The *Maximin score* of x is $S(x) = \min_{y \neq x} N(x, y)$. The candidate with the highest Maximin score is the winner.

DEFINITION 3.1. *In the* CONSTRUCTIVE COALITIONAL UNWEIGHTED MANIPULATION (CCUM) *problem, we are given a set C of candidates, with a distinguished candidate $p \in C$, a set of (unweighted) voters S that have already cast their votes (these are the non-manipulators), and a set T of (unweighted) voters that have not yet cast their votes (these are the manipulators). We are asked whether there is a way to cast the votes in T so that p wins the election.*

DEFINITION 3.2. *In the* UNWEIGHTED COALITIONAL OPTIMIZATION (UCO) *problem we are given a set C of candidates, with a distinguished candidate $p \in C$, and a set of (unweighted) voters S that have already cast their votes (the non-manipulators). We are asked for the* minimal *n such that a set T of size n of (unweighted) manipulators can cast their votes in order to make p win the election.*

REMARK 3.3. *We implicitly assume here that the manipulators have full knowledge about the non-manipulators' votes (this is the common assumption in the literature). Unless explicitly stated otherwise, we also assume that ties are broken adversarially to the manipulators, so that if p ties with another candidate, p loses. The latter assumption is equivalent to formulating the manipulation problems in their* unique winner *version, when one assumes that all candidates with maximal score win, but asks that p be the only winner.*

Throughout this paper we will use the convention, unless explicitly stated otherwise, that $|C| = m$, $|S| = N$ and $|T| = n$. We will denote $N_i(x, y) = |\{j \mid x \succ_j y, \succ_j \in S \cup \{1, \ldots, i\}\}|$. That is, $N_i(x, y)$ will denote the number of voters from S and from the first i voters of T that prefer x over y (assuming S is fixed, and fixing some order on the voters of T). Furthermore, we will denote by $S_i(c)$ the accumulated score of candidate c from the voters of S and the first i voters of T. By definition, for each $x \in C$, $S_i(x) = \min_{y \neq x} N_i(x, y)$. Also, we denote for $x \in C$, $\text{MIN}_i(x) = \{y \in C \setminus \{x\} \mid S_i(x) = N_i(x, y)\}$. We denote for $0 \leq i \leq n$, $ms(i) = \max_{c \in C \setminus \{p\}} S_i(c)$. That is, $ms(i)$ is the maximum score of the opponents of p after i manipulators have voted.

DEFINITION 3.4. *The* Condorcet winner *of an election is the candidate who, when compared with every other candidate, is preferred by more voters.*

Next we give a lower bound on the approximation ratio of any polynomial-time algorithm for the UCO problem under Maximin.

PROPOSITION 3.5. *No polynomial-time algorithm approximating the UCO problem under Maximin can do better than $1\frac{1}{2}$, unless $\mathcal{P} = \mathcal{NP}$.*

PROOF. Suppose, for contradiction, that there exists a polynomial-time approximation algorithm $\mathcal{A}$ to the UCO problem under Maximin having approximation ratio $r < 1\frac{1}{2}$. Then when $opt = 2$, the minimal size of manipulating coalition returned by $\mathcal{A}$ is $n \leq r \cdot opt < 3$. Since the size of the coalition is an integer, it follows that $n = 2$. Therefore, $\mathcal{A}$ can decide the CCUM problem for the coalition of 2 manipulators, which contradicts the fact that this problem is $\mathcal{NP}$-complete [13] (unless $\mathcal{P} = \mathcal{NP}$). □

4. THE ALGORITHM

Our algorithm for the CCUM problem under the Maximin voting rule is given as Algorithm 1 (see the final page of the paper). The intuition behind Algorithm 1 is as follows. The algorithm tries in a greedy manner to maximize the score of p, and to minimize the scores of p's opponents. To achieve this, for all i, manipulator i puts p first in his preference list, making the score of p grow by 1. He then builds a digraph $G^{i-1} = (V, E^{i-1})$, where $V = C \setminus \{p\}$, $(x, y) \in E^{i-1}$ iff ($y \in \text{MIN}_{i-1}(x)$ and $p \notin \text{MIN}_{i-1}(x)$). He tries first to rank candidates without any outgoing edges from them, since their score will not grow this way (because their score is achieved vs. candidates who were already ranked above them). When there are no candidates without outgoing edges, the algorithm tries to find a cycle with two adjacent vertices having the lowest score. If it finds such a cycle, then it picks the front vertex of these two. Otherwise, any candidate with the lowest score is chosen. After ranking each candidate, the edges in the graph are updated, so that all candidates whose minimal candidate has already been ranked will be with outgoing degree 0. For an edge (x, y), if y has already been ranked, we remove all the edges going out of x, since if we rank x now, its score will not go up, and so it does not depend on other candidates in $\text{MIN}_{i-1}(x)$. There is no need of an edge (x, y) if $p \in \text{MIN}_{i-1}(x)$, since for all $x \in C \setminus \{p\}$, p is always ranked above x, and so whether y is ranked above x or not, the score of x will not grow.

Let us note a few points regarding the algorithm:

- When picking a candidate with an out-degree 0, the algorithm first chooses candidates with the lowest score (among the candidates with an out-degree 0). It appears that this issue is critical for getting the approximation ratio of $1\frac{2}{3}$.

- The candidates with out-degree 0 are kept in stacks in order to guarantee a DFS-like order among candidates with the same score (this is needed for Lemma 6.4, below, to work).
- After a candidate b is added to the manipulator's preference list, for each candidate y who has an outgoing edge (y, b), the algorithm removes all the outgoing edges of y, puts it into the appropriate stack, and assigns b to be y's "father". Essentially, the assignment $y.father \leftarrow b$ means that due to b the score of y did not grow. The "father" relation is used to analyze the algorithm.
- Note the subtle difference between calculating the scores in Algorithm 1 in this paper, as compared to Algorithm 1 in [14]. In the latter, the manipulator i calculates what the score would be of the current candidate x if he put x at the current place in his preference list; in the algorithm we are now presenting, manipulator i just calculates $S_{i-1}(x)$. This difference is due to the fact that here, when we calculate the score of x, we know whether $d_{out}(x) > 0$, i.e., we know whether the score of x will grow by 1 if we put it at the current available place. So we separately compare the scores of candidates with out-degree > 0, and the scores of candidates with out-degree 0.

DEFINITION 4.1. *We refer to an iteration of the main* **for** *loop in lines 3–37 of Algorithm 1 as a* stage *of the algorithm. That is, a stage of the algorithm is a vote of any manipulator.*

DEFINITION 4.2. *In the digraph G^i built by the algorithm, if there exists an edge (x, y), we refer to $N_i(x, y) = S_i(x)$ as the* weight *of the edge (x, y).*

5. 2-APPROXIMATION

We first prove that Algorithm 1 has an approximation ratio of 2. We then use this result in the subsequent proof of the $1\frac{2}{3}$ approximation ratio.

THEOREM 5.1. *Algorithm 1 has a 2-approximation ratio for the UCO problem under the Maximin voting rule.*

To prove the above theorem, we first need the following two lemmas. In the first lemma, we prove that a certain sub-graph of the graph built by the algorithm contains a cycle passing through some distinguished vertex. We first introduce some more notation.

Let $G^i = (V, E^i)$ be the directed graph built by Algorithm 1 in stage $i + 1$. For a candidate $x \in C \setminus \{p\}$, let $G^i_x = (V^i_x, E^i_x)$ be the graph G^i reduced to the vertices that were ranked below x in stage $i + 1$, including x.

Let $V^i(x) = \{y \in V^i_x \mid \text{there is a path in } G^i_x \text{ from } x \text{ to } y\}$. Also, let $G^i(x)$ be the sub-graph of G^i_x induced by $V^i(x)$.

LEMMA 5.2. *Let i be an integer, $0 \leq i \leq n - 1$. Let $x \in C \setminus \{p\}$ be a candidate. Denote $t = ms(i)$. Suppose that $S_{i+1}(x) = t + 1$. Then $G^i(x)$ contains a cycle passing through x.*

PROOF. First of all note that for all $c \in V^i(x)$, $S_i(c) = t$. It follows from the fact that by definition $S_i(c) \leq t$. On the other hand, $S_i(x) = t$, and all the other vertices in $V^i(x)$ were ranked below x. Together with the fact that the out-degree of x was greater than 0 when x was picked, it gives us that for all $c \in V^i(x)$, $S_i(c) \geq t$, and so for all $c \in V^i(x)$, $S_i(c) = t$. We claim that for all $c \in V^i(x)$, $\text{MIN}_i(c) \subseteq V^i(x)$. If, by way of contradiction, there exists $c \in V^i(x)$ s.t. there is $b \in \text{MIN}_i(c)$ where $b \notin V^i(x)$, then $b \notin V^i_x$, since otherwise, if $b \in V^i_x$, then from $c \in V^i(x)$ and $(c, b) \in E^i_x$ we get that $b \in V^i(x)$. So $b \notin V^i_x$, which means that b was ranked by $i + 1$ above x. After we ranked b we removed all the outgoing edges from c, and so we chose c before x since $d_{out}(c) = 0$ and $d_{out}(x) > 0$ (since the score of x increased in stage $i + 1$). This contradicts the fact that $c \in V^i(x) \subseteq V^i_x$. Therefore, for every vertex $c \in V^i(x)$ there is at least one edge in $G^i(x)$ going out from c. Hence, there is at least one cycle in $G^i(x)$. Since at the time of picking x by voter $i + 1$, for all $c \in V^i(x)$, $d_{out}(c) > 0$, and by the observation that for all $c \in V^i(x)$, $S_i(c) = t$, we have that the algorithm picked the vertex x from a cycle (lines 21–22 of the pseudocode). □

In the following lemma, we show an upper bound on the growth rate of the scores of p's opponents.

LEMMA 5.3. *For all $0 \leq i \leq n - 2$, $ms(i + 2) \leq ms(i) + 1$.*

PROOF. Let $0 \leq i \leq n - 2$. Let $x \in C \setminus \{p\}$ be a candidate. Denote $t = ms(i)$. By definition, $S_i(x) \leq t$. We would like to show that $S_{i+2}(x) \leq t + 1$. If $S_{i+1}(x) \leq t$, then $S_{i+2}(x) \leq S_{i+1}(x) + 1 \leq t + 1$, and we are done. So let us assume now that $S_{i+1}(x) = t + 1$.

Let $V^i(x)$ and $G^i(x)$ be as before. By Lemma 5.2, $G^i(x)$ contains at least one cycle. Let U be one such cycle. Let $a \in U$ be the vertex that was ranked highest among the vertices of U in stage $i + 1$. Let b be the vertex before a in the cycle: $(b, a) \in U$. Since b was ranked below a in stage $i + 1$, it follows that $S_{i+1}(b) = S_i(b) \leq t$.

Suppose, for contradiction, that $S_{i+2}(x) > t+1$. Then the score of x increased in stage $i + 2$, and so when x was picked by $i + 2$, its out-degree in the graph was not 0. x was ranked by $i + 2$ at place s^*. Then b was ranked by $i + 2$ above s^*, since otherwise, when we had reached the place s^*, we would not pick x since b would be available (with out-degree 0, or otherwise—with score $S_{i+1}(b) \leq t < t + 1 = S_{i+1}(x)$)—a contradiction.

Denote by Z_1 all the vertices in $V^i(x)$ that have an outgoing edge to b in $G^i(x)$. For all $z \in Z_1$, $b \in \text{MIN}_i(z)$, i.e., $S_i(z) = N_i(z, b)$. We claim that all $z \in Z_1$ were ranked by $i + 2$ above x. If, by way of contradiction, there is $z \in Z_1$, s.t. until the place s^* it still was not added to the preference list, then two cases are possible:

1. If $(z, b) \in E^{i+1}$, then after b was added to $i+2$'s preference list, we removed all the outgoing edges of z, and we would put in z (with out-degree 0) instead of x, a contradiction.
2. $(z, b) \notin E^{i+1}$. Since $(z, b) \in E^i$, we have $S_i(z) = N_i(z, b)$. Also since z was ranked by $i + 1$ below x, it follows that $S_i(z) = t$. So from $(z, b) \notin E^{i+1}$, we have that $S_{i+1}(z) = t$ and $N_{i+1}(z, b) = t + 1$. Therefore, when reaching the place s^* in the $i + 2$'s preference list, whether $d_{out}(z) = 0$ or not, we would not pick x (with the score $S_{i+1}(x) = t+1$) since z (with the score $S_{i+1}(z) = t$) would be available, a contradiction.

Denote by Z_2 all the vertices in $V^i(x)$ that have an outgoing edge in $G^i(x)$ to some vertex $z \in Z_1$. In the same manner we can show that all the vertices in Z_2 were ranked in stage $i + 2$ above x. We continue in this manner, by defining sets $Z_3, \ldots$, where the set Z_l contains all vertices in $V^i(x)$ that have an outgoing edge to some vertex in Z_{l-1}; the argument above shows that all elements of these sets are ranked above x in stage $i + 2$. As there is a path from x to b in $G^i(x)$, we will eventually reach x in this way, i.e., there is some l such that Z_l contains a vertex y, s.t. $(x, y) \in E^i(x)$.

Now, if $(x, y) \in E^{i+1}(x)$, then since y was ranked by $i + 2$ above x, we have $S_{i+2}(x) = S_{i+1}(x) = t + 1$, a contradiction.

And if $(x,y) \notin E^{i+1}(x)$, then since $(x,y) \in E^i(x)$ we get that $N_{i+1}(x,y) = t+1$ and $S_{i+1}(x) = t$, a contradiction. □

We are now ready to prove Theorem 5.1.

PROOF OF THEOREM 5.1. Let *opt* denote the minimum size of coalition needed to make p win. It is easy to see that $opt \geq ms(0) - S_0(p) + 1$. We set $n = 2ms(0) - 2S_0(p) + 2 \leq 2opt$. Then, by Lemma 5.3:

$$ms(n) \leq ms(0) + \left\lceil \frac{n}{2} \right\rceil = 2ms(0) - S_0(p) + 1.$$

Whereas:

$$S_n(p) = S_0(p) + n = 2ms(0) - S_0(p) + 2 > ms(n).$$

So p will win when the coalition of manipulators is of size n. □

6. $1\frac{2}{3}$-APPROXIMATION

Our next goal is to prove that Algorithm 1 has an approximation ratio of $1\frac{2}{3}$ when there are no 2-cycles in the graphs built by the algorithm.

THEOREM 6.1. *For instances where there are no 2-cycles in the graphs G^i built by Algorithm 1, it gives a $1\frac{2}{3}$-approximation of the optimum.*

Let us give a general short overview of the proof of the above theorem (we will give an intuitive description rather than a formal/rigorous one). In Lemmas 6.2–6.5 we aim to prove that the maximum score of p's opponents grows 3 times slower than the score of p, at the most. After proving this, the theorem will easily follow. Recall that we proved in Lemma 5.2 that there is a cycle passing through x after i stages. Then we prove that at least one such cycle stays after stage $i+1$ (Lemma 6.2). In this cycle there are 2 consecutive vertices with a low score ($= t$) (Lemma 6.3). During stage $i+2$ only the score of one of them will increase (at the most), so the score of the second one will remain t (Lemma 6.4). Then, in stage $i+3$ this second vertex will be ranked above x, and the score of x will not grow (and remain $t+1$) (Lemma 6.5). This way, during 3 stages the score of x increases only by 1, whereas the score of p grows by 1 every stage.

Let us now state and prove the lemmas more formally.

LEMMA 6.2. *Let $x \in C \setminus \{p\}$ be a candidate such that $S_{i+1}(x) = t+1$ (where $t = ms(i)$). Let $G^i(x)$ be as before. Then at least one cycle in $G^i(x)$ that passes through x will stay after stage $i+1$, i.e., in G^{i+1}.*

PROOF. In Lemma 5.2 we have proved that in $G^i(x)$ at least one cycle passes through x. Since x appears in the preference list of $i+1$ above all the $\text{MIN}_i(x)$, it follows that each edge going out of x in $G^i(x)$, stays also in G^{i+1}. After we added x to the preference list of $i+1$, all the vertices in all the cycles passing through x were added in some order to the preference list of $i+1$, while they were with out-degree 0 at the time they were picked (it can be proved by induction on the length of the path from the vertex to x). Therefore, their "father" field was not null when they were picked. We have to prove that there is at least one cycle whose vertices were added in the reverse order (and then all the edges of the cycle stayed in G^{i+1}). Let $z_1 \in C \setminus \{p, x\}$ be some vertex such that $(x, z_1) \in G^i(x)$ and there is a path in $G^i(x)$ from z_1 to x. Let $z_2 = z_1.father$. As observed earlier, $z_2 \neq null$. We first show that when z_2 was picked by $i+1$, it was with out-degree 0. Indeed, if, by contradiction, we suppose otherwise, then z_2 would have been picked after z_1 (the proof is by induction on the length of the shortest path from vertex to x, that each vertex such that there is a path from it to x was picked before z_2), and this is a contradiction to the fact that $z_2 = z_1.father$. Therefore, the "father" field of z_2 after stage $i+1$ is not null.

Let $z_3 = z_2.father$. If $z_3 = x$ then we are done because we have found a cycle $x \to z_1 \to z_2 \to z_3 = x$ which was ranked in stage $i+1$ in the reverse order. Otherwise, by the same argument as before, we can show that when z_3 was picked, its out-degree was 0. This way we can pass from a vertex to its father until we reach p or null. We now show that we cannot reach p this way. Indeed, if, by contradiction, we reach p, then there is a path from x to p in G^i, and so all the vertices in this path, including x, were picked when their out-degree was 0, and this is a contradiction to the fact that the score of x went up in stage $i+1$. Therefore, we cannot reach p when we go from a vertex to its father starting with z_1. Now, let z_j be the last vertex before null in this path. We would like to show that $z_j = x$. If, by contradiction, z_j was picked before x by voter $i+1$, then all the vertices $z_{j-1}, \ldots, z_2, z_1$ would have been picked before x, when their out-degree is 0, and then x would have been picked when its out-degree is 0. This is a contradiction to the fact that x's score increased in stage $i+1$. Now suppose by contradiction that z_j was picked after x in stage $i+1$. Then all the vertices that have a path from them to x, including z_1, would have been picked before z_j in stage $i+1$, since the out-degree of z_j was greater than 0 when it was picked. This is a contradiction to the fact that z_j was picked before z_1. So, $z_j = x$. This way we got a cycle $x \to z_1 \to \ldots \to z_{j-1} \to x$ which was ranked in the reverse order in stage $i+1$. □

LEMMA 6.3. *Suppose that there are no 2-cycles in the graphs built by the algorithm. Let $x \in C \setminus \{p\}$ be a candidate such that $S_{i+1}(x) = t+1$ (where $t = ms(i)$), and let $G^i(x)$ be as described before Lemma 5.2. For each cycle U in $G^i(x)$, if U exists in G^{i+1}, i.e., after stage $i+1$, then there are 3 distinct vertices a, b, c, s.t. $(c,b) \in U$, $(b,a) \in U$ and $S_{i+1}(b) = N_{i+1}(b,a) = S_{i+1}(c) = N_{i+1}(c,b) = t$.*

PROOF. Let $U \subseteq E^i(x)$ be a cycle which stays also after $i+1$ stages. Let a be the vertex which in stage $i+1$ was chosen first among the vertices of U. Let b be the vertex before a in U, i.e., $(b,a) \in U$, and let c be the vertex before b in U, i.e., $(c,b) \in U$. Since there are no 2-cycles, a, b, c are all distinct vertices. Recall that for each $y \in V^i(x)$, $S_i(y) = t$. Since b was ranked below a in stage $i+1$, we have $S_{i+1}(b) = N_{i+1}(b,a) = N_i(b,a) = S_i(b) = t$. If c was chosen after b in stage $i+1$, then $S_{i+1}(c) = N_{i+1}(c,b) = N_i(c,b) = t$ and we are done. We now show that c cannot be chosen before b in stage $i+1$. If, by way of contradiction, c were chosen before b, since after ranking a, $d_{out}(b) = 0$, it follows that when c was picked, its out-degree was also 0. Hence, there exists $d \in \text{MIN}_i(c)$ which was picked by $i+1$ before c. And so, $S_{i+1}(c) = t$. On the other hand, since c was picked before b, we have $N_{i+1}(c,b) = t+1 > S_{i+1}(c)$, and so the edge (c,b) does not exist in G^{i+1}, a contradiction to the fact that the cycle U stayed after stage $i+1$. □

LEMMA 6.4. *Suppose that there are no 2-cycles in the graphs built by the algorithm. Let $x \in C \setminus \{p\}$ be a candidate such that $S_{i+1}(x) = t+1$ (where $t = ms(i)$). Then after stage $i+2$ at least one of the following will hold:*

1. *There will be a vertex w in G^{i+2} s.t. $p \in MIN_{i+2}(w)$ and there will be a path from x to w.*
2. *There will be a vertex w in G^{i+2} with $S_{i+2}(w) \leq t$, s.t. there will be a path from x to w.*

PROOF. 1. If there is a vertex w s.t. $p \in \text{MIN}_{i+1}(w)$ and there is a path from x to w in G^{i+1}, w.l.o.g. let us assume that w was picked first in stage $i+2$ among all such vertices. It is easy to see that $p \in \text{MIN}_{i+2}(w)$. If $x = w$, then trivially condition 1 holds, and we are done. Otherwise, in stage $i+2$, w was ranked above x. Let us build a chain of vertices, starting from x, by passing from a vertex to its father, as was assigned in the stage $i+2$. The chain stops when we reach p or null. If we reach p this way then we are done, because $a = b.father$ means that there is an edge (b, a) in G^{i-1}, and it stayed in G^{i+2} (because a was ranked above b). Now we show that we can't reach null this way. Suppose, for contradiction, that we reach null, and let z be the vertex before null in the chain. If z was ranked above w in stage $i+2$, then we get a contradiction since at the time of ranking z, $d_{out}(w) = 0$, whereas $d_{out}(z) > 0$, and we would prefer w over z. On the other hand, if w was ranked above z, then x should have been ranked above z too, since there is a path in G^{i+1} from x to w whereas $d_{out}(z) > 0$. So we got a contradiction since, by definition, z was ranked above x.

2. Now suppose that the condition in the first item does not hold. If there is a vertex w, s.t. $S_{i+1}(w) < t$ and there is a path in G^{i+1} from x to w, again, w.l.o.g. let us assume that w was picked first in stage $i+2$ among all such vertices. Then $S_{i+2}(w) \leq t$, and similarly to item 1 above, there is a path in G^{i+2} from x to w.

   Now let us suppose that the above conditions do not hold. Let us look at the vertex y which in stage $i+2$ was picked first from a cycle U s.t. there is a path from x to U, and there are two consecutive edges in U, each with weight t. By Lemma 6.3 and Lemma 6.2 such a vertex y exists. According to the algorithm (lines 21–22) and to the definition of y, before y only vertices s.t. there is no path from x to them, could be picked. Therefore, there is no path from y to earlier-picked vertices. So when y was picked, its out-degree was > 0, and hence all the edges going out of y stayed after stage $i+2$. According to the algorithm (lines 21–22), there is a vertex w s.t. $S_{i+1}(w) = N_{i+1}(w, y) = t$ and there is a path from y to w in G^{i+1}. Let W be the set of all such vertices w. According to the algorithm (lines 17–18), in stage $i+2$, all the vertices in W will be picked before all the vertices z with $S_{i+1}(z) = N_{i+1}(z, y) = t+1$.

   Now let us go from a vertex to its father, starting with x (like in Lemma 6.2) till we reach null (we cannot reach p this way since condition 1 does not hold). Similarly to Lemma 6.2, it can be verified that the last vertex before null is y. If we passed this way through some vertex $w \in W$ then we are done, since we got a path from x to w, and $S_{i+2}(w) = t$ (because w was picked in stage $i+2$ after y). Otherwise we are in the next situation: the path from x to U, connects to U through the vertex y (if this is not the case then we will pass through some $w \in W$ since according to the algorithm (lines 16–18 and 32), all the vertices which have a path from them to w will be picked before all other vertices w' with an edge (w', y) with weight $t+1$ and a path from y to w'). Let b be a vertex s.t. $(y, b) \in G^{i+1}$ (and so also $(y, b) \in G^{i+2}$) and there is a path in G^{i+1} from b to some $w \in W$. Since b belongs to a cycle with two edges of the weight t and there is a path from x to b, it follows that b was picked by $i+2$ after y. As there is a path from b to w, it follows that b was picked when $d_{out}(b) = 0$, and hence $b.father \neq null$. Like in Lemma 6.2, we go from a vertex to its father, starting with b, until we reach w. This way we got a path in G^{i+2} from x through y and b to w, and as mentioned earlier, $S_{i+2}(w) = t$.

□

The next lemma is central in the proof of Theorem 6.1. It states that the maximum score of p's opponents grows rather slowly.

LEMMA 6.5. *If there are no 2-cycles in the graphs built by the algorithm, then for all i, $0 \leq i \leq n-3$ it holds that $ms(i+3) \leq ms(i)+1$.*

PROOF. Let i, $0 \leq i \leq n-3$. Let $x \in C \setminus \{p\}$ be a candidate. Denote $ms(i) = t$. We need to prove that $S_{i+3}(x) \leq t+1$. If $S_{i+1}(x) \leq t$, then similarly to Lemma 5.3 we can prove that $S_{i+3}(x) \leq t+1$. So now we assume that $S_{i+1}(x) = t+1$. By Lemma 5.3, we have that $S_{i+2}(x) = t+1$. Suppose by contradiction that $S_{i+3}(x) = t+2$. x was ranked in stage $i+3$ at the place s^*. By Lemma 6.4 there exists a vertex w s.t. there is a path in G^{i+2} from x to w, and $p \in \text{MIN}_{i+2}(w)$ or $S_{i+2}(w) \leq t$. Then w was ranked in stage $i+3$ above the place s^*, because the score of x increased in stage $i+3$, and if, by contradiction, w was not ranked above the place s^*, then when we got to the place s^* we would prefer w over x. It is easy to see that all the vertices that have a path in G^{i+2} from them to w, and which were ranked below w in stage $i+3$, did not have their scores increased in that stage (since we took them one after another in the reverse order on their path to w when they were with out-degree 0). And as x was ranked below w, its score did not increase as well, and so $S_{i+3}(x) = S_{i+2}(x) = t+1$, a contradiction. □

LEMMA 6.6. *If the minimum number of manipulators needed to make p win is equal to 1, then Algorithm 1 performs optimally, i.e., finds the manipulation for $n = 1$.*

PROOF. Let us denote by *opt* the minimum number of manipulators needed to make p win the election. Let $S_i^*(a)$ denote the score of $a \in C$ after i manipulators voted in the optimal algorithm, and let $ms^*(i)$ be the maximum score of p's opponents after i manipulators voted in the optimal algorithm. Assume that $opt = 1$. If $S_0(p) > ms(0)$ then $opt = 0$, a contradiction. On the other hand, if $S_0(p) < ms(0)$, then $S_1^*(p) \leq ms(0) \leq ms^*(1)$, so p is not a unique winner after the manipulator voted, a contradiction. Therefore, $S_0(p) = ms(0)$. Also, $S_1^*(p) = ms(0)+1$ and $ms^*(1) = ms(0)$ (otherwise, p would not be a unique winner of the election). We need to show that $ms(1) = ms(0)$. Let $x \in C \setminus \{p\}$. If $S_0(x) < ms(0)$ then trivially $S_1(x) \leq ms(0)$ and we are done. Now suppose that $S_0(x) = ms(0)$. Suppose, by contradiction, that $S_1(x) = ms(0)+1$. Then when x was ranked by the first manipulator, $d_{out}(x) > 0$. Denote, as before, by V_x^0 the candidates that were ranked by Algorithm 1 below x, including x, in stage 1. For each $y \in V_x^0$, $S_0(y) = ms(0)$ and $d_{out}(y) > 0$. Therefore, if we put y instead of x, its score will increase to $ms(0)+1$. Let $b \in V_x^0$ be the candidate ranked highest among candidates in V_x^0 by the optimal algorithm. Then $ms^*(1) \geq S_1^*(b) = ms(0)+1$, contradicting the fact that $ms^*(1) = ms(0)$. □

We are now ready to prove the main theorem.

PROOF OF THEOREM 6.1. Let *opt* be as before. It is easy to see that $opt \geq ms(0) - S_0(p) + 1$. We shall prove first that Algorithm 1 will find a manipulation for $n = \left\lceil \frac{3ms(0) - 3S_0(p) + 3}{2} \right\rceil \leq \lceil \frac{3}{2} opt \rceil$. And indeed, by Lemma 6.5,

$$ms(n) \leq ms(0) + \left\lceil \frac{n}{3} \right\rceil = ms(0) + \left\lceil \frac{ms(0) - S_0(p) + 1}{2} \right\rceil.$$

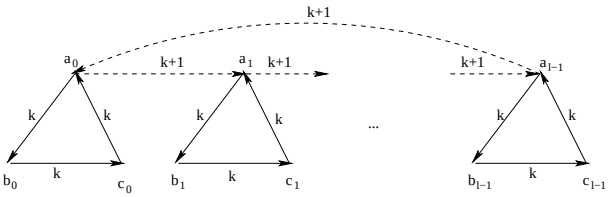


Figure 1: Example for lower bound on approximation ratio

Whereas,

$$\begin{aligned}
S_n(p) &= S_0(p) + n \\
&= S_0(p) + (ms(0) - S_0(p) + 1) + \left\lceil \frac{ms(0) - S_0(p) + 1}{2} \right\rceil \\
&= ms(0) + 1 + \left\lceil \frac{ms(0) - S_0(p) + 1}{2} \right\rceil \\
&> ms(0) + \left\lceil \frac{ms(0) - S_0(p) + 1}{2} \right\rceil \\
&\geq ms(n).
\end{aligned}$$

Now, by Lemma 6.6, when $opt = 1$, the algorithm performs optimally (i.e., finds the manipulation for $n = 1$). We have just proved that for $opt = 2$ the algorithm finds the manipulation when $n \leq \lceil \frac{3}{2} opt \rceil = 3$. When $opt = 3$, the algorithm finds the manipulation when $n \leq \lceil \frac{3}{2} opt \rceil = 5$. It is easy to see that for all $opt > 3$, $\lceil \frac{3}{2} opt \rceil < \frac{5}{3} opt$. Therefore, the approximation ratio of Algorithm 1 is $\leq \frac{5}{3} = 1\frac{2}{3}$. □

It is worth noting that from the proof above we have that as *opt* tends to infinity, the bound on the ratio between the size of the manipulating coalition returned by Algorithm 1 and *opt* tends to $1\frac{1}{2}$, since n is bounded by $\lceil 1\frac{1}{2} opt \rceil$.

THEOREM 6.7. *The $1\frac{2}{3}$-approximation ratio of Algorithm 1 is valid also when there are 2-cycles in the graphs built by the algorithm.*

We omit the proof of the above theorem due to space limitations.

7. LOWER BOUND ON THE APPROXIMATION RATIO OF THE ALGORITHM

THEOREM 7.1. *There is an asymptotic lower bound of $1\frac{1}{2}$ to the approximation ratio of Algorithm 1.*

PROOF. Consider the following example (see Figure 1). Let $m = |C|$ be of the form $m = 3^t + 1$ for an integer $t \geq 2$. Denote $l = 3^{t-1} = \frac{m-1}{3}$. Let $C = \{p, a_0, b_0, c_0, a_1, b_1, c_1, \ldots, a_{l-1}, b_{l-1}, c_{l-1}\}$. Let N be a multiple of 3, $N \geq 6$. Let $k = \frac{N}{3}$. $S_0(p) = 0$; for all $j, 0 \leq j \leq l-1$: $S_0(a_j) = N_0(a_j, b_j) = S_0(b_j) = N_0(b_j, c_j) = S_0(c_j) = N_0(c_j, a_j) = k$. In addition, for each $j, 0 \leq j \leq l-2$: $N_0(a_j, a_{j+1}) = k+1$, and $N_0(a_{l-1}, a_0) = k+1$. We first show that there exists a profile of non-manipulators that induces the above scores. We will have non-manipulator voters of 6 types; $\frac{N-3}{3}$ voters of each of the types (1), (2) and (3), and one voter of each of the types (4), (5) and (6). In all types, p is ranked in last place. To conserve space, we denote by A_j the fragment $a_j \succ c_j \succ b_j$ of the preference, by B_j the fragment $b_j \succ a_j \succ c_j$, and by C_j the fragment $c_j \succ b_j \succ a_j$. When showing the preference lists of the voters, it is convenient to use the trinary representation of the indices. We have candidates $\{p, a_0, b_0, c_0, a_1, b_1, c_1, \ldots, a_{22\ldots2}, b_{22\ldots2}, c_{22\ldots2}\}$. For preference list of type (1), we define the order $0 \succ_1 2 \succ_1 1$. In this preference list, we have the fragments A_j ordered by the order $\succ_1$, and p is at the end. In type (2), we have the fragments C_j ordered by the order $2 \succ_2 1 \succ_2 0$, with p at the end. In type (3), we have the fragments B_j ordered by the order $1 \succ_3 0 \succ_3 2$, with p at the end. In type (4), we have the fragments A_j ordered by $0 \succ_4 1 \succ_4 2$, with p at the end. In type (5), we have the fragments C_j ordered by $1 \succ_5 2 \succ_5 0$, with p at the end. Finally, in type (6) there are the fragments B_j ordered by $2 \succ_6 0 \succ_6 1$, with p at the end.

For instance, the next example illustrates the above profile for $t = 3$ ($m = 28$), and it easily generalizes to any $t \geq 2$.

$$\begin{aligned}
&(1){:}A_0 \succ A_2 \succ A_1 \succ A_{20} \succ A_{22} \succ A_{21} \succ A_{10} \succ A_{12} \succ A_{11} \succ p \\
&(2){:}C_{22} \succ C_{21} \succ C_{20} \succ C_{12} \succ C_{11} \succ C_{10} \succ C_2 \succ C_1 \succ C_0 \succ p \\
&(3){:}B_{11} \succ B_{10} \succ B_{12} \succ B_1 \succ B_0 \succ B_2 \succ B_{21} \succ B_{20} \succ B_{22} \succ p \\
&(4){:}A_0 \succ A_1 \succ A_2 \succ A_{10} \succ A_{11} \succ A_{12} \succ A_{20} \succ A_{21} \succ A_{22} \succ p \\
&(5){:}C_{11} \succ C_{12} \succ C_{10} \succ C_{21} \succ C_{22} \succ C_{20} \succ C_1 \succ C_2 \succ C_0 \succ p \\
&(6){:}B_{22} \succ B_{20} \succ B_{21} \succ B_2 \succ B_0 \succ B_1 \succ B_{12} \succ B_{10} \succ B_{11} \succ p
\end{aligned}$$

It could be verified that the graph G^0 which matches the above profile looks as in Figure 1 (we omitted some of the dotted edges).

Now we will show that for the above example the approximation ratio of the algorithm is at least $1\frac{1}{2}$. Consider the following preference list of the manipulators:

$$\begin{aligned}
&p \succ A_{l-1} \succ A_{l-2} \succ \ldots \succ A_0 \\
&p \succ A_{l-2} \succ A_{l-3} \succ \ldots \succ A_0 \succ A_{l-1} \\
&p \succ A_{l-3} \succ A_{l-4} \succ \ldots \succ A_0 \succ A_{l-1} \succ A_{l-2} \\
&\ldots
\end{aligned}$$

It can be verified that in the above preference list, the maximum score of p's opponents ($ms(i)$) grows by 1 every $\frac{m-1}{3}$ stages (starting with $k+1$). In addition, p's score grows by 1 every stage. Therefore, when we apply the voting above, the minimum number of stages (manipulators) n^* needed to make p win the election should satisfy $n^* > k + 1 + \lceil \frac{3n^*}{m-1} \rceil$. Since $\lceil \frac{3n^*}{m-1} \rceil < \frac{3n^*}{m-1} + 1$, the sufficient condition for making p win is:

$$n^* > k + 1 + \frac{3n^*}{m-1} + 1.$$

So, we have,

$$\begin{aligned}
(m-1)n^* &> (m-1)(k+2) + 3n^* \\
(m-4)n^* &> (m-1)(k+2) \\
n^* &> \frac{(m-1)(k+2)}{m-4}.
\end{aligned}$$

For large-enough m, $\frac{(m-1)(k+2)}{m-4} < k+3$, and so $n^* = k+3$ would be enough to make p win the election.

Now let us examine what Algorithm 1 will do when it gets this example as input. One of the possible outputs of the algorithm looks like this:

$$\begin{aligned}
&p \succ C_0 \succ C_1 \succ \ldots \succ C_{l-1} \\
&p \succ B_1 \succ B_2 \succ \ldots \succ B_{l-1} \succ B_0 \\
&p \succ A_2 \succ A_3 \succ \ldots \succ A_{l-1} \succ A_0 \succ A_1 \\
&p \succ C_3 \succ C_4 \succ \ldots \succ C_{l-1} \succ C_0 \succ C_1 \succ C_2 \\
&\ldots
\end{aligned}$$

It can be verified that in the above preference list, $ms(i)$ grows by 1 every 3 stages, and p's score grows by 1 every stage. Therefore, the number of stages n returned by Algorithm 1 that are needed to make p win the election satisfies $n > k + \lceil \frac{n}{3} \rceil$. Since $\lceil \frac{n}{3} \rceil \geq \frac{n}{3}$, the necessary condition for making p win the election is:

$$n > k + \frac{n}{3}.$$

And then we have,

$$3n > 3k + n$$
$$2n > 3k$$
$$n > \frac{3}{2}k$$

So we find that the ratio $\frac{n}{n^*}$ tends to $1\frac{1}{2}$ as m and N (and k) tend to infinity. □

8. DISCUSSION

In spite of the popularity of the approach of using computational complexity as a barrier against manipulation, this method has an important drawback: although for some voting rules the manipulation problem has been proven to be $\mathcal{NP}$-complete, these results apply only to the worst case instances; for most instances, the problem could be computationally easy. There is much evidence that this is indeed the case, including work by Friedgut et al. [7], and Isaksson et al. [9]. They prove that a single manipulator can manipulate elections with relatively high probability by simply choosing a random preference. This is true when the voting rule is far from a dictatorship, in some well-defined sense.

Additional evidence for the ease of manipulating elections on average is the work of Procaccia and Rosenschein [10], and Xia and Conitzer [12]. They connected the frequency of manipulation with the fraction of manipulators out of all the voters. Specifically, they found that for a large variety of distributions of votes, when $n = o(\sqrt{N})$, then with high probability the manipulators can affect the outcome of the elections. The opposite is true when $n = \omega(\sqrt{N})$.

The current work continues this line of research. It strengthens the results of Zuckerman et al. [14], giving an algorithm with a better approximation ratio for the Unweighted Coalitional Optimization (UCO) problem under Maximin. Equivalently, it narrows the error window of the algorithm for the decision problem CCUM under Maximin. The result can be viewed as another argument in favor of the hypothesis that most rules are usually easy to manipulate.

9. CONCLUSIONS AND FUTURE WORK

We introduced a new algorithm for approximating the UCO problem under the Maximin voting rule, and investigated its approximation guarantees. In future work, it would be interesting to prove or disprove that Algorithm 1 presented in [14] has an approximation ratio of $1\frac{2}{3}$, for those instances where there is no Condorcet winner.[2] Another direction is to implement both algorithms, so as to empirically measure and compare their performance.

Acknowledgments

We would like to thank Reshef Meir, Aviv Zohar, Jerome Lang and Noam Nisan for helpful discussions on topics of this research. We also thank Edith Elkind and Piotr Faliszewski for valuable comments on an earlier version of this paper. This work was partially supported by Israel Science Foundation grant #898/05, and Israel Ministry of Science and Technology grant #3-6797.

[2]We have an example showing that that algorithm is no better than a 2-approximation when there is a Condorcet winner.

10. REFERENCES

[1] J. Bartholdi and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

[2] J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[3] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007.

[4] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of SAGT 2009, Springer-Verlag LNCS 5814*, pages 299–310, October 2009.

[5] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, pages 768–773, July 2010.

[6] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. In *The Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 128–133, Pasadena, California, July 2009.

[7] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *Proc. 49th FOCS. IEEE Computer Society Press*, pages 243–249, 2008.

[8] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[9] M. Isaksson, G. Kindler, and E. Mossel. The geometry of manipulation — a quantitative proof of the Gibbard-Satterthwaite theorem. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 319–328, October 2010.

[10] A. D. Procaccia and J. S. Rosenschein. Average-case tractability of manipulation in elections via the fraction of manipulators. In *The Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pages 718–720, Honolulu, Hawaii, May 2007.

[11] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[12] L. Xia and V. Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of ACM EC-08*, pages 109–118, July 2008.

[13] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *The Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 348–353, Pasadena, California, July 2009.

[14] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. *Journal of Artificial Intelligence*, 173(2):392–412, February 2009.

Algorithm 1 Decides CCUM for Maximin voting rule

procedure MAXIMIN(C, p, X_S, n) ▷ X_S is the set of preferences of voters in S, n is the number of voters in T
  $X \leftarrow \emptyset$ ▷ Will contain the preferences of T
  for $i = 1, \ldots, n$ **do** ▷ Iterate over voters
    $P_i \leftarrow (p)$ ▷ Put p at the first place of the i-th preference list
    Build a digraph $G^{i-1} = (V, E^{i-1})$ ▷ $V = C \setminus \{p\}$, $(x, y) \in E^{i-1}$ iff ($y \in \text{MIN}_{i-1}(x)$ and $p \notin \text{MIN}_{i-1}(x)$)
    for $c \in C \setminus \{p\}$ **do** ▷ This for loop is used in the algorithm's analysis
      if $d_{out}(c) = 0$ **then**
        $c.father \leftarrow p$
      else
        $c.father \leftarrow null$
      end if
    end for
    while $C \setminus P_i \neq \emptyset$ **do** ▷ while there are candidates to be added to i-th preference list
      Evaluate the score of each candidate based on the votes of S and $i-1$ first votes of T
      if there exists a set $A \subseteq C \setminus P_i$ with $d_{out}(a) = 0$ for each $a \in A$ **then** ▷ if there exist vertices in the digraph G^{i-1} with out-degree 0
        Add the candidates of A to the stacks Q_j, where to the same stack go candidates with the same score
        $b \leftarrow Q_1.popfront()$ ▷ Retrieve the top-most candidate from the first stack—with the lowest scores so far
        $P_i \leftarrow P_i + \{b\}$ ▷ Add b to i's preference list
      else
        Let $s \leftarrow \min_{c \in C \setminus P_i}\{S_{i-1}(c)\}$
        if there is a cycle U in G^{i-1} s.t. there are 3 vertices a, b, c, s.t. $(c, b), (b, a) \in U$, and $S_{i-1}(c) = S_{i-1}(b) = s$ **then**
          $P_i \leftarrow P_i + \{b\}$ ▷ Add b to i's preference list
        else
          Pick $b \in C \setminus P_i$ s.t. $S_{i-1}(b) = s$ ▷ Pick any candidate with the lowest score so far
          $P_i \leftarrow P_i + \{b\}$ ▷ Add b to i's preference list
        end if
      end if
      for $y \in C \setminus P_i$ **do**
        if $(y, b) \in E^{i-1}$ **then** ▷ If there is a directed edge from y to b in the digraph
          Remove all the edges of E^{i-1} originating in y
          $y.father \leftarrow b$ ▷ This statement is used in the algorithm's analysis
          Add y to the front of the appropriate stack Q_j—according to $S_{i-1}(y)$
        end if
      end for
    end while
    $X \leftarrow X \cup \{P_i\}$
  end for
  $X_T \leftarrow X$
  if argmax$_{c \in C}\{$Score of c based on $X_S \cup X_T\} = \{p\}$ **then**
    return true ▷ p wins
  else
    return false
  end if
end procedure

Computational Complexity of Two Variants of the Possible Winner Problem

Dorothea Baumeister Magnus Roos Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
{baumeister, roos, rothe}@cs.uni-duesseldorf.de

ABSTRACT

A possible winner of an election is a candidate that has, in some kind of incomplete-information election, the possibility to win in a complete extension of the election. The first type of problem we study is the Possible co-Winner with respect to the Addition of New Candidates (PcWNA) problem, which asks, given an election with strict preferences over the candidates, is it possible to make a designated candidate win the election by adding a limited number of new candidates to the election? In the case of unweighted voters we show NP-completeness of PcWNA for a broad class of pure scoring rules. We will also briefly study the case of weighted voters. The second type of possible winner problem we study is Possible Winner/co-Winner under Uncertain Voting System (PWUVS and PcWUVS). Here, uncertainty is present not in the votes but in the election rule itself. For example, PcWUVS is the problem of whether, given a set C of candidates, a list of votes over C, a distinguished candidate $c \in C$, and a class of election rules, there is at least one election rule from this class under which c wins the election. We study these two problems for a class of systems based on approval voting, the family of Copeland$^\alpha$ elections, and a certain class of scoring rules. Our main result is that it is NP-complete to determine whether there is a scoring vector that makes c win the election, if we restrict the set of possible scoring vectors for an m-candidate election to those of the form $(\alpha_1, \ldots, \alpha_{m-4}, x_1, x_2, x_3, 0)$, with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Economics, Theory

Keywords

voting protocols, social choice theory, computational social choice, possible winner problem

Cite as: Computational Complexity of Two Variants of the Possible Winner Problem, Dorothea Baumeister, Magnus Roos, and Jörg Rothe, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 853-860.

1. INTRODUCTION

A central task in computational social choice is the study of the algorithmic and computational properties of voting systems (see, e.g., the bookchapters [15, 4]). One of the classical problems in this field is the Manipulation problem, which deals with the question of whether a voter can benefit from strategic behavior. The celebrated Gibbard–Satterthwaite theorem [18, 23] says that in every nondictatorial voting system a strategic voter can alter the outcome of an election to his or her advantage by voting insincerely. Bartholdi et al. [2, 1] were the first to show that computational complexity can be used as a barrier to protect elections from manipulation attempts: In some voting systems, though manipulable in principle, it is computationally hard to compute successful manipulative preferences to cast.

Conitzer, Sandholm, and Lang [11] defined a more general version of this problem, called Coalitional Weighted Manipulation, where voters have weights and a whole group of voters can coordinate their strategic efforts. The complexity of this problem has been studied for many voting systems, including plurality, Borda, veto, Copeland, STV, maximin, plurality with run-off, regular cup, randomized cup, and including a dichotomy result for the class of pure scoring rules [11, 19]. In the case of unweighted voters the complexity of coalitional manipulation is still unknown for most pure scoring rules.

Another generalization of Manipulation is the Possible Winner (PW) problem, which was first introduced by Konczak and Lang [21]. Here the voters do not provide linear orders over the candidates, but partial orders. The question is whether there is an extension of the partial orders into linear ones such that a distinguished candidate wins the election. Manipulation is the special case of PW in which all voters but one report linear orders and one voter reports no preference at all. This implies that NP-hardness results for the Manipulation problem carry over to the PW problem. For the important class of pure scoring rules and the case of unweighted voters, the computational complexity of this problem is also settled by a full dichotomy result (see [6, 5]): It is solvable in polynomial time for plurality and veto, and NP-complete for all other pure scoring rules. These results also hold for PcW, the corresponding co-winner problem.

One variant of the PW problem was defined by Chevaleyre et al. [9] and also studied by Xia et al. [25]: Possible co-Winner with respect to the Addition of New Candidates (PcWNA). In this setting the voters report linear orders over an initial set of candidates and after reporting their preferences some new candidates are introduced. The

problem is to determine whether one distinguished candidate among the initial ones can be a winner if the voters' preferences are extended to linear orders over the initial and the new candidates. PcWNA is a special case of PcW and is in some sense dual to the coalitional manipulation problem [9, 25]. In particular, the NP-hardness results for the PcW problem are not inherited by PcWNA. Note that PcWNA is also closely related to—but different from—the problem of control via adding candidates [3, 20] and to the cloning problem in elections [13].

We study the problem PcWNA in the case of unweighted voters and pure scoring rules, giving a deeper insight into a question raised by Chevaleyre et al. [9]. They showed that if one new candidate is added in the case of unweighted voters, PcWNA is polynomial-time solvable for a certain class of pure scoring rules but is NP-complete for one specific pure scoring rule (see Table 1), and they asked if that result can be extended to other pure scoring rules. Our main result in Section 3 establishes NP-completeness of PcWNA for a whole class of pure scoring rules if one new candidate is added. This result is obtained even for the case of unweighted voters. In addition, we briefly study the complexity of the PcWNA problem in the case of weighted voters.

In the second setting we consider, the possible winner problem is related to uncertainty about the election rule used. A similar setting has been previously studied by several authors. Conitzer, Sandholm, and Lang [11] showed that the computational complexity of manipulation can be increased by using a random instantiation for the cup protocol. Pini et al. [22] studied the problem of determining winners by sequential majority voting if preferences may be incomplete and the agenda is uncertain.

In general, we study the problem Possible Winner/co-Winner under Uncertain Voting System (PWUVS and PcWUVS), which asks whether a distinguished candidate, after all votes have been cast, can be made a winner of the election by choosing one election rule from a given class of rules. Specifically we will consider this problem with respect to a class of systems based on approval voting, the family of Copeland$^\alpha$ elections [14], and a certain class of scoring rules. Walsh [24] proposed to investigate PWUVS for the class of scoring rules, but to the best of our knowledge this issue has not been studied before. As a main result in Section 4, we show that PcWUVS and PWUVS are NP-hard for scoring rules if we restrict the set of possible scoring vectors for an m-candidate election, $m \geq 4$, to those of the form $(\alpha_1, \ldots, \alpha_{m-4}, x_1, x_2, x_3, 0)$, with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$. Note that some important scoring rules, such as Borda and veto for $m \geq 4$ candidates, are contained in this restricted set of scoring vectors.

A motivation for uncertainty about the voting system used is that this may prevent the voters from attempting to manipulate the election, since reporting an insincere preference might result in a worse outcome for them. For example, consider an election with three candidates (a, b, and c), nine sincere voters (six cast the vote $c > a > b$, two $b > a > c$, and one $b > c > a$), and three strategic voters (whose true preferences are $a > b > c$). If the strategic voters would know for sure that the election is held under the plurality rule (which values a first position by one point and all other positions by zero points), they might have an incentive to not waste their votes by voting sincerely ($a > b > c$) but rather to help their second preferred candidate, b, to tie for winner with c by casting the three votes $b > a > c$. However, if the election is held under the Borda rule (which, for three candidates, values a first position by two points, a second position by one point, and a last position by zero points), casting the three insincere votes $b > a > c$ would make their most despised candidate c win with 13 points in total (leaving b second with 12 points and a last with 11 points), whereas the three sincere votes $a > b > c$ would make their favorite candidate a win with 14 points in total (leaving c second with 13 points and b last with 9 points). This means that uncertainty about the scoring rule may give the voters a strong incentive to reveal their true preferences.

2. DEFINITIONS AND NOTATION

An election (C, V) is given by a set C of candidates and a list V of votes over C. In preference-based voting systems, each vote in V is a (strict) linear ordering of the candidates in C, where the underlying binary relation $>$ on C is *total* (either $c > d$ or $d > c$ for all $c, d \in C$, $c \neq d$), *transitive* (for all $c, d, e \in C$, if $c > d$ and $d > e$ then $c > e$), and *asymmetric* (for all $c, d \in C$, if $c > d$ then $d > c$ does not hold). Here, $c > d$ means that candidate c is (strictly) preferred to candidate d. A voting system is a rule to determine the winner(s) of an election. We will consider three different types of voting systems: (pure) scoring rules, Copeland$^\alpha$ elections, and (variants of) approval voting.

Scoring rules (a.k.a. scoring protocols): Each scoring rule with m candidates is specified by an m-dimensional scoring vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ satisfying that

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m, \tag{1}$$

where each weight α_j is a nonnegative integer. For an election (C, V), a candidate $c \in C$ ranked at jth position in a vote $v \in V$ receives α_j points from v. The *score of c in* (C, V), denoted by $score_{(C,V)}(c)$, is the sum of all points c receives from all voters in V, and the winners of (C, V) are the candidates with maximum score. We may assume that the last weight, α_m, in the scoring vector is always zero, since each scoring rule not satisfying this condition can easily be transformed into one that satisfies it (see [19]). Adopting a notion introduced by Betzler and Dorn [6], we say a scoring rule is *pure* if for each $m \geq 2$, the scoring vector for m candidates can be obtained from the scoring vector for $m - 1$ candidates by inserting one additional weight at any position subject to satisfying (1). One class of pure scoring rules is *k-approval*. Here the scoring vector has a one in the first k positions and a zero in all remaining positions. 1-approval—which may be better known under the name *plurality*—has the vector $(1, 0, \ldots, 0)$, and $(m-1)$-approval for m candidates—which may be better known under the name *antiplurality* or *veto*—has the vector $(1, \ldots, 1, 0)$. Another prominent scoring rule is the Borda rule, which has the scoring vector $(m-1, m-2, \ldots, 1, 0)$ for m candidates.

Copeland$^\alpha$, for a rational number α, $0 \leq \alpha \leq 1$: The winners are determined by pairwise comparisons of the candidates. For each $c \in C$, let $win(c)$ denote the number of candidates c beats in a pairwise comparison, and let $tie(c)$ denote the number of candidates c ties with in a pairwise comparison. The *Copeland$^\alpha$ score of a candidate c* is $win(c) + \alpha \cdot tie(c)$, and the candidates with maximum score win the election.

Approval voting: Every voter either approves or disapproves of each candidate, and the *approval score of a candi-*

date is the sum of his or her approvals. The candidates with the highest approval score win the election.

In the above voting systems, if there is only one candidate with maximum score, he or she is the unique winner.

The Possible co-Winner with respect to the Addition of New Candidates problem for a given voting system $\mathcal{E}$ is defined as follows:

Name: $\mathcal{E}$-Possible co-Winner with respect to the Addition of New Candidates ($\mathcal{E}$-PcWNA).
Given: A set of candidates $C = \{c_1, \ldots, c_m\}$, a list of votes $V = \{v_1, \ldots, v_n\}$ that are linear orders over C, a set C' with $|C'| = k$, $k \in \mathbb{N}$, of new candidates, and a distinguished candidate $c \in C$.
Question: Is there an extension of the votes in V to linear orders over $C \cup C'$ such that c is a winner of the election held under voting system $\mathcal{E}$.

In contrast to the above-defined problem where uncertainty is in the preferences, in Section 4.1 we will study another possible winner problem where uncertainty is in the voting system itself. This is the Possible co-Winner under Uncertain Voting System problem for a given class $\mathcal{V}$ of voting systems, which formally is defined as follows:

Name: $\mathcal{V}$-Possible co-Winner under Uncertain Voting System (PcWUVS).
Given: An election $E = (C, V)$, with the set of candidates C, a list of voters V consisting of linear orders over C, and a distinguished candidate $c \in C$.
Question: Is there a voting system $\mathcal{E}$ in $\mathcal{V}$ such that c is a winner of the election held under $\mathcal{E}$?

The problem is stated for the co-winner case. The unique-winner variant, PWUVS, is defined analogously by replacing "a winner" by "the unique winner" in the Question field above.

For the study of the computational complexity of the problems defined above, we will always assume that voters are unweighted and that the number of both voters and candidates is unbounded, unless stated otherwise.

3. POSSIBLE WINNER WRT. THE ADDITION OF NEW CANDIDATES

3.1 Unweighted Voters

In this section we study the problem PcWNA for pure scoring rules in the case of unweighted voters. Table 1 shows the results about the complexity of PcWNA for pure scoring rules that are already known from earlier work [9, 10, 25], where it is always assumed that voters are unweighted and that the number of initial candidates is unbounded. In particular, PcWNA is in P for the Borda rule for any fixed number of candidates, yet is NP-complete for the scoring vector $(3, 2, 1, 0, \ldots, 0)$ when the number of candidates is unbounded. Thus, this NP-completeness result is about a more general problem and does not contradict with the polynomial-time solvability of Borda in the restricted case of four candidates.

We now extend the result of Chevaleyre et al. [9] that PcWNA is NP-complete for pure scoring rules with vector $(3, 2, 1, 0, \ldots, 0)$ when one new candidate is added by showing that NP-completeness of PcWNA holds even for the class of pure scoring rules of the form $(\alpha_1, \alpha_2, 1, 0, \ldots, 0)$ with $\alpha_1 > \alpha_2 > 1$.

Scoring rule	PcWNA
Plurality	in P (see [9])
Veto	in P (see [9])
Borda	in P (see [9])
2-Approval	in P (see [10])
k-Approval, $\lvert C'\rvert \leq 2$	in P (see [9, 10])
k-Approval, $k \geq 3$, $\lvert C'\rvert \geq 3$	NP-complete (see [9, 10])
$(\alpha_i - \alpha_{i+1}) \leq (\alpha_{i+1} - \alpha_{i+2})$, $1 \leq i \leq m-2$	in P (see [9])
$(3, 2, 1, 0, \ldots, 0)$, $\lvert C'\rvert = 1$	NP-complete (see [9])

Table 1: Previous results on the complexity of PcWNA for pure scoring rules.

Theorem 3.1. PcWNA *is* NP*-complete for pure scoring rules of the form* $(\alpha_1, \alpha_2, 1, 0, \ldots, 0)$ *with* $\alpha_1 > \alpha_2 > 1$*, if one new candidate is added.*

Proof. Membership in NP is obvious, and the proof of NP-hardness is by a reduction from the NP-complete 3-DM problem, which is defined as follows (see [17]):

Name: Three-Dimensional Matching (3-DM).
Given: A set $M \subseteq W \times X \times Y$, with $W = \{w_1, \ldots, w_q\}$, $X = \{x_1, \ldots, x_q\}$, and $Y = \{y_1, \ldots, y_q\}$.
Question: Is there a subset $M' \subseteq M$ with $|M'| = q$, such that no two elements of M' agree in any coordinate?

Let $M \subseteq W' \times X' \times Y'$ be an instance of 3-DM with $W' = \{w'_1, \ldots, w'_q\}$, $X' = \{x'_1, \ldots, x'_q\}$, and $Y' = \{y'_1, \ldots, y'_q\}$, where $m = |M|$. Let $p(s)$ be the number of elements in M in which $s \in W' \cup X' \cup Y'$ occurs.

Construct an instance of the PcWNA problem with the election (C, V) having the set $C = W \cup X \cup Y \cup \{b, c\} \cup D$ of candidates, with $W = \{w_1, \ldots, w_q\}$, $X = \{x_1, \ldots, x_q\}$, and $Y = \{y_1, \ldots, y_q\}$. The new candidate to be added is a, so $C' = \{a\}$. D contains only dummy candidates, needed to pad the votes so as to make the reduction work. Table 2 shows the list $V = V_1 \cup V_2 \cup V_3 \cup V_4$ of votes. Note that only the first three candidates of each vote will be specified, since all other candidates do not receive any points. The numbers behind each vote denote their multiplicity. All places that need to be filled by a dummy candidate will be indicated by d (with no explicit subscript specified). Note that it is possible to substitute the d's by a polynomial number of dummy candidates such that none of them receives more than $q\alpha_1$ points.

V_1	$w_i > x_j > y_k$	1, $\forall (w'_i, x'_j, y'_k) \in M$
V_2	$w_i > d > d$	$q + m + 1 - p(w'_i)$, $\forall w_i \in W$
	$d > d > x_i$	$(q+m)\alpha_1 + (2 - p(x'_i))\alpha_2 - 1$, $\forall x_i \in X$
	$d > d > y_i$	$(q+m)\alpha_1 + \alpha_2 + 1 - p(y'_i)$, $\forall y_i \in Y$
V_3	$c > d > d$	$q + m$
	$d > c > d$	1
V_4	$d > d > b$	$(q + 2m)\alpha_1 + 2\alpha_2$

Table 2: Construction for the proof of Theorem 3.1.

The scores of the single candidates in election (C, V) are:

$$\begin{aligned} score_{(C,V)}(c) &= (q+m)\alpha_1 + \alpha_2, \\ score_{(C,V)}(w_i) &= (q+m+1)\alpha_1, \quad 1 \le i \le q, \\ score_{(C,V)}(x_i) &= (q+m)\alpha_1 + 2\alpha_2 - 1, \quad 1 \le i \le q, \\ score_{(C,V)}(y_i) &= (q+m)\alpha_1 + \alpha_2 + 1, \quad 1 \le i \le q, \\ score_{(C,V)}(b) &= (q+2m)\alpha_1 + 2\alpha_2, \\ score_{(C,V)}(d) &< (q+m)\alpha_1 + \alpha_2, \quad \forall d \in D. \end{aligned}$$

Note that $score_{(C,V)}(d) < score_{(C,V)}(c)$ for all dummy candidates $d \in D$.

We claim that c is a possible winner (i.e., a can be inserted such that c wins in the election held over the candidates $C \cup C'$) if and only if there is a matching M' for the 3-DM instance M.

($\Leftarrow$) Assume that there exists a matching M' for M. Extend the votes in V to V', where a is inserted at a position with zero points in all votes of V_2 and V_3, and the votes in V_1 and V_4 are extended as shown in Table 3:

V_1	$a > w_i > x_j > y_k$	$1, \forall(w_i', x_j', y_k') \in M'$
	$w_i > x_j > y_k > a$	$1, \forall(w_i', x_j', y_k') \in M \setminus M'$
V_4	$d > d > a > b$	$m\alpha_1 + \alpha_2$
	$d > d > b > a$	$(q+m)\alpha_1 + \alpha_2$

Table 3: Showing ($\Leftarrow$) in the proof of Theorem 3.1.

Then all candidates except the dummy candidates have exactly $(q+m)\alpha_1+\alpha_2$ points. Hence c has the highest score and is a winner of the election.

($\Rightarrow$) Assume that c is a winner of the election $(C \cup C', V')$, where V' is an extension of the linear votes in V. This implies that the score of all other candidates in this election is less than or equal to the score of c. The score of c will always be $(q+m)\alpha_1+\alpha_2$, since c gets all of his or her points from the voters in V_3, where he or she is placed at the top position in $m+q$ votes and at second position in one vote.

Since $score_{(C,V)}(w_i) = (q+m+1)\alpha_1$ points, each of the candidates w_i, $1 \le i \le q$, must lose at least $\alpha_1 - \alpha_2$ points when inserting a. Due to the requirement that $\alpha_1 > \alpha_2$, each w_i has to take at least one second position in a vote where he or she was ranked first originally. For the candidates x_i, $1 \le i \le q$, we have $score_{(C,V)}(x_i) = (q+m)\alpha_1 + 2\alpha_2 - 1$. Again, since $\alpha_2 > 1$, each x_i must lose at least $\alpha_2 - 1$ points, and since $score_{(C,V)}(y_i) = (q+m)\alpha_1 + \alpha_2 + 1$, each y_i must lose at least one point so as to not beat c.

The new candidate a can get at most $(q+m)\alpha_1 + \alpha_2$ points, since otherwise a would beat c.

To prevent w_i, $1 \le i \le q$, from beating c, a must be placed in a first position in q votes of V_1 or V_2. Then a can get at most $m\alpha_1 + \alpha_2$ points from the remaining votes without beating c. In the current situation, b would beat c by $m\alpha_1+\alpha_2$ points. So a must take $m\alpha_1+\alpha_2$ third positions in these votes such that b has a score of $(q+m)\alpha_1+\alpha_2$. Then the score of a is $(q+m)\alpha_1 + \alpha_2$. Since we assumed that c is a winner of the election, every x_i, $1 \le i \le q$, must end up having $\alpha_2 - 1$ points less, and every y_i, $1 \le i \le q$, must end up having one point less. This is possible only if a is at the first position in some vote from V_1. Hence the q first positions of a must shift every candidate x_i and y_i by one position to the right. Then the triples corresponding to the three elements w_i, x_j, and y_k corresponding to these q votes must form a matching for the 3-DM instance M. ❑

3.2 Weighted Voters

In this section we study the case of weighted voters for the PcWNA problem. Obviously, all NP-hardness results obtained for PcWNA in the case of unweighted voters also hold in the case of weighted voters. However, the polynomial-time algorithms for the case of unweighted voters cannot directly be transferred to the weighted-voters case. In fact, we will show NP-hardness of PcWNA in the weighted case for some voting rules where this problem is known to be polynomial-time solvable in the unweighted case. Specifically, we will consider the plurality rule for weighted voters in this section. For plurality, polynomial-time algorithms are known for PW in the case of unweighted voters, and for Manipulation both in the unweighted-voters and in the weighted-voters case. In contrast, we now show that PcWNA is NP-complete for plurality in the case of weighted voters, even if there are only two initial candidates and one new candidate to be added.

Theorem 3.2. PcWNA *is* NP*-complete for plurality in the case of weighted voters, even if there are only two initial candidates and one new candidate to be added.*

Proof. Membership in NP is obvious. To show NP-hardness of PcWNA for plurality in the case of weighted voters, we now give a reduction from the NP-complete Partition problem, which is defined as follows (see [17]):

Name: Partition.
Given: A nonempty, finite sequence $(s_1, s_2, \ldots, s_n)$ of positive integers.
Question: Is there a subset $A' \subset A = \{1, 2, \ldots, n\}$ such that

$$\sum_{i \in A'} s_i = \sum_{i \in A \setminus A'} s_i \, ?$$

For a given Partition instance $(s_1, \ldots, s_n)$, let $\sum_{i \in A} s_i = 2K$, where $A = \{1, 2, \ldots, n\}$. We construct an election (C, V) with the set of candidates $C = \{c, d\}$, where c is the distinguished candidate, and the list of votes $V = V_1 \cup V_2$ with the corresponding weights as shown in Table 4.

V_1	$c > d$	one vote of weight K
V_2	$d > c$	one vote of weight s_i for each $i \in A$

Table 4: Construction for the proof of Theorem 3.2.

The new candidate to be added is a, so $C' = \{a\}$. In the initial situation, the score of candidate c is K, and candidate d receives $2K$ points and hence wins the election. We now show that c can be made a winner by introducing candidate a into the election if and only if there is a partition for the given Partition instance.

($\Leftarrow$) Assume that there is a subset $A' \subset A$ such that $\sum_{i \in A'} s_i = \sum_{i \in A \setminus A'} s_i$. If the new candidate a is placed at the first position in each of those votes from V_2 that correspond to the $i \in A'$, and at the last position in all remaining votes, then the score of all three candidates is exactly K, and c is a co-winner of the election.

($\Rightarrow$) Assume that c is a winner of the election, after candidate a has been introduced. It must hold that candidates a and d receive at most K points. Hence candidate d must

lose K points due to inserting candidate a. This is possible only if a is placed at the first position in some votes from V_2 with a total weight of K. These votes now correspond to a valid partition. ❑

Next, we study 2-approval and give in Theorem 3.3 a result for the case of weighted voters and an unbounded number of candidates.

THEOREM 3.3. PCWNA *is* NP*-complete for* 2*-approval in the case of weighted voters, where the number of candidates is unbounded and one new candidate is to be added.*

PROOF. To prove the problem NP-hard, we again give a reduction from PARTITION, which was defined in the proof of Theorem 3.2. Let $(s_1, \ldots, s_n)$ be an instance of PARTITION with $\sum_{i \in A} s_i = 2K$, where $A = \{1, 2, \ldots, n\}$.

We introduce a set C of $n+3$ candidates:

- c (the candidate we want to win),
- b (the candidate who wins the original election), and
- a set $\{d_0, d_1, \ldots, d_n\}$ of dummy candidates.

The votes are specified as follows:

- For each s_j, we define a vote $d_j > b > \overline{C}$ with weight s_j, where $\overline{C}$ denotes the set of candidates not yet mentioned in the vote, so in this case we have $\overline{C} = C \setminus \{b, d_j\}$. Note that the ranking of the candidates $\overline{C}$ cannot influence the outcome of the election, since we deal with 2-approval.
- There is one vote $c > d_0 > \overline{C}$ with weight K.

Since $\sum_{j \in A} s_j = 2K$, candidate b has a score of $2K$ and wins the election.

We now prove that c can be made a winner by adding one new candidate, a, if and only if there is a subset $A' \subset A$ that induces a valid partition for the given instance.

($\Leftarrow$) Suppose we have a partition $A' \subset A$. By putting a in the first position of each vote having a weight of s_i and for which $i \in A'$, a will get exactly K points. Furthermore, b loses these K points, since he or she moves to the third position in these votes. Now there is a tie between a, b, c, and d_0, each having K points. Since $s_j \leq K$, $1 \leq j \leq n$, no candidate d_j, $1 \leq j \leq n$, has a higher score. Thus, c is a co-winner of the election.

($\Rightarrow$) Suppose that c can be made a winner by adding candidate a. It follows that b has to lose at least K points. Hence, a has to be added in the votes of the form $d_j > b > \overline{C}$ at first or second position. Thus, a gets each point that b loses. But since c is made a winner by inserting a, the new candidate a can get no more than K points. Therefore, we have to insert a in a subset of votes such that the weights of these votes sum up to exactly K. Consequently, there exists a partition.

Since PARTITION is NP-complete, this proves NP-hardness. Membership in NP is straightforward. Thus PCWNA is NP-complete for 2-approval. ❑

It is easy to see that the proof of Theorem 3.3 can be transferred to k-approval: In each vote $k-2$ dummy candidates are added in the first $k-2$ positions, which gives a total number of $(k-1)(n+1)+2$ initial candidates and one new candidate. Thus we can state the following corollary.

COROLLARY 3.4. PCWNA *is* NP*-complete for* k*-approval in the case of weighted voters where the number of candidates is unbounded and one new candidate is to be added.*

Note that, in Corollary 3.4, the k in k-approval cannot depend on the number of candidates, since the proof is for an *unbounded* number of candidates. Table 5 summarizes the results of this section.

Scoring rule	PCWNA
Plurality, $\|C\| = 2$, $\|C'\| = 1$	NP-complete
k-Approval, $\|C'\| = 1$	NP-complete

Table 5: New results on the complexity of PcWNA in the case of weighted voters.

4. UNCERTAINTY ABOUT THE VOTING SYSTEM

4.1 Scoring Rules

In this section we study the POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM problem with respect to the class of scoring rules. Recall that c is the distinguished candidate we want to make a winner in the given m-candidate election, by specifying the values α_i of the scoring vector $(\alpha_1, \ldots, \alpha_m)$ appropriately. In the proof of Theorem 4.3 below we will need the following notions.

DEFINITION 4.1. *For an election* $E = (C, V)$*, let* $pos_i(x)$ *denote the total number of times candidate* $x \in C$ *is at position* i*,* $1 \leq i \leq |C|$*, in the list* V *of votes, and for all* $a \in C \setminus \{c\}$*, let* $plus_{(c,i)}(a) = pos_i(a) - pos_i(c)$*.*

If the election is held under scoring vector $(\alpha_1, \ldots, \alpha_m)$, candidate c wins if and only if for each $a \in C \setminus \{c\}$, we have $\sum_{i=1}^{|C|} plus_{(c,i)}(a) \cdot \alpha_i \leq 0$ in the co-winner case. For the unique-winner case, replace the zero on the right-hand side of the inequality by one.

In the following lemma we will show how to construct a list of votes for given values $plus_{(c,i)}(a)$ under some conditions. Let $M_{(d,i)}$ denote a circular block of $|C|-1$ votes, where candidate d is always at position i and all other candidates take all the remaining positions exactly once, by shifting them in a circular way. For example, for the set $C = \{d, c_1, \ldots, c_m\}$ of candidates the circular block $M_{(d,1)}$ looks as follows:

$$\begin{array}{ccccccccccc}
d & > & c_1 & > & c_2 & > & \ldots & > & c_{m-1} & > & c_m \\
d & > & c_2 & > & c_3 & > & \ldots & > & c_m & > & c_1 \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
d & > & c_m & > & c_1 & > & \ldots & > & c_{m-2} & > & c_{m-1}
\end{array}$$

LEMMA 4.2. *Let* C *be a set of* m *candidates,* $c \in C$ *be a distinguished candidate,* $d \in C$ *be a dummy candidate, and let the values* $plus_{(c,i)}(a) \in \mathbb{Z}$*,* $1 \leq i \leq m-1$*, for all candidates* a *in* $C \setminus \{c, d\}$ *be given. Let* $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ *be an arbitrary scoring vector with* $\alpha_m = 0$*. One can construct in time polynomial in* m *a list* V *of votes satisfying that:*

1. *Every candidate* $a \in C \setminus \{c, d\}$ *has the given values* $plus_{(c,i)}(a)$*,* $1 \leq i \leq m-1$*, in election* (C, V)*, and*
2. *candidate* d *cannot beat candidate* c *in election* (C, V)*.*

PROOF. Let $m = |C|$ be the number of candidates. For each positive value $plus_{(c,i)}(a)$, $1 \le i \le m-q$, $a \in C \setminus \{c,d\}$, we construct two types of circular blocks of votes. The first block is of type $M_{(d,i)}$, except that in the vote in which candidate a is at position m, the positions of a and d are swapped. For this block it holds that $plus_{(c,i)}(a) = 1$, and all other values $plus_{(c,j)}(b)$ and $plus_{(c,j)}(a)$, $b \in C \setminus \{c,d,a\}$, $1 \le j \le m-1$, remain unchanged. These blocks will be added with multiplicity $plus_{(c,i)}(a)$. To ensure that candidate d has no chance to beat candidate c, we add the votes of the circular block $M_{(d,m)}$ with multiplicity $m \cdot plus_{(c,i)}(a)$. Clearly, this block does not affect the values $plus_{(c,j)}(b)$, $1 \le j \le m-1$, $b \in C \setminus \{c,d\}$.

If $plus_{(c,i)}(a)$ is negative, we add the block of type $M_{(d,m)}$, where the places of a and d are swapped in the vote in which a is at position i, with multiplicity $-plus_{(c,i)}(a)$. The effect is that $plus_{(c,i)}(a)$ is decreased by 1 for each of these blocks. Again, to ensure that candidate d will not be able to beat candidate c, we add the circular block $M_{(d,m)}$ with multiplicity $-plus_{(c,i)}(a)+1$.

By construction, the values $plus_{(c,i)}(d)$, $1 \le i \le n$, are never positive, so obviously d has no chance to beat or to tie with c in the election whatever scoring rule will be used. Since the votes can be stored as a list of binary integers representing their corresponding multiplicities, these votes can be constructed in time polynomial in m. ❑

To make use of Lemma 4.2, we assume succinct representation of the election (see [16]) in the following theorem. As mentioned in the above proof, this means that the votes are not stored ballot by ballot for all voters, but as a list of binary integers giving their corresponding multiplicities.

THEOREM 4.3. *Let $\mathcal{S}$ be the class of scoring rules with $m \ge 4$ candidates that are defined by a scoring vector of the form $\alpha = (\alpha_1, \ldots, \alpha_{m-4}, x_1, x_2, x_3, 0)$, with $x_i = 1$ for at least one $i \in \{1,2,3\}$. $\mathcal{S}$-*PCWUVS *and $\mathcal{S}$-*PWUVS *are* NP*-complete (assuming succinct representation).*

PROOF. Membership in NP is obvious, and the proof of NP-hardness will be by a reduction from the NP-complete problem INTEGER KNAPSACK (see, e.g., [17]):

Name: INTEGER KNAPSACK
Instance: A finite set of elements $U = \{u_1, \ldots, u_n\}$, two mappings $s, v : U \to \mathbb{Z}^+$, and two positive integers, b and k.
Question: Is there a mapping $c : U \to \mathbb{Z}^+$ such that

$$\sum_{i=1}^{n} c(u_i)s(u_i) \le b \text{ and } \sum_{i=1}^{n} c(u_i)v(u_i) \ge k\,?$$

We first focus on the co-winner case and then show how to transfer the proof to the unique-winner case. Let (U, s, v, b, k) be an instance of INTEGER KNAPSACK with $U = \{u_1, \ldots u_n\}$ and let $c : U \to \mathbb{Z}^+$ be a mapping. Then it holds that

$$\begin{aligned} &\sum_{i=1}^{n} c(u_i) \cdot s(u_i) \le b \\ &\sum_{i=1}^{n} c(u_i) \cdot v(u_i) \ge k \end{aligned} \tag{2}$$

$$\Leftrightarrow \quad \begin{pmatrix} s(u_1) & s(u_2) & \ldots & s(u_n) \\ -v(u_1) & -v(u_2) & \ldots & -v(u_n) \end{pmatrix} \begin{pmatrix} c(u_1) \\ c(u_2) \\ \vdots \\ c(u_n) \end{pmatrix} \le \begin{pmatrix} b \\ -k \end{pmatrix}$$

$$\Leftrightarrow \quad \begin{pmatrix} & & -b' \\ & & k' \\ & & nb \\ A & & (n-1)b \\ & & \vdots \\ & & b \end{pmatrix} \begin{pmatrix} c'(u_1) \\ c'(u_2) \\ \vdots \\ c'(u_n) \\ 1 \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3}$$

$$\text{with } A = \begin{pmatrix} s(u_1) & s(u_2) & \ldots & s(u_n) \\ -v(u_1) & -v(u_2) & \ldots & -v(u_n) \\ -1 & 0 & \ldots & 0 \\ 0 & -1 & \ldots & 0 \\ \vdots & & & \\ 0 & \ldots & 0 & -1 \end{pmatrix}, \text{ where}$$

$$\begin{aligned} c'(u_i) &= c(u_i) + (n-i+1)b, \quad 1 \le i \le n, \\ b' &= b + \sum_{i=1}^{n} b \cdot s(u_i) \cdot (n-i+1), \text{ and} \\ k' &= k + \sum_{i=1}^{n} k \cdot v(u_i) \cdot (n-i+1). \end{aligned}$$

The last n rows of the matrix ensure that

$$c'(u_i) \ge (n-i+1)b, \quad 1 \le i \le n,$$

and so there are no new solutions added for which the values $c(u_i)$ may be negative. Furthermore, since $c(u_i) \le b$, it is now ensured that $c'(u_1) \ge c'(u_2) \ge \cdots \ge c(u_n) \ge b$. Hence it still holds that c is a solution for the given INTEGER KNAPSACK instance if and only if c' is a solution for (3).

We will now build an election $E = (C, V)$ with candidate set $C = \{c, d, e, f, g_1, \ldots, g_n\}$, where c is the distinguished candidate and d is a dummy candidate who cannot beat c in the election whatever scoring rule will be used. The list of votes will be built using Lemma 4.2 according to the matrix in (3). The $n+2$ rows in the matrix correspond to the candidates e, f, and $g_1, \ldots, g_n$. Since the matrix has only $n+1$ columns, the positions $n+2$ and $n+3$ in the votes will have no effect on the outcome of the election, and thus the corresponding $plus_{(c,i)}(a)$ values, $n+2 \le i \le n+3$, can be set to zero for all candidates $a \in \{e, f, g_1, \ldots, g_n\}$. The corresponding values in the scoring vector can be set to either zero or one, respecting the conditions for a valid scoring vector. Hence, the votes in V have to fulfill the following properties:

$$plus_{(c,i)}(e) = \begin{cases} s(u_i) & \text{for } 1 \le i \le n \\ -b' & \text{for } i = n+1 \\ 0 & \text{for } n+2 \le i \le n+3, \end{cases}$$

$$plus_{(c,i)}(f) = \begin{cases} -v(u_i) & \text{for } 1 \le i \le n \\ k' & \text{for } i = n+1 \\ 0 & \text{for } n+2 \le i \le n = n+3, \end{cases}$$

$$plus_{(c,i)}(g_j) = \begin{cases} -1 & \text{for } 1 \le i \le n, i = j \\ (n-i+1)b & \text{for } i = n+1, 1 \le j \le n \\ 0 & \text{for } 1 \le i \le n+3, \\ & \quad 1 \le j \le n,\ i \ne j. \end{cases}$$

According to Lemma 4.2, these votes can be constructed in polynomial time such that the dummy candidate d has no influence on c being a winner of the election, whatever scoring rule of type $\alpha = (\alpha_1, \ldots, \alpha_n, 1, \alpha_{n+2}, \alpha_{n+3}, 0)$ will be used.

Since the $plus_{(c,i)}(a)$ values assigned to the candidates $a \in C \setminus \{c, d\}$ are set according to the matrix in (3), it holds that c can be a winner in election $E = (C, V)$ by choosing a scoring rule of the form $\alpha = (\alpha_1, \ldots, \alpha_n, 1, \alpha_{n+2}, \alpha_{n+3}, 0)$ if and only if for each $a \in C \setminus \{c\}$, we have

$$\sum_{i=1}^{n} plus_{(c,i)}(a) \cdot c(u_i) + plus_{(c,n+1)}(a) \le 0.$$

As described above, the values in the scoring vector for positions $n+2$ and $n+3$, have no effect on the outcome of the election. Hence, by switching rows in the matrix we can extend the set of possible scoring rules to scoring rules of the form $\alpha = (c(u_1), \ldots, c(u_n), x_1, x_2, x_3, 0)$, with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$. Hence, c can be made a winner of the election $E = (C, V)$ if and only if there is a solution to (3). Since we have shown above that there is a solution to (2) if and only if there is a solution to (3), it holds that there is a solution c to our Integer Knapsack instance if and only if there is a scoring rule α, of the form described above, under which c wins the election $E = (C, V)$.

To see that this reduction also settles the unique-winner case, note that (3) is equivalent to the following inequality:

$$\begin{pmatrix} & -b'+1 \\ & k'+1 \\ & nb+1 \\ A & (n-1)b+1 \\ & \vdots \\ & b+1 \end{pmatrix} \begin{pmatrix} c'(u_1) \\ c'(u_2) \\ \vdots \\ c'(u_n) \\ 1 \end{pmatrix} \le \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \qquad (4)$$

The election we need to construct has the same candidate set as above and the voters are constructed according to the values $plus_{(c,n+1)}(a)$ for $a \in C \setminus \{c, d\}$ in the matrix of (4). Thus, c is the unique winner of the modified election if and only if for each $a \in C \setminus \{c\}$, we have

$$\sum_{i=1}^{n} plus_{(c,i)}(a) \cdot c(u_i) + plus_{(c,n+1)}(a) \le 1.$$

By a similar argument as above, there is a scoring rule of the form $\alpha = (\alpha_1, \ldots, \alpha_n, x_1, x_2, x_3, 0)$ with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$ in which c wins the election if and only if there is a solution c for the given Integer Knapsack instance. ❑

4.2 Copeland$^\alpha$ Elections

In Copeland$^\alpha$ elections [14], the parameter α is a rational number from the interval $[0, 1]$ that specifies how ties are rewarded in the pairwise comparisons between candidates.

Theorem 4.4. *$\mathcal{C}$-PcWUVS and $\mathcal{C}$-PWUVS are polynomial-time solvable for the family of Copeland$^\alpha$ elections:*

$$\mathcal{C} = \{Copeland^\alpha \mid \alpha \text{ is a rational number in } [0, 1]\}.$$

Proof. To decide whether a distinguished candidate c can be made a winner of the election by choosing the parameter α after all the votes have been cast, we do the following. In the co-winner case, for each $c_i \in C \setminus \{c\}$, compute

$$f(c_i) = \begin{cases} \frac{win(c) - win(c_i)}{tie(c) - tie(c_i)} & \text{if } tie(c) \ne tie(c_i) \\ win(c) - win(c_i) & \text{otherwise.} \end{cases}$$

If $f(c_i) \ge 0$ for all $c_i \in C$, c can be made a winner of the election by setting $\alpha = \min_{c_i \in C}\{f(c_i), 1\}$, and otherwise c cannot be made a winner. So $\mathcal{C}$-PcWUVS is in P.

In the unique-winner case, for c to be the unique winner winner of the election, it must hold that $f(c_i) > 0$ and α is set to a value greater than $\min_{c_i \in C}\{f(c_i)\}$ if this value is less than one, or else to one. Otherwise, c cannot be made the unique winner of the election. So $\mathcal{C}$-PWUVS is in P. ❑

4.3 Preference-Based Approval Voting

In approval voting the situation is a bit different, since approval voting is not a class of voting systems, and the voters usually do not report linear preferences but approval vectors. Brams and Sanver [7, 8] proposed various voting systems that combine preference-based voting and approval voting. Here the voters report a strict preference order, along with an approval line indicating that the voter approves of all candidates to the left of this line and disapproves of all candidates to the right of this line. They require votes to be *admissible* [7], which means that each voter approves of his or her first ranked candidate and disapproves of his or her last ranked candidate. If we assume that the approval lines are not set by the voters (who thus only report their linear orders) but are set by the voting system itself (after all votes have been cast), we obtain (for m candidates and n voters) a class $\mathcal{A}_{m,n}$ of $(m-1)^n$ voting systems. For each such system, the candidates with the highest number of approvals win. Note that these voting systems are not very natural (as they do not let the voters themselves choose their approval strategies) and do not possess generally desirable social-choice properties (e.g., the systems in $\mathcal{A}_{m,n}$ are not even anonymous, as changing the order of votes may result in a different outcome).

In this setting, given an election where voters report their preference orders, setting the approval lines afterwards corresponds to choosing a system from $\mathcal{A}_{m,n}$. It is easy to see that PcWUVS and PWUVS are polynomial-time solvable for this class. To make the distinguished candidate c win the election, choose the system that sets the approval line in each vote that does not rank c at the last position right behind c, and in the votes that do rank c last right behind the top candidate. If c is not a winner (unique winner) of this election, c cannot win (be a unique winner of) the election whatever system from the class is chosen. Thus, PcWUVS and PWUVS are polynomial-time solvable for this class of preference-based approval voting systems.

In contrast to this result, Elkind et al. [12] show NP-hardness for a related bribery problem, even if the briber is only allowed to move the approval line.

5. CONCLUSIONS AND FUTURE WORK

For the Possible Winner problem, a full dichotomy result for the class of pure scoring rules is known [6, 5]. In contrast, the complexity of the related problem PcWNA has not yet been completely settled and the question raised by Chevaleyre et al. [9] remains open. Our result stated in Theorem 3.1 makes a further step towards this goal by showing NP-completeness of PcWNA for a whole class of pure scoring rules. An interesting task for future work would be to characterize this problem for all pure scoring rules in terms of a dichotomy result. Moreover, our initial work on weighted voters for PcWNA might be extended, and for both the weighted and the unweighted case the unique-winner variant PWNA should be further explored (see also [9, 25]). Another problem also stated in [9] concerns the number of new candidates to be added. Up to now NP-hardness results for pure scoring rules are known only for the case where one new candidate is added. What about adding more than one candidate? Note that the problem becomes easy if an unbounded number of new candidates is to be added.

For the PcWUVS and PWUVS problems, the next obvious step would be to extend Theorem 4.3 to unrestricted scoring rules, ideally with the goal of obtaining a complete dichotomy result. It would also be interesting to study these problems for other natural classes of voting systems, for example, for all voting systems sharing some important social-choice property (e.g., for all Condorcet systems).

Acknowledgment

We thank the reviewers for the very helpful comments. This work was supported in part by DFG grants RO 1202/{11-1, 12-1} and the ESF EUROCORES program LogICCC.

6. REFERENCES

[1] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[2] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.

[4] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editors, *Handbook on Approval Voting*, chapter 10, pages 199–251. Springer, 2010.

[5] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 1019–1020. IOS Press, Aug. 2010. Short paper.

[6] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science*, pages 124–136. Springer-Verlag *Lecture Notes in Computer Science #5734*, Aug. 2009.

[7] S. Brams and R. Sanver. Critical strategies under approval voting: Who gets ruled in and ruled out. *Electoral Studies*, 25(2):287–305, 2006.

[8] S. Brams and R. Sanver. Voting systems that combine approval and preference. In S. Brams, W. Gehrlein, and F. Roberts, editors, *The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn*, pages 215–237. Springer, 2009.

[9] Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible winners when new candidates are added: The case of scoring rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 762–767. AAAI Press, July 2010.

[10] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. Technical Report Cahiers du LAMSADE 302, Université Paris-Dauphine, 2010.

[11] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[12] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. Technical Report cs.GT/0905.3885, ACM Computing Research Repository (CoRR), May 2009.

[13] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 768–773. AAAI Press, July 2010.

[14] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[15] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pages 375–406. Springer, 2009.

[16] P. Faliszewski, L. Hemaspaandra, and E. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[17] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[18] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[19] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[20] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[21] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[22] M. Pini, F. Rossi, K. Venable, and T. Walsh. Dealing with incomplete agents' preferences and an uncertain agenda in group decision making via sequential majority voting. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning*, pages 571–578. AAAI Press, Sept. 2008.

[23] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[24] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 3–8. AAAI Press, July 2007.

[25] L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: New results coming up! In V. Conitzer and J. Rothe, editors, *Proceedings of the 3rd International Workshop on Computational Social Choice*, pages 199–210. Universität Düsseldorf, Sept. 2010.

Trust and Organisational Structure

Trust as Dependence: A Logical Approach

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA

singh@ncsu.edu

ABSTRACT

We propose that the trust an agent places in another agent declaratively captures an architectural *connector* between the two agents. We formulate trust as a *generic* modality expressing a relationship between a truster and a trustee. Specifically, trust here is *definitionally independent* of, albeit constrained by, other relevant modalities such as commitments and beliefs. Trust applies to a variety of attributes of the relationship between truster and trustee. For example, an agent may trust someone to possess an important capability, exercise good judgment, or to intend to help it. Although such varieties of trust are hugely different, they respect common logical patterns. We present a logic of trust that expresses such patterns as reasoning postulates concerning the static representation of trust, its dynamics, and its relationships with teamwork and other agent interactions. In this manner, the proposed logic illustrates the general properties of trust that reflect natural intuitions, and can facilitate the engineering of multiagent systems.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*; D.2.1 [**Software Engineering**]: Requirements Specifications—*Methodologies*

General Terms

Theory

Keywords

Trust, commitments, service-oriented computing

1. INTRODUCTION

We develop a novel approach to trust in multiagent systems that relates the intuition of trust as reliance with the notion of an architectural *connector* [17]. When the components of a software architecture are agents (understood as active, autonomous entities), each connector between any two agents is naturally understood in terms of the trust they place in each other. In this manner, we not only relate intuitions about two heretofore isolated subfields of multiagent systems (trust and agent-based software engineering), but also provide a new basis for formalizing those intuitions to use as a basis for improved engineering methodologies.

Classically, following Castelfranchi and Falcone [1], one may understand an agent (the *truster*) as trusting another (the *trustee*) when the truster puts its plans in the hands of the trusted agent. In general terms, the above is a valuable intuition that we seek to preserve. However, Castelfranchi and Falcone take a staunchly cognitive stance wherein a "plan" is reflected in the intentions and beliefs of the truster with respect to the trustee.

In contrast, we take the position that the notion of "plan" in general multiagent settings is often, though not always, far removed from the cognitive view. Referrals, which are crucial for inducing trust in social settings, often involve plans that might be quite tenuous. In other cases, one may spot a plan only based on strong assumptions about the truster and trustee, the tasks involved, and the context. Therefore, we advocate here an architectural intuition where the parties may not have strongly cognitive plans either.

Trust arises in many settings. For this reason, we develop a modular, "minimalist" formalization of trust, which captures the essential properties that any model of trust would follow. Our approach does not demand agreement on the additional aspects of trust—such as belief, intentions, plans, similarity, probability, utility—that specific models might incorporate and specific applications may demand. Thus our approach can provide a conceptual basis for organizing systems without having to delve into the details of trust.

We treat trust as a high-level architectural connector. A truster's trust in a trustee expresses the expectations the truster holds of the trustee. This interpretation of an architectural connector as the dependence of a truster on a trustee generalizes the classical software architecture [15] idea of one component's "assumptions" about another. Traditionally, such assumptions reduce to operational details of control and data flow, but in agent-oriented software engineering we ought to treat them as interagent dependencies.

Singh and Chopra [19] propose to use commitments as a basis for multiagent systems architecture. Commitments are appropriate bases for interaction where a protocol specifies the commitments involved. However, in flexible, emergent settings, such specifications might be incomplete or even nonexistent. That is, the agents should be prepared to interact with others even in the absence of commitments. In such cases, the basis for their interactions would be the trust that each agent places in the other. Even when a commitment exists, the creditor of the commitment would need to trust

Cite as: Trust as Dependence: A Logical Approach, Munindar P. Singh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 863-870.

the debtor in order to rationally act on the assumption that the debtor will discharge the commitment in consideration.

When we apply our trust-based approach on traditional software components, any modeling of trust would be implicit and hard-coded in the components—reflected only in the minds of the designers. When we apply our approach on sophisticated intelligent agents, the modeling of trust would be more explicit and subject to reasoning by the agents themselves.

Notice that sometimes architecture is conflated with notations for expressing it, especially established notations such as UML. Such notations have no abstractions geared toward trust and other high-level concepts, so we avoid them here.

Contributions. We present a formal semantics of trust, motivating several reasoning postulates for trust and relating those postulates to architectural connectors. Our contributions bear relevance also to the study of commitments, which we treat as correlates of trust. Further, the notion of architecture pursued here, although far removed from traditional software architecture, is inspired by taking a truly agent-oriented stance. Not only are agents a natural abstraction but also the trust between them is core to their interactions.

Organization. The rest of this paper is organized as follows. Section 2 discusses some intuitions about trust as it relates to architecture. Section 3 introduces our technical framework for trust. Section 4 presents a variety of postulates for trust describing potential properties of relevance to active trust, integrity, structure, meaning, teamwork, and dynamics. Section 5 presents a case study demonstrating our approach in relation to both traditional and more recent commitment-based approaches. Section 6 places our work in the broader setting of architecture and brings out some directions for future work.

2. INTUITIONS ABOUT TRUST

Trust is central to several disciplines. So it is not surprising that it has garnered a lot of research attention. Existing approaches differ a lot on the complexity of the conceptual model in which they consider trust. The following main lines of research reflect the intuition of dependence are relevant.

Subjective, which treat trust as a suitably structured set of beliefs and intentions [1]. Indeed, Demolombe [5] reduces trust to (graded) beliefs. Liau [12] and Dastani et al. [4] consider how a truster may absorb information from a trustee, e.g., by adopting a belief if a trusted sender says so.

Measured, which treat trust as a numeric weight based on heuristics [9], subjective probability [11, 23], a utility [3], or a grade [5]. These are subjective approaches albeit with representations geared toward numeric or ordinal values.

Social, which understand trust in terms of social relationships [20]. Falcone and Castelfranchi [7] distinguish objective and subjective dependence as well as unilateral, reciprocal, and mutual dependence. Our basic framework accounts for all of these, albeit with specific postulates describing different situations, e.g., teamwork. Johnson et al. [10] examine teamwork via social interdependence, which is crucial as a basis for trust.

A commonality of the existing approaches is that they conflate aspects of the *representation* of trust on the one hand with the complex of features that go into making a *judgment* of trustworthiness on the other. The latter involve reasoning techniques (often domain-specific and heuristic) for updating the extent of trust placed by a truster in a trustee. Indeed, there is a common confusion when talking about trust in that many researchers expect to see the above kinds of heuristics, and do not appreciate the value of a generic method, such as ours. As an analogy, one can think of rules of Bayesian inference or axioms of belief. Such rules and axioms do not in themselves produce an answer of what an agent should infer or believe, but constrain the probabilistic or binary truth values an agent may assign to various propositions. In the same way, our approach describes how an agent or a designer may reason soundly about trust.

We formalize a general-purpose semantically motivated representation of trust. Interestingly, this representation provides a basis for stating a variety of constraints on the modeling of trust with respect to the integrity and structure of architectural connectors, and of reasoning about trust. Although it is not focused on trust measures, it also provides a basis for such measures.

Conditionality of Trust. We posit that, in general, trust must be conditional. Each assignment of trust presupposes some preconditions (which we can capture as antecedents) and expectations (which we can capture as consequents). Blind trust is merely a boundary condition. This holds in normal usage: e.g., a customer may trust a merchant as follows "if I pay, (I trust) the merchant will deliver the goods," expressing the customer's expectation and presumably linking it to further plans of the customer.

Trust as Dependence, Architecturally. Let us consider an agent formulating and enacting a plan that relies upon the contributions of others—in essence, trusting the others to make their contributions to its plan. More generally, the interactions of an agent with other agents may be described at a high level in terms of the trust each of them places upon the others. We model further aspects of the interactions such as whether trusted agents are indeed trustworthy based on how the trust maps to relevant concepts.

Although the antecedent and consequent are generic, nominally, we associate them with the truster and trustee, respectively. When the antecedent becomes true, the connector *activates* and when the consequent becomes true, the connector *completes*. It is helpful to relate the antecedent and consequent of a trust expression to the structure of the connector it describes. Intuitively, a trust expression becomes stronger as its antecedent becomes weaker and its consequent becomes stronger. We can understand the antecedent becoming weaker with the connector becoming *broader* because it would activate more easily. Likewise, we can understand the consequent becoming stronger with the connector becoming *tighter* because it would complete with greater effort on part of the trustee, and thus sustain enhanced expectations on part of the truster.

This paper develops an organizational approach, especially from the standpoint of the connectors among autonomous agents understood conceptually. As explained above, in this view, a relationship from one agent to another can be understood as the trust the first agent places in the second agent.

An agent may implement such an interconnection based on concepts such as beliefs.

3. TECHNICAL FRAMEWORK

Our technical framework is based on modal logic with a possible worlds semantics. In addition to trust, we capture commitments as an abstraction because they help us state various important postulates reflecting dependence.

We include an explicit notion of *reality* in our model. That is, we identify a path (corresponding to a particular execution of the multiagent system) as being the real one. This is *not* to suggest that we have found a way to predict the future; rather, it is a way to accommodate nondeterminism by merely claiming (as appropriate) that whatever the real path might be, it satisfies some property, desirable or otherwise. For example, we might define trust as being well-placed if the proposition that is being trusted occurs on the real path. In this manner, incorporating reality explicitly enables us to state constraints that we cannot state otherwise.

3.1 Syntax and Formal Model

Putting together the intuitions about architectural connectors and the inherent conditionality of trust, we propose to formalize trust-as-dependence as a modal operator that takes two parties and two propositions, as in

$$\mathsf{T}_{\mathsf{truster,trustee}}(\mathsf{antecedent}, \mathsf{consequent})$$

The first two arguments describe the end points of the given connector, and the last two its logical structure. In logical terms, trust bears a syntactic similarity with commitments but the two are independent concepts. More generally, we can view trust and commitment as correlates of each other. Some of the postulates below relate trust and commitments.

$\mathcal{L}$, our formal language, takes a linear-time logic enhanced with a modality C for commitments [18] with a modality T for trust. Below, *Atom* is a set of atomic propositions and $\mathcal{X}$ is a set of agent names. We further define agents that are composed from other agents; in other words, an agent may be a simplistic multiagent system. L and X are nonterminals corresponding to $\mathcal{L}$ and $\mathcal{X}$, respectively.

L_1. $L \longrightarrow \mathit{Trust} \mid \mathit{Commit} \mid \mathit{Atom} \mid L \wedge L \mid \neg L \mid \mathsf{R}L \mid L\,\mathsf{U}L$

L_2. $\mathit{Trust} \longrightarrow \mathsf{T}_{\mathit{Agent},\mathit{Agent}}(L, L)$

L_3. $\mathit{Commit} \longrightarrow \mathsf{C}_{\mathit{Agent},\mathit{Agent}}(L, L)$

L_4. $\mathit{Agent} \longrightarrow X \mid \langle\{\mathit{Agent}\}\rangle$

We use the following conventions: x, etc. are agents, ψ, etc. are atomic propositions, p, q, r, etc. are formulae in $\mathcal{L}$, t, etc. are moments, and P, etc. are paths. We drop agent subscripts when they can be understood. A model for $\mathcal{L}$ is a tuple, $M = \langle \mathbb{S}, <, \mathbb{R}, \mathbb{I}, \mathbb{T}, \mathbb{C} \rangle$:

- $\mathbb{S}$ is a set of possible moments, each a possible snapshot (i.e., a state) of the world.
- $< \subseteq \mathbb{S} \times \mathbb{S}$ is a discrete linear order on $\mathbb{S}$, which induces *paths* at each moment. A path is a contiguous set of moments beginning at a moment. Two paths are either disjoint or one is a subset of the other. $[P; t, t']$ denotes a *period* on path P from t to t'. Formally, $[P; t, t']$ is the intersection of P with the set of moments between t and t', both inclusive. $\mathbb{P}$ is the set of all periods and $\mathbb{P}_t$ of periods that begin at t ($\mathbb{P}_t \neq \emptyset$).
- $\mathbb{R}$ identifies the *real path* that initiates from a moment. A real path must be self-consistent in that if a moment initiates a real path τ, every subsequent moment that occurs on path τ initiates a suffix of τ as its real path.
- The interpretation, $\mathbb{I}$, of an atomic proposition is the set of moments at which it is true. That is, $\mathbb{I} : \mathit{Atom} \mapsto \wp(\mathbb{S})$. We show below, through the definition of moment-intension (which lifts $\mathbb{I}$ to all propositions), that the denotations of all propositions are sets of moments.
- At each moment, $\mathbb{T} : \mathbb{S} \times \mathcal{X} \times \mathcal{X} \times \wp(\mathbb{S}) \mapsto \wp(\wp(\mathbb{P}))$ yields a set of periods for each moment and proposition for each truster-trustee (ordered) pair of agents.
- At each moment, $\mathbb{C} : \mathbb{S} \times \mathcal{X} \times \mathcal{X} \times \wp(\mathbb{S}) \mapsto \wp(\wp(\mathbb{P}))$ yields a set of periods for each moment and proposition for each debtor-creditor (ordered) pair of agents.

Models for modal logics are commonly based on Kripke structures, which define a set of possible worlds along with an accessibility relation that maps each world to a set of worlds. The semantics of a modal operator tests for *inclusion* in that set of worlds. The models proposed here are *not* Kripke structures and do not involve an accessibility relation. Instead they are based on the Montague (and Scott) approach [14] to define a "standard" of correctness by mapping each world to a set of sets of worlds. The semantics of a modal operator tests for *membership* in the set of sets of worlds. Montague's approach offers greater flexibility in allowing or denying some inferences that the Kripke approach requires. In many (though not all) cases, it is straightforward to map this semantics to a Kripke semantics but we find the proposed formulation more natural and modular.

$\mathbb{T}$ and $\mathbb{C}$ capture the standards for trust and commitments, respectively, for each moment and truster-trustee pair. Given an antecedent proposition, $\mathbb{T}$ yields a set, each of whose members is a set of periods. Each set of periods is the representation in the model of a consequent proposition, specifically, the proposition whose period-intension (defined below as the set of periods at whose culmination it holds) equals that set of periods. The truster trusts the trustee to bring about any such consequent if the antecedent holds. Likewise, $\mathbb{C}$ yields a set each of whose members is a set of periods, each culminating in the consequent proposition that the debtor commits to bringing about. As in many (arguably most) logics of intention and obligation, we do not model actions explicitly: $\mathbb{T}$ and $\mathbb{C}$ are simply understood as describing the conditions an agent would bring about.

3.2 Semantics

The semantics of $\mathcal{L}$ is given relative to a model, a path, and a moment on the path. $M \models_{P,t} p$ expresses "M satisfies p at t on path P." The truth of several constructs is independent of the path and depends only on the moment. An expression p is *satisfiable* (respectively, *valid*) iff for some (respectively, all) M, P, and $t \in P$, $M \models_{P,t} p$. Formally, we have:

M_1. $M \models_{P,t} \psi$ iff $t \in \mathbb{I}(\psi)$, where $\psi \in \mathit{Atom}$

M_2. $M \models_{P,t} p \wedge q$ iff $M \models_{P,t} p$ and $M \models_{P,t} q$

M_3. $M \models_{P,t} \neg p$ iff $M \not\models_{P,t} p$

M_4. $M \models_{P,t} \mathsf{R}p$ iff $M \models_{\mathbb{R}_t,t} p$

M$_5$. $M \models_{P,t} p\mathsf{U}q$ iff $(\exists t'' \in P : t \leq t''$ and $M \models_{P,t''} q$ and $(\forall t' : t \leq t' < t'' \Rightarrow M \models_{P,t'} p))$

Disjunction ($\vee$), implication ($\rightarrow$), equivalence ($\equiv$), false, and true are the usual abbreviations. $p\mathsf{U}q$ means "p holds until q": thus $\mathsf{true}\mathsf{U}q$ (abbreviated $\mathsf{F}q$) means "eventually q." And, $\mathsf{R}p$ means that p holds on the real path of the current moment.

We define the *moment-intension* of formula p as the set of moments where it is true: $[\![p]\!] = \{t | M \models_{P,t} p\}$. We define *period-intension* of formula p as the set of periods culminating in its becoming true: $\langle\!\langle p \rangle\!\rangle = \{[P; t, t'] | M \models_{P,t'} p\}$. In these periods, p occurs at the last moment but may possibly occur earlier as well. Thus these are all possible ways in which p may be brought about. Based on these, we can now specify the formal semantics of trust and commitments. As explained in connection with $\mathbb{T}$ above, $\mathsf{T}_{x,y}(r, u)$ holds precisely at points where the period-intension of u belongs to the standard for trust. (Likewise, for commitments).

M$_6$. $M \models_{P,t} \mathsf{T}_{x,y}(r, u)$ iff $\langle\!\langle u \rangle\!\rangle \in \mathbb{T}_{x,y}(t, [\![r]\!])$

M$_7$. $M \models_{P,t} \mathsf{C}_{x,y}(r, u)$ iff $\langle\!\langle u \rangle\!\rangle \in \mathbb{C}_{x,y}(t, [\![r]\!])$

4. REASONING POSTULATES

Let's now consider several postulates that reflect common reasoning patterns that apply uniformly to trust. It is worth emphasizing that we consider atomic propositions that are *stable*, meaning that they include any temporal requirements within them. Thus a proposition that is true is generally true forever. For example, let *pay* mean the agent pays by noon on May 1. If *pay* is true at one point on a run, it is true on all points on the run. Consequently, most of our postulates do not involve any temporal operators. Trust and commitments (which can become active and then inactive) are themselves not stable; thus some postulates that deal with them involve the until operator. We expand the notion of agents to treat simplified multiagent systems.

4.1 Postulates for Active Trust

We treat trust in the sense of a living, functioning architectural connector. That is, we consider the case of *active* trust. When a truster places trust in a trustee, the corresponding connector is activated. When the trustee has performed as expected, there is no more for the truster to expect of the trustee based solely upon the given connector. In such a case, the connector is no longer active.

Our approach helps distinguish between a connector that is inactive and one that which has been activated but not completed. The former is perfect; the latter is worrisome. As a result, often, we would formulate trust expressions as including the possibility of success. As a specific example, an agent x may deal with an agent y because it trusts y to deliver the goods if it pays. That is, we would have $\mathsf{T}_{x,y}(\mathit{pay}, \mathit{deliver})$. But to accommodate the unknown or early performance of *deliver*, we might instead formulate the trust expression as $\mathit{deliver} \vee \mathsf{T}_{x,y}(\mathit{pay}, \mathit{deliver})$

For each postulate below that uses truster x and trustee y, for brevity, we write $\mathsf{T}(r, u)$ instead of $\mathsf{T}_{x,y}(r, u)$.

T$_1$. COMPLETE A CONNECTOR. $u \rightarrow \neg\mathsf{T}(r, u)$

When u holds, the trust in u is completed and is, therefore, no longer *active* (this treatment is neutral as to whether u is the provision of information or the performance of a domain action). Notice that the above yields $\neg\mathsf{T}(r, \mathsf{true})$ for any r.

T$_2$. ACTIVATE A CONNECTOR. $\mathsf{T}(r \wedge s, u) \wedge r \rightarrow \mathsf{T}(s, u)$

A typical case is when a truster performs part or all of what it needs to do to activate a connector. For example, if you push money over a coffee counter you trust that the barista would push back a cup of coffee for you. If you trusted the barista to give you a cup of coffee upon your paying $1, upon handing over $1 you trust the barista to give you the cup of coffee without further ado.

More generally, a connector may be activated piecemeal. When "part of" the antecedent of a connector holds, the connector strengthens to one for the "remainder" of the antecedent and with the original consequent comes into being. Notice that this postulate means that a connector does not need to be activated in a single shot: as more and more of its antecedent becomes true, the connector becomes incrementally closer to being activated. When the connector is of the form $\mathsf{T}(\mathsf{true}, u)$, then it is fully activated. For such a connector, failure by the trustee to complete the connector is tantamount to a betrayal of trust.

T$_3$. PARTITION A CONNECTOR. $\mathsf{T}(r, u \wedge v) \wedge \neg u \rightarrow \mathsf{T}(r, u)$

In general, if you trust a trustee for two propositions, you trust it for each of the propositions. In other words, you would expect to be able to partition a connector into its components. However, the obvious formulation $\mathsf{T}(r, u \wedge v) \rightarrow \mathsf{T}(r, u)$ is inconsistent with T$_1$, because if u holds, T$_1$ would eliminate $\mathsf{T}(r, u)$. Since T$_1$ is fundamental to capturing an active connector, we include $\neg u$ on the left-hand side in T$_3$. Thus a connector partitions into component connectors as long as none of the components have already been completed. For example, if you trust a merchant to send both the goods you ordered and a warranty, then you trust the merchant to send you the goods—unless the goods are already sent.

4.2 Postulates for Connector Integrity

These postulates describe the integrity of connectors.

T$_4$. AVOID CONFLICT. $\mathsf{T}(r, u) \rightarrow \neg\mathsf{T}(r, \neg u)$

A connector cannot both ask for and prevent the same thing. This postulate is stronger than merely stating that a connector for a logical impossibility cannot exist, which would be formalized as $\neg\mathsf{T}(r, \mathsf{false})$. However, in the presence of T$_8$, AVOID CONFLICT is the same as $\neg\mathsf{T}(r, \mathsf{false})$.

T$_5$. NONVACUITY. From $r \vdash u$ infer $\neg\mathsf{T}(r, u)$

Since $r \vdash u$, if r holds so does u. Or, $\mathsf{T}(r, u)$ completes as soon as it is activated, and is thus vacuous. Because $r \vdash r$, we have $\neg\mathsf{T}(r, r)$. The intuition is that a nonvacuous connector must not require an antecedent stronger than its consequent. The architectural implication of a vacuous connector is that we might as well disconnect the two agents, because the trustee would deliver no value to the truster.

T$_6$. TIGHTEN. From $\mathsf{T}(r, u), s \vdash r, s \nvdash u$ infer $\mathsf{T}(s, u)$

Any connector that holds for a weaker antecedent also holds for a stronger antecedent. In other words, we can always broaden a connector in the logical ways specified. For example, if you trust your customer will pay you $1 if you give them a coffee, then you can safely trust they will

pay you $1 if you give them a coffee and a cookie. Some useful consequences are $\mathsf{T}(r \vee s, u) \rightarrow \mathsf{T}(r, u)$, $\mathsf{T}(r, u) \rightarrow \mathsf{T}(r \wedge s, u)$, and $\mathsf{T}(\mathsf{true}, u) \rightarrow \mathsf{T}(r, u)$.

Note that $p \vdash q$ means we can prove q from p: this is stronger than implication $p \rightarrow q$, which holds merely if p is false. Clearly, $\mathsf{T}(r, u) \wedge \neg s \rightarrow \mathsf{T}(s, u)$ is bogus, i.e., we would not conclude $\mathsf{T}(s, u)$ simply because s happens to be false.

4.3 Postulates for Connector Structure

These postulates describe structural properties.

T_7. COMBINE ANTECEDENTS. $\mathsf{T}(r, u) \wedge \mathsf{T}(s, u) \rightarrow \mathsf{T}(r \vee s, u)$

To the left of the $\rightarrow$ are two connectors, together meaning that the truster expects the trustee to do u if r or if s hold, which is the connector on the right. Hence, this broadens a connector, in contrast with T_6.

T_8. COMBINE CONSEQUENTS. $\mathsf{T}(r, u) \wedge \mathsf{T}(r, v) \rightarrow \mathsf{T}(r, u \wedge v)$

Combine consequents of connectors between the same truster and trustee with the same antecedent. The truster would become committed to u and to v if r holds, which is the meaning of the connector on the right. For example, if you trust a merchant to give you an item for your payment and a warranty for the same payment, then you can expect both the item and the warranty for your payment. This postulate relies upon the propositions being not temporally indexed, as Section 4 explains.

T_9. INFERENCE CHAIN. From $\mathsf{T}(r, u), u \vdash s, \mathsf{T}(s, v)$ infer $\mathsf{T}(r, v)$

Assume you trust someone to bring about u if r and to bring about v if u. Then, you trust them to bring about v if r. T_9 generalizes the above intuition to when $u \neq s$. Here we have a situation where the connectors being chained exist between the same truster and trustee pair. The situation becomes more interesting with teamwork, as in T_{17}.

4.4 Postulates for Connector Meaning

These postulates pertain to the content of trust, especially as it relates to commitments [18]. These are important because in some respects commitments are the flip side of trust.

T_{10}. EXPOSURE. $\mathsf{C}_{x,y}(r, u) \rightarrow \mathsf{T}_{y,x}(r, u)$

A debtor is exposed when the creditor of the commitment trusts the debtor for the same content as the given commitment. Now the debtor cannot cancel the commitment without betraying the trust the creditor placed in it. This signifies architectural minimality in that a commitment is being included in a multiagent system only if there is a trust relationship that relies upon the commitment.

T_{11}. TRANSIENT ALIGNMENT. $\mathsf{T}_{x,y}(r, u) \rightarrow \mathsf{C}_{y,x}(r, u)$

A creditor and debtor of a commitment are aligned when if the creditor trusts the debtor for something, the debtor is committed to bringing it about. That is, the connector between the debtor and creditor is covered. This postulate relates to Chopra and Singh's [2] notion of commitment alignment, although their notion considers commitments alone.

T_{12}. WELL-PLACED TRUST. $\mathsf{T}_{x,y}(\mathsf{true}, u) \rightarrow \mathsf{R}u$

This says that whenever a truster trusts a trustee, the consequent comes true on the real path. The success may be incidental, but the trust is not betrayed.

T_{13}. WHOLE-HEARTED ALIGNMENT.

$\mathsf{T}_{x,y}(s, v) \rightarrow \mathsf{R}(s \rightarrow (\mathsf{C}_{y,x}(s, v) \mathsf{U} v))$

When a truster connects to a trustee, the trustee commits (as debtor) to the truster for the relevant propositions *and* remains committed until success. Thus success is achieved, but as an outcome of the debtor's persistent commitment, not incidentally. Thus, this postulate describes a stronger connector than does TRANSIENT ALIGNMENT.

The formulas below are not suitable to be asserted as constraints, but describe important situations. They could be used for problem diagnosis or in engineering effective systems.

Unexercised connector. $\mathsf{T}(r, u) \wedge \mathsf{R}\neg r$. This indicates a connector that is never activated. For example, you may trust that your banker will loan you money if you apply for one, but you may never file the requisite application.

Misplaced trust. $\mathsf{T}(r, u) \wedge \mathsf{R}\neg u$ A connector may fail because when it is activated, the trustee fails to deliver the consequent. Notice that the trustee may never have committed with respect to this connector: therefore, the trustee cannot be faulted for noncompliance.

4.5 Postulates Involving Multiple Agents

These postulates provide a basis for architecting multiagent settings such as teams. They can be thought of as specifying the structures of different types of teams in logical terms, based on the trust relationships among the members. Since, in intuitive terms, trust is an important aspect of teams, we take this to be a promising theme. Below, $\langle x, y \rangle$ represents a simplified team consisting of x and y.

T_{14}. MUTUAL PROGRESS.

$\mathsf{T}_{x,y}(r, u) \wedge \mathsf{T}_{y,x}(u, r) \rightarrow \mathsf{T}_{x,\langle x,y \rangle}(\top, r \wedge u)$

When two agents trust each other reciprocally, each of them trusts their team to make progress on both propositions. This postulate arises commonly in instances of teamwork, including successful business interactions, where each participant concedes to the other, thereby achieving progress. We can think of it as a strengthening of reciprocal dependence [7]. Trust in this sense also provides a complementary aspect to commitments in understanding concession [24].

T_{15}. TRUSTEE'S TEAM. $\mathsf{T}_{x,y}(r, u) \rightarrow \mathsf{T}_{x,\langle y,z \rangle}(r, u)$

Participation by the trustee in a team does not alter the truster's placement of trust in it. This can be thought of as describing cooperative teams in which any conflicts are resolved. For example, if z conflicted with y and prevented y from being trustworthy for u, then the above postulate would not hold for the team $\langle y, z \rangle$. In other words, the connector between the truster and trustee applies equally to the team including the trustee. For example, if you trust your local postman to deliver your mail, you can trust the local post office to deliver your mail. This inference applies when participation in the team does not alter the nature of the connection. For example, you can trust your friend to take your side in a dispute, but not against his employer.

T_{16}. TRUSTER'S TEAM. $\mathsf{T}_{x,y}(r,u) \rightarrow \mathsf{T}_{\langle x,z\rangle,y}(r,u)$

In contrast with T_{15}, here the connector applies to any team that the truster may belong to.

T_{17}. PARALLEL TEAMWORK.

$$\mathsf{T}_{x,y}(r,u) \wedge \mathsf{T}_{x,z}(u,v) \rightarrow \mathsf{T}_{x,\langle y,z\rangle}(r, u \wedge v)$$

When a truster connects to two trustees, the truster connects to their team as a composite trustee. For example, if you trust one friend to bring you bread and one to bring you soup, you trust them as a team to bring you bread and soup. This postulate is an alternative to T_9 (INFERENCE CHAIN) and shows how the connectors to two trustees can be combined.

T_{18}. PROPAGATE.

$$\text{From } \mathsf{T}_{x,y}(r,u), \mathsf{T}_{y,z}(s,v), v \vdash u, r \vdash s \text{ infer } \mathsf{T}_{x,\langle y,z\rangle}(r,v)$$

Here, x trusts y and y trusts z. Because of how the antecedents and consequents mutually relate, x trusts $\langle y,z\rangle$.

4.6 Postulates Involving Dynamism

The postulates involving updates are largely heuristic in nature. The following illustrate three aspects of dynamism: these deal with persistence when nothing changes; reduction in trust ratings when trust is betrayed; and enhancement in ratings when trust is kept. The intuition behind these is based on the notion of relational or trust capital [7], which agents can build up through trustworthy behavior and drain through untrustworthy behavior.

T_{19}. PERSISTENCE. $\mathsf{T}(r,u) \rightarrow \mathsf{T}(r,u)\mathsf{U}(u \vee r)$

A truster persists in its connector unless it acquires evidence that the connector has failed or completed. That is, a connector persists at the same strength as long as the connector is not activated (until r holds), meaning that the substantive aspect of the trust has not been exercised, or the connector has not been completed (until u holds). Assume you trust a merchant to deliver if you pay, i.e., as $\mathsf{T}(\mathsf{pay}, \mathsf{deliver})$. If you have not paid, then your not receiving a delivery should not affect your trust in the trustee.

Notice that the above postulate is silent about success or failure. Below, skepticism and faith identify domain-specific notions, outside our language, of how a truster respectively reduces or increases its level of trust in a trustee.

T_{20}. SKEPTICISM.

$$\mathsf{skepticism}_{x,y}(s,v) \rightarrow (\mathsf{T}(r,u) \wedge r \wedge \neg u) \rightarrow \neg\mathsf{T}(s,v)$$

A truster lowers its trust in a trustee if the trustee fails for an activated connector, i.e., one whose antecedent has been achieved. This can be thought as an agent narrowing or weakening its connectors with another agent based on the second agent's performance.

T_{21}. FAITH. $\mathsf{faith}_{x,y}(s,v) \rightarrow (\mathsf{T}(r,u)\mathsf{U}u) \rightarrow \mathsf{T}(s,v)$

A truster adjusts its trust in a trustee based on whether the trustee achieves the consequent. This can be thought of as an agent broadening or strengthening its connectors with another based on the second agent's performance.

In addition, we can compare trust ratings as follows.

Compare ratings. The expression $\mathsf{T}_{x,y}(r,u) \wedge \mathsf{T}_{x,w}(r, u \wedge v)$ signifies that x trusts y less than it trusts w. This reflects some intuitions of Falcone et al.'s [8] contracting approach. The deeper underlying intuition is that sets of possible paths (being different outcomes) map naturally to probabilities.

5. APPLYING THE THEORY

Let us consider a cross-organizational scenario of auto insurance claims [21], which relates naturally to multiagent systems. Figure 1 (from [21]) describes the intended operations in this scenario, which deals with auto insurance claims processing by AGFIL, an insurance company. Interestingly, this figure omits the policy holder whom the scenario serves. A policy holder, John Doe, is in an accident and files a claim with Europ Assist, who runs AGFIL's call center. Europ Assist identifies a mechanic shop (garage) in consultation with Doe, sends Doe there, and forwards his claim to AGFIL. AGFIL passes the claim to Lee Consulting Services (Lee CS), which interacts with Doe to complete the claim, obtains estimates from the mechanic, and decides whether to honor Doe's claim. Skipping ahead a few steps, this episode would normally end with the mechanic repairing Doe's car and getting paid by AGFIL.

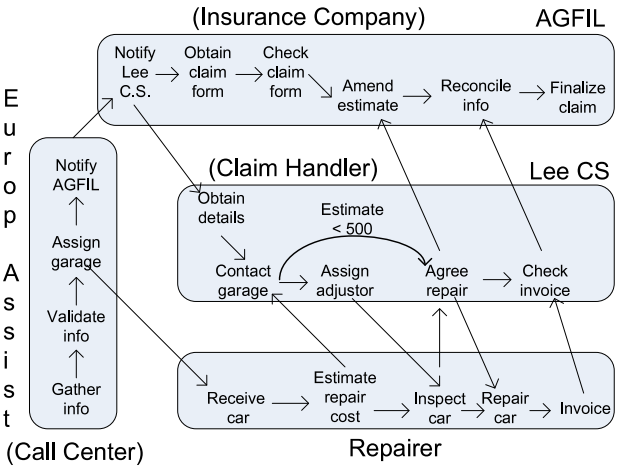


Figure 1: Insurance scenario modeled operationally

The traditional low-level representation emphasizes the steps performed by each party and their mutual control flow. It provides no support for meaning. Desai et al. [6] formalized this scenario in terms of commitments, identifying the contractual business relationships among the parties involved. However, such relationships are founded on a substrate of trust. An additional benefit of modeling trust is that it focuses on the architecture, which we can use as a basis (in an engineering methodology) for determining the necessary contractual relationships.

Let us consider the following examples. First, not only do Lee CS and AGFIL have commitments toward one another, they must also trust one another to perform accordingly. Second, the importance of trust becomes more important when we consider architectures that are not highly regimented. For example, when John Doe talks to Europ Assist, out of the many mechanics who are preapproved, Doe would select one of those whom he deemed trustworthy,

because the existence of commitments does not adequately characterize the outcomes, although the existence of a commitment by AGFIL to ameliorate a failed interaction with a preapproved mechanic may be a reason to place greater trust in the mechanic. Third, when the system in question is open, i.e., John Doe can have his car seen by any mechanic, the importance of trust goes up further.

In each of these cases, the participants would apply some of the above reasoning postulates. For example, Doe would ACTIVATE his dependence on the mechanic by bringing his car in for repairs (T_2); the mechanic would COMPLETE the dependence by repairing the car (T_1); the mechanic gives Doe a loaner car for a week: the loaner is PARTITIONED from the repair itself via (T_3); under T_7, Doe can COMBINE his dependence on the mechanic to trust the mechanic to repair the car whether Doe brings it in or asks the mechanic to tow it to his shop. Under PERSISTENCE (T_{19}), the mechanic holds his trust in being paid in a timely fashion by AGFIL until he submits a bill or gets paid. Doe and the mechanic demonstrate whole-hearted alignment (T_{13}) because the mechanic remains committed to completing the repairs until he does so. Doe applies PARALLEL TEAMWORK (T_{17}) to place his trust in the team consisting of AGFIL, Lee CS, and the mechanic to process his claim.

The foregoing points illustrate the kinds of reasoning involving trust, which can be used as criteria for judging specialized trust approaches. Existing approaches do not readily apply in the above kinds of settings: they either (1) make unrealistic assumptions about their models or (2) fail to support inferencing. In the first category we place approaches for adopting beliefs from reports [4, 12], which are simply inapplicable because trust here (and often) is about actions, not truthfulness; cognitive approaches, which presume deeper representations of beliefs and plans than may hold in practice [1, 5]; current heuristic [9] and probabilistic [11, 23] approaches, which do not provide the essential logical structure for this case (thus making it difficult to use them architecturally). In the second category, we place the social approaches to trust [7, 20] and dependence [10] which, though conceptually suited in principle to architecture, are mostly informal in their details.

More importantly, we can characterize the trust relationships among the parties with or without any contractual relationships among them. Specifically, in the above setting, we can define an auto repair ecosystem in which a party's dependencies can be expressed as trust, and reasoned about to determine if the ecosystem will prove effective: for example, if the respective dependencies are supported by capabilities or commitments of the agent's involved.

Architecture

More generally, an architecture is described not only by its components and connectors but also by its constraints and styles [17]. We propose an approach that enables specifying architectures for specific multiagent systems:

Components: Application-specific roles, such as mechanic and call center.

Connectors: The trust relationships between the roles: a connector better reflects a flow of trust not just a flow of information, as in traditional approaches. For example, the mechanic trusts AGFIL to pay for repairs.

Constraints: The reasoning postulates discussed in the foregoing. Of these, the integrity and structure constraints are of broad use; some of the others would apply in specific settings. For example, if Lee CS arranges to take care of Doe's car, the mechanic and Doe may have no direct connectors to each other.

Styles: The sets of constraints geared toward different applications. For instance, teamwork is a kind of architectural style. For example, the mechanic and policy holder may trust each other reciprocally; or the mechanic and policy holder may trust a common party, such as Lee CS or AGFIL.

One can imagine a design episode based on the above architecture. Here the designers would identify the key roles in their system-to-be, and identify the trust relationships among the (agents playing these) roles. Such trust relationships would describe the system in architectural terms. Upon further refinement, the designers could identify the commitments among the roles that would help realize the trust interactions. These could arise partly by (1) engendering trust (John Doe might trust a mechanic to complete a task after the mechanic commits to doing so) and (2) partly by yielding trust by fiat (Doe would not trust any arbitrary mechanic but a commitment from AGFIL or Lee CS to get Doe's car repaired would produce trust in an approved mechanic or limit Doe's liability and thus reduce the need for such trust). Trust as dependence can thus conceptually precede commitments. In other words, we would first identify the necessary trust relationships and then induce commitments that would support such trust. Trust is thus complementary to goal-based approaches such as Tropos, which capture dependencies between goals. Further, it can help address some of the challenges of high variability that recent work on Tropos has identified [16].

As Singh and Chopra [19] observe, recent agent-oriented software engineering approaches either follow mentalist models based on beliefs and intentions (and are thus ill-suited for multiagent architecture, since they inevitably describe an agent's internal state), or adopt low-level ideas from traditional software engineering (and are thus ill-suited for multiagent systems). Trust, as we have formalized it here, can help provide a systematic basis for including the mentalist concepts by showing how they may relate to the high-level architecture of a multiagent system.

6. DISCUSSION

The above approach considers trust in propositional terms. Most practical settings need parameters, which we can accommodate in a fairly straightforward manner. Similarly, an expansion to graded or measured notions of trust would be valuable. We can potentially develop such a notion by adopting some ideas of Demolombe [5]. Indeed, there is a conceptually straightforward mapping of our models to the above, which would arise by assigning relative weights to the sets of runs that our model-theoretic standard of trust $\mathbb{T}$ identifies. When such sets of runs can be assigned likelihoods of occurrence, they can additionally be used as a basis for a probabilistic definition of trust.

Trust is inherently contextual. As a result, in some uses the preconditions that apply on a claim of trust may not be explicit. Such implicit preconditions can be mapped to

antecedents in an explicit representation. Organizational context is particularly relevant from our architectural perspective: an agent may depend upon another when they are both part of the same team or organization.

Following a similar distinction for commitments [18], we can distinguish two main kinds of trust: (1) *dialectical*, i.e., about assertions or arguments relating to reports [4, 12]; or (2) *practical*, i.e., about actions, as in the present paper. We can relate the above dichotomy to trust in an agent viewed as a service provider and an agent viewed as a referrer. Examples are "if the interest rate has fallen, (I trust) my banker to grant my mortgage application (practical) or (I trust) my banker's assertion of my new loan payment (dialectical).

Following the spirit of correspondence theory as proposed by van Benthem [22], the above postulates can be given a model-theoretic basis wherein for each postulate we state a corresponding semantic constraint (in essence, a closure property) on the model. For reasons of space, we defer such constraints and theorems to a longer version of this paper.

Directions

Some important directions of future work fall out naturally from the above formal, architectural development of trust.

In *conceptual terms*, a deeper study of the reasoning postulates would be beneficial in a wide range of multiagent applications. In particular, it would be important to determine additional architectural styles. We considered simplistic multiagent systems above. This is an important start in formalizing trust, but it would be valuable to expand on this theme to specify richer systems and postulates about them. Specifically, above we treated agents as either individuals or sets of agents. In general, multiagent systems would demonstrate rich structures and consist of roles that feature in a variety of operational and institutional relationships with each other. Such relationships would naturally bear a significant impact on trust understood architecturally.

In *theoretical terms*, a rich formal language for expressing constraints and reasoning about them to determine if a particular architecture style or instance will satisfy desirable properties such as a guarantee of progress under appropriate assumptions on the behaviors of the participants. Makinson and van der Torre [13] introduced the idea of input-output logics as a general way to treat conditionalization. Our approach can be thought of as specializing their ideas for the setting of trust with inferences for completion, commitments, and teamwork that do not arise with conditionals in general, but are important for an understanding of trust. It would be interesting to explore what insights we can adopt from input-output logics.

In *practical terms*, an important consideration is of a pattern language for expressing architectures. Such a language could provide a basis for a tool and methodology for specifying architectures. A greater goal is to develop an extensive approach for *service-oriented computing* in the broadest sense of the term that considers not technical (web or grid) services as emphasized today but service engagements mediated by flexible and expressive trust relations.

Acknowledgments

Thanks to the anonymous referees and to Amit Chopra for helpful comments. Thanks to the Army Research Laboratory for partial support under Cooperative Agreement Number W911NF-09-2-0053.

7. REFERENCES

[1] C. Castelfranchi, R. Falcone. Principles of trust for MAS. *ICMAS*, pp. 72–79, 1998.

[2] A. K. Chopra, M. P. Singh. Constitutive interoperability. *AAMAS*, pp. 797–804, May 2008.

[3] P. Dasgupta. Trust as a commodity. In D. Gambetta, ed., *Trust: Making and Breaking Cooperative Relations*, ch. 4, pp. 49–72. 2000.

[4] M. Dastani, A. Herzig, J. Hulstijn, L. van der Torre. Inferring trust. *AAMAS CLIMA*, *LNCS* 3487, pp. 144–160. Springer, 2004.

[5] R. Demolombe. Graded trust. *AAMAS Trust*, pp. 1–12, 2009.

[6] N. Desai, A. K. Chopra, M. P. Singh. Amoeba: A methodology for modeling and evolution of cross-organizational business processes. *ACM TOSEM*, 19(2):6:1–6:45, October 2009.

[7] R. Falcone, C. Castelfranchi. From dependence networks to trust networks. *AAMAS Trust*, 2009.

[8] R. Falcone, G. Pezzulo, C. Castelfranchi, G. Calvi. Contract nets for evaluating agent trustworthiness. *Proc. 6th & 7th Trust Workshops*, *LNCS* 3577, ch. 3, pp. 43–58. Springer, 2005.

[9] K. Fullam, K. S. Barber. Dynamically learning sources of trust information. *AAMAS*, pp. 1062–1069, 2007.

[10] M. Johnson, J. M. Bradshaw, P. Feltovich, C. Jonker, M. B. van Riemsdijk, M. Sierhuis. Coactive design. *AAMAS COIN Workshop*, pp. 49–56, 2010.

[11] A. Jøsang. A subjective metric of authentication. *ESORICS*, *LNCS* 1485, pp. 329–344, 1998. Springer.

[12] C.-J. Liau. Belief, information acquisition, and trust in multi-agent systems. *Art. Intell.*, 149(1):31–60, 2003.

[13] D. Makinson, L. van der Torre. Input-output logics. *J. Philosophical Logic*, 29:383–408, 2000.

[14] R. Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.

[15] D. Parnas. Information distribution aspects of design methodology. *Proc. IFIP*, TA-3, pp. 26–30, 1971.

[16] L. Penserini, A. Perini, A. Susi, J. Mylopoulos. High variability design for software agents: Extending Tropos. *ACM TAAS*, 2(4):16:1–16:27, November 2007.

[17] M. Shaw, D. Garlan. *Software Architecture*. Prentice-Hall, 1996.

[18] M. P. Singh. Semantical considerations on dialectical and practical commitments. *AAAI*, pp. 176–181, 2008.

[19] M. P. Singh, A. K. Chopra. Programming multiagent systems without programming agents. *ProMAS 2009 Workshop*, *LNAI* 5919, pp. 1–14. Springer, 2010.

[20] P. Sztompka. *Trust: A Sociological Theory*. Cambridge University Press, 1999.

[21] C. J. van Aart et al. Use case outline and requirements. IST CONTRACT Project, 2007.

[22] J. F. A. K. van Benthem. Correspondence theory. In D. Gabbay, F. Guenthner, eds., *Hbk Phil. Log*, vol. II, pp. 167–247. Reidel, 1984.

[23] Y. Wang, M. P. Singh. Formal trust model for multiagent systems. *IJCAI*, pp. 1551–1556, 2007.

[24] P. Yolum, M. P. Singh. Enacting protocols by commitment concession. *AAMAS*, pp. 116–123, 2007.

Multi-Layer Cognitive Filtering by Behavioral Modeling

Zeinab Noorian
University of New Brunswick
Fredericton,Canada
z.noorian@unb.ca

Stephen Marsh
Communications Research
Centre, Canada
stephen.marsh@crc.gc.ca

Michael Fleming
University of New Brunswick
Fredericton,Canada
mwf@unb.ca

ABSTRACT

In the absence of legal enforcement procedures for the participants of an open e-marketplace, trust and reputation systems are central for resisting against threats from malicious agents. Such systems provide mechanisms for identifying the participants who disseminate unfair ratings. However, it is possible that some of the honest participants are also victimized as a consequence of the poor judgement of these systems. In this paper, we propose a two-layer filtering algorithm that cognitively elicits the behavioral characteristics of the participating agents in an e-marketplace. We argue that the notion of *unfairness* does not exclusively refer to deception but can also imply differences in dispositions. The proposed filtering approach aims to go beyond the inflexible judgements on the quality of participants and instead allows the human dispositions that we call optimism, pessimism and realism to be incorporated into our trustworthiness evaluations. Our proposed filtering algorithm consists of two layers. In the first layer, a consumer agent measures the competency of its neighbors for being a potentially helpful adviser. Thus, it automatically disqualifies the deceptive agents and/or the newcomers that lack the required experience. Afterwards, the second layer measures the credibility of the surviving agents of the previous layer on the basis of their behavioral models. This tangible view of trustworthiness evaluation boosts the confidence of human users in using a web-based agent-oriented e-commerce application.

Categories and Subject Descriptors

[**distributed artificial intelligence**]: multi-agent systems

General Terms

Human Factors, Design, Measurement

Keywords

Trust, Reputation, Cognitive filtering, Behavioral modeling

1. INTRODUCTION

The inherent uncertainties in an open e-marketplace inhibit participants from reaching a mutual understanding and confidence about each other's intentions [3]. This matter affects the formation of agent-based e-commerce applications handled by human users since their personal dispositions are not allowed to be reflected in their decisions or, if so, other participants are not able to identify them[8]. As such, despite the intrinsic honesty of their recommendations, they may not be considered trustworthy. This is undoubtedly justifiable with the existence of malicious participants. In particular, in order to diminish the risk of being misled by unfair advisers, a consumer agent restricts itself to seek advice from the participants with the most similar ratings[5, 14].

In this paper we intend to amend this common view of trustworthiness [11, 16] by introducing a new definition for *unfairness*. We discuss that the intuition of unfairness could be examined across two categories: 1) *intentional*, a) participants consistently act malevolently and b) participants occasionally engage in deceitful activities. And 2) *unintentional*, as a result of a) lack of personal experience and b) various behavioral characteristics resulting in different rating attitudes.

We propose a two-layered filtering algorithm that combines cognitive and probabilistic views of trust [3] to mainly target the intentional group of unfair advisers. We show that modeling the trustworthiness of advisers based on a strict judgement of the quality of their recommendations is not complete unless it is accompanied by the analysis of their dispositions. Thus, through the comprehension of their rating attitudes, a consumer agent could take appropriate steps to evaluate them.

The main contributions of this paper are twofold: First, we propose competency evaluation methods to detect newcomers with a lack of experience and thereafter disqualify them from the role of advisers. Second, we introduce a classification schema to identify the behavioral characteristics of participants and design credibility assessment measures for each of them.

Our experimental results show the utility of our approach in terms of recognizing dispositions of various participants and, specifically, how consumers with personalized thought-frames evaluate the same adviser differently. Our filtering model can therefore be seen as an effective approach in modeling the reputation of advisers in a multi-agent system.

2. TWO-LAYERED COGNITIVE FILTERING ALGORITHM

To formalize the proposed cognitive filtering algorithm, we consider the scenario where, in an electronic marketplace, consumer agents with distinctive behavioral patterns want to bootstrap relationships with new neighbors. We assume

Cite as: Multi-Layer Cognitive Filtering by Behavioural Modeling, Zeinab Noorian, Stephen Marsh and Michael Fleming, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 871-878.

that the consumer agents have some record of interactions with transaction partners, i.e, providers. We also assume that participating agents are cooperative and willing to communicate with each other.
To analyze the neighbors' trustworthiness, each consumer agent C needs two types of information. The first type of information, which helps to build the first layer of our filtering algorithm, is used to identify malicious participants with a complementary model of deception. It also detects newly-joined agents with an insufficient number of personal experiences.
In the second layer of the filtering algorithm, the second type of information helps C to recognize the behavioral characteristics of the neighbors. As such, it will be able to evaluate their degree of trustworthiness. Note that, in this layer, C takes an analytical approach in order to detect deceitful participants with volatile dispositions who cheat opportunistically. By hiding their true intentions, this group of deceitful participants imposes greater risk and insecurity to the system compared with those with a frequently deceptive attitude[1, 6, 7].
The detailed explanation of this multi-dimensional filtering technique is provided in the following sub-sections.

2.1 First Layer: Evaluating the Competency Degree of Neighbors

The consumer agent C sends a query to a circle of its neighbors $N = \{N_1, N_2, ..., N_i\}$ requesting numbers of successful and unsuccessful outcomes experienced with providers $P = \{P_1, P_2, ..., P_r\} \subseteq \{P_1, P_2, ..., P_m\}, r \leq m$, occurring before a certain time T. Such a time threshold diminishes the risk of changeability in a provider's behavior. It is also accompanied by the Quality of Service (QoS) threshold Ω to imply C's belief about an acceptable minimum level of trust. For example, for a consumer with a *risk-averse* pattern, Ω could be 0.7 whereas for the *risk-taking* consumer this amount might be reduced to 0.5.
The neighbor N_k responds by providing a rating vector $R_{(N_k,P_j)}$ for each provider. It contains a tuple of $\langle r, s\rangle$ which indicates the number of successful (r) and unsuccessful (s) interaction results with provider P_j respectively. Note that, in the first layer of the filtering algorithm, neighbors are asked to provide merely a *binary rating* ("1" or "0" for example), in which "1" means that P_j is reputable and "0" means not reputable. Thus, considering a consumer's QoS threshold, they will send reputation reports as a collection of positive and negative interaction outcomes.
Once the evidence is received, for each $R_{(N_k,P_j)}$, C calculates the expected value of the probability of the positive outcome for a provider P_j [9] as:

$$E(pr_r, P_j) = \frac{r+1}{r+s+2} \tag{1}$$

To generally present this formula to include all participants in an e-marketplace, we update the presentation of $E(pr_r, P_j)$ to $E(pr_r, P_j)_{Par}$, where $Par \in \{C\} \cup N$ implies participants of the community. Clearly, $0 < E(pr_r, P_j)_{Par} \leq 1$ and as it approaches 0 or 1, it indicates *unanimity* in the body of evidence[4]. That is, particularly large values of s or r provide better intuition about an overall tendency and service quality of providers. In contrast, $E(pr_r, P_j)_{Par} = 0.5$ (i.e, $r = s$) signifies the maximal conflict in gathered evidence, resulting in increasing the uncertainty in determining the service quality of providers. Based on these intuitions, we are able to calculate the degree of reliability and certainty of ratings provided by neighbors.
Let x represent the probability of a successful outcome for a certain provider. Based on the Definitions(2) and (3) in [12], the *Reliability degree* of each $R_{(N_k,P_j)}$ is defined as:

$$c(r, s) = \frac{1}{2}\int_0^1 \mid \frac{x^r(1-x)^s}{\int_0^1 x^r(1-x)^s\ dx} - 1 \mid\ dx \tag{2}$$

Similar to $E(pr_r, P_j)_{Par}$, we update the presentation of $c(r, s)$ to $c(r, s)_{Par}$.
Theoretical analysis [12] demonstrates that, for a fixed ratio of positive and negative observations, the reliability increases as evidence increases. On the contrary, given a fixed amount of evidence, as the extent of conflict increases, the reliability of the provided ratings decreases proportionately. That is, reliability is at its minimum value when $E(pr_r, P_j)_{Par} = 0.5$. As such, the less conflict in their ratings, the more reliable the neighbors would be.
However, in the proposed filtering algorithm, C would not strictly judge the neighbors with rather low reliability in their $R_{(N_k,P_j)}$ as deceptive participants since this factor could signify both dishonesty of neighbors and the dynamicity and fraudulent behavior of providers. That is, some malicious providers may adopt a strategy of providing satisfactory quality of service in most situations when there is not much at stake and acting conversely in occasions associated with a large gain. As such, even though they retain a certain level of trustworthiness, their associated reliability degree is low. To address this ambiguity, C computes the $E(pr_r, P_j)_C$ and $c(r, s)_C$ of its personal experiences; $R_{(C,P_j)}$, for a common set of providers. Through the comparison of neighbors' metrics with its own, it would select those with a similar rating pattern and a satisfactory level of honesty as its *advisers*.
To formalize this, it measures an average level of dishonesty of N_k by differentiating their $E(pr_r, P_j)_{Par}$ as:

$$\overline{d}_{(N_k)} = \frac{\sum_{j=1}^{|P|} \mid E(pr_r, P_j)_C - E(pr_r, P_j)_{N_k} \mid}{|P|} \tag{3}$$

As pointed out, increasing the amount of evidence leads to an increase in the reliability degree. The problem arises when malicious neighbors disseminate a large number of spurious ratings so as to promote their reliability. Besides, it may happen that a truthful neighbor lacks in number of experiences. Thus, despite its inherent honesty, its reliability degree is low and it is not qualified to play the role of adviser. To clarify these issues, we define an uncertainty function $\overline{U}_{(N_k)}$ to capture the intuition of information imbalance between C and N_k as follows:

$$\overline{U}_{(N_k)} = \frac{\sum_{j=1}^{|P|} \mid (c(r, s)_C - c(r, s)_{N_k})_{P_j} \mid}{|P|} \tag{4}$$

In light of the uncertainty function, the opinions of deceptive neighbors who attempt to mislead consumer agents by supplying a large number of ratings are discounted. Similarly, it hinders short-term observations of newly-joined agents from having influence on a consumer agent's decision making process.
Given the formulae (3) and (4), the *competency degree* of N_k is calculated by reducing its honesty based on its certainty

degree. Thus, it could be determined as:

$$Comp_{(N_k)} = (1 - \overline{d}_{(N_k)}) * (1 - \overline{U}_{(N_k)}) \quad (5)$$

By comparing their competency degree with a pre-defined incompetency tolerance threshold μ, C evaluates the qualification and eligibility of the neighbors to play the role of *adviser*. As such, It chooses the neighbors with $(1 - Comp_{(N_k)}) \leq \mu$ as its potential advisers and filters out the rest. It is worthwhile to note that, since in this layer we target the participants with a significant lying pattern, detecting fraudulent agents with oscillating rating attitudes is left for the next layer.

2.2 Second Layer: Calculating a Credibility Degree of Advisers

In the first phase of the filtering algorithm, neighbors are asked to send their subjective opinions of providers. By aggregating their opinions and computing their degree of reliability, a consumer agent has obtained a rough estimation of the honesty level of neighbors and selects a subset of them as its advisers. However, this method cannot thoroughly address the inherent complications of an open environment. To explain, the nature of the open marketplace allows various kinds of participants with distinctive behavioral characteristics [2] to engage in the system.

Besides, the basis of the employed multi-dimensional rating system provides tools for a consumer agent to objectively evaluate the performance of service providers across several criteria with different degrees of preference. Evidently, the measured QoS is mainly dependent on how much the criteria with a high preference degree are fulfilled[7]. Owing to the different purchasing behavior of the agents, it is expected that preference degrees vary from one participant to another, resulting in dissimilar assessment of the quality of the *same* service. As such, computing the credibility of advisers regardless of their behavioral characteristics and rating attitudes, and merely based on their subjective opinions, would not sufficiently ensure high quality judgements of their trustworthiness.

To tackle these problems, in a second layer of the filtering algorithm, consumer agent C steps forward and analytically gives credits to advisers to the extent that their evaluation of each criterion of a negotiated context is similar to its own experiences. For this purpose, it asks advisers about mutually agreed criteria on which they have bargained with *highly-reliable* providers[1] whose reputation values have been recently released in the form of binary ratings. They also are requested to include the most recent interaction time with such information so as to give a higher weight to more recent feedback. That is, feedback gradually loses its importance as time progresses. This improves the correctness and accuracy in predicting the credibility of advisers through alleviating the risk of changeability in a provider's behavior.

To formulate this, we adopt the concept of forgetting factor presented in [9, 16]:

$$z = \lambda^{T_A - T_C} \quad (6)$$

We customize it for our model and define a recency factor as:

$$T_{(C,A_k)_{P_j}} = \frac{1}{z} \quad (7)$$

Here, T_A and T_C indicate the adviser's and consumer's time windows when they had an experience with a provider P_j. Also, the λ represents the forgetting parameter and $0 < \lambda \leq 1$. When $\lambda = 1$, there is no forgetting and all the ratings are treated as though they happened in the same time period. In contrast, $\lambda \approx 0$ specifies that ratings from different time windows will not be significantly taken into account. Similarly to [16], in this filtering algorithm, the recency factor is characterized with a discrete integer value where 1 is the most recent time period and 2 is the time period just prior. Also, it is presumed that the adviser's ratings are prior to those a consumer agent supplies so that $T_A \geq T_C$.

Adviser A_k will respond, providing an interaction context $IC_{(A_k,P_j,T_A)}$ that contains a tuple of *weight* and *value*: $\{W_i.V_i | i = 1..n\}$ and the latest interaction time T_A for each provider.[2]

Given A_k's interaction context, a consumer agent would estimate the possible interaction outcomes of an adviser based on its own perspective. That is, C will examine its $IC_{(C,P_j,T_C)}$, which contains pairs of weight and value: $\{Y_i.R_i | i = 1..n\}$. It will then modify the interaction context of A_k by replacing A_k's preferences W_i with its own personal preference degrees Y_i. Based on this, the interaction context of A_k is updated to: $IC'_{(A_k,P_j,T_A)} = \{Y_i.V_i | i = 1..n\}$. To formalize a similarity of A_k's rating approach with C, we compute a ratio of the consumer's interaction context $IC_{(C,P_j,T_C)}$ with the updated version of the adviser's interaction context as:

$$Sim_{(C,A_k)_{P_j}} = \frac{\sum_{i=1}^{n} Y_i \times R_i}{\sum_{i=1}^{n} Y_i \times V_i} \quad (8)$$

and then

$$Diff_{(C,A_k)_{P_j}} = 1 - Sim_{(C,A_k)_{P_j}}$$

represents the difference of C and A_k in assessing P_j.

Based on Equations (7) and (8), C would calculate the *average* differences between the transaction result of A_k and its own experiences with a same set of providers as:

$$\overline{Diff}_{(C,A_k)} = \frac{\sum_{j=1}^{|P|} \mid Diff_{(C,A_k)_{P_j}} \mid * T_{(C,A_k)_{P_j}}}{|P|} \quad (9)$$

Existing trust models [5, 9, 11, 14, 16] evaluate the trustworthiness of advisers mainly based on their average deviation from a consumer's opinion and exploit the *same* credibility measures for all types of advisers. Moreover, they define a threshold value[3] to separate the honest advisers from dishonest ones. However, adjusting a threshold to an efficient value has always been a controversial issue. The quality of advisers is compromised when a threshold is set to a high value. In this situation, deceitful participants who maintain a minimum level of trustworthiness remain undetected and could actively contribute to a consumer's decision making process. On the other hand, a lower threshold

[1]Obviously, a consumer only inquires about the providers with high reliability and ignores those that are possibly deceptive.

[2]Note that in this model we assume that each provider can only provide one particular service. Dealing with providers offering multiple services is left for future work.

[3]A threshold can be explicitly determined as in [5] and [14] or implicitly as in[16].

leads to the contribution of a smaller number of advisers. Clearly, adjusting a threshold value is a trade-off between the number of credible advisers and the risk of being misled by deceptive peers.

Furthermore, in a real-life e-commerce application, the differences in a consumer's behavioral patterns lead to divergent evaluations of the credibility degree of advisers[8]. For instance, the opinion of one particular adviser may seem highly credible for a risk-taking consumer while it is not so for a risk-averse one. We note that the credibility degree of advisers *not only* depends on their evaluator's dispositions but it is also related to their own individual behavioral patterns. That is, advisers' recommendations could be affected by endogenous factors[3]. As such, it may happen that two honest advisers with different attitudes have conflicting evaluations of the same provider. Characterizing the disposition of advisers helps a consumer agent to take a proportionate strategy in assessing their future recommendations. For instance, a risk-averse consumer would underestimate the ratings provided by optimistic advisers whilst overrating those provided by pessimistic advisers. This mechanism shows its practicality in a community where credible advisers are scarce and the majority of participants behave malevolently. In this state, modeling a behavior of advisers helps a consumer to get the most benefits from their opinions in such a way that the scarcity of credible advisers would not have a serious effect on the quality of predictions.

For all these reasons, in this model we take a further step and embrace the diversity in participants as an influential factor in our credibility measures. We believe that quantity should not necessarily be sacrificed for quality or vice versa. Instead, by employing a suitable mechanism, consumer agents are able to have a large number of advisers with high-quality judgements. As such, C captures the overall tendency of A_k in evaluating the providers' QoS as:

$$Tendency_{(C,A_k)} = \frac{\sum_{j=1}^{|P|} Diff_{(C,A_k)_{P_j}}}{|P|} \tag{10}$$

As the name suggests, the consumer agent could exploit a tendency metric to get an intuition about the general trends of advisers in rating a common set of providers. That is, a positive value of $Tendency_{(C,A_k)}$ indicates that an adviser has the attitude of overrating providers while a negative value declares that an adviser has a tendency to underrate providers.

Following that, to identify a behavioral pattern of advisers, we determine a pre-defined boundary β such that if A_k's $IC'_{(A_k,P_j,T_A)}$ is compatible with those experienced by C ($\overline{Diff}_{(C,A_k)} \leqslant \beta$), they will be counted as *credible* advisers. However, in this model, C would not thoroughly exclude the advisers who rate otherwise. Instead, it narrowly analyzes the $\overline{Diff}_{(C,A_k)}$ in such a way that if it is marginally greater than β with a negative $Tendency_{(C,A_k)}$, the corresponding adviser's attitude is identified as *pessimistic*.

Similarly, in case their differences marginally exceed β with a positive $Tendency_{(C,A_k)}$, the respective adviser's attitude is recognized as *optimistic*. We define such a marginal error ϵ as a ratio of the credibility threshold β and it is subjectively determined by a consumer agent. Evidently, if A_k's $IC'_{(A_k,P_j,T_A)}$ significantly deviates from the consumer agent's direct experiences, they will be detected as *malicious* advisers with *deceitful* behavioral models. We believe that the filtered advisers have a deceitful behavioral pattern; otherwise, they would have been expelled in the first layer.

Note that the thresholds are used to identify different kinds of unfair participants. These thresholds should be set with the goals of each particular layer in mind. In the first layer, the value of μ should be high, to ensure that dishonest participants are expelled. In the second layer, when analyzing participants' behavioral characteristics, a low value of β is desirable. Thus, we can conclude that $\beta \leq \mu$.

The classification mechanism of the behavioral pattern of A_k based on C's interaction context is formally presented as follows:

$$BP_{(C,A_k)} = \begin{cases} Realistic/Credible: \\ \quad \overline{Diff}_{(C,A_k)} \leqslant \beta \\ Optimistic: \\ \quad \beta < \overline{Diff}_{(C,A_k)} \leqslant \beta + \epsilon \ \&\ Tendency_{(C,A_k)} > 0 \\ Pessimistic: \\ \quad \beta < \overline{Diff}_{(C,A_k)} \leqslant \beta + \epsilon \ \&\ Tendency_{(C,A_k)} < 0 \\ Deceitful: \\ \quad \overline{Diff}_{(C,A_k)} > \beta + \epsilon \end{cases} \tag{11}$$

Given the $BP_{(C,A_k)}$, the credibility measure $CR_{(C,A_k)}$ is formulated as:

$$CR_{(C,A_k)} = \begin{cases} 1 - \overline{Diff}_{(C,A_k)}: \\ \qquad BP_{(A_k)} = Credible \\ (1 - \overline{Diff}_{(C,A_k)}) \times e^{-\theta * \overline{Diff}_{(C,A_k)}}: \\ \qquad BP_{(A_k)} = Optimistic \\ (1 - \overline{Diff}_{(C,A_k)}) \times e^{-\sigma * \overline{Diff}_{(C,A_k)}}: \\ \qquad BP_{(A_k)} = Pessimistic \\ 0: \\ \qquad BP_{(A_k)} = Deceitful \end{cases} \tag{12}$$

Here, θ and σ represent the optimistic and pessimistic coefficients respectively. A consumer agent takes a personalized adaptive approach to calculate them. Depending on its behavioral characteristics, such coefficients are initialized differently. For instance, recommendations of pessimistic advisers may seem more credible in the perspective of a risk-averse consumer and they are considered to be better peers to cooperate with than optimistic advisers [2]. Hence the risk-averse consumer promotes the credibility of a pessimistic adviser by adjusting the pessimistic coefficient to a lower value than the optimistic coefficient$\langle 0 \leq \sigma < \theta \rangle$. On the contrary, the disposition of a risk-taking buying agent compels it to consider the reputation information provided by optimistic advisers as more important. Therefore, it assigns a great deal of influence to their ratings by properly setting up the optimistic coefficient to a lower value than the pessimistic coefficent $\langle 0 \leq \theta < \sigma \rangle$.

As such, the coefficients are adaptively defined for each adviser. For initializing θ, a risk-averse agent considers the maximum difference of the adviser's ratings with a consumer's opinions upon evaluating the same providers. For a risk-taking agent, this process is reversed. That is, the optimistic coefficient is defined as the minimum deviation of the adviser's recommendations with the consumer's opinions across a common set of providers. Thus, coefficients θ and σ are formalized as:

$$\theta = \begin{cases} max\{|\ Diff_{(C,A_k)_{P_i}}\ | \ | i = 1...m\} & Risk\text{-}Averse\ consumer \\ min\{|\ Diff_{(C,A_k)_{P_i}}\ | \ | i = 1...m\} & Risk\text{-}Taking\ consumer \end{cases} \tag{13}$$

$$\sigma = \begin{cases} min\{|\ Diff_{(C,A_k)_{P_i}}\ |\ |i = 1...m\} & Risk\text{-}Averse\ consumer \\ max\{|\ Diff_{(C,A_k)_{P_i}}\ |\ |i = 1...m\} & Risk\text{-}Taking\ consumer \end{cases} \quad (14)$$

Through these principles, A_k's recommendations are discounted such that its influence on C's prediction depends on its honesty in *each* of its interaction contexts. The coefficient parameters ensure that the recommendation of advisers with volatile behavior who have a high variability in their opinions is heavily discounted.

3. EXAMPLES

In an electronic marketplace, a consumer C_1 needs to make a decision on whether to interact with a provider P_1. This depends on how much C_1 trusts P_1. To model the trustworthiness of P_1, when the consumer does not have an adequate number of experiences with P_1, it ought to seek advice from its neighbors. However, it first needs to acquire enough information about their credibility value in order to assign a proper credibility to their provided ratings.

In the first phase of the filtering algorithm, the risk-averse C_1 asks its surrounding neighbors $\{N_1, N_2, .., N_6\}$ about the overall performance of the providers $\{P_1, P_2, .., P_8\}$ before time T, given the QoS threshold $\Omega = 0.7$.

Consider the case where the neighbors $\{N_1, .., N_5\}$ have rated only the five providers $\{P_1, P_2, P_3, P_4, P_5\}$. Using Equations (1) and (2), C_1 would calculate the expected value of the probability of positive ratings along with a degree of reliability of their evidence. Table 1 lists a number of successful/unsuccessful ratings provided by $N_i (i \in \{1, .., 5\})$ and C_1 for the five providers along with their $E(pr_r, P_j)_{N_i}$ and $c(r, s)_{N_i}$.

Table 1: Ratings provided by the neighbors and C_1 along with their corresponding metrics

Participants	P_i	$\langle r, s \rangle$	$E(pr_r, P_j)_{Par}$	$c(r,s)_{Par}$
C_1	P_1	$\langle 16, 1 \rangle$	0.89	0.71
	P_2	$\langle 7, 4 \rangle$	0.61	0.47
	P_3	$\langle 2, 10 \rangle$	0.21	0.57
	P_4	$\langle 15, 0 \rangle$	0.94	0.77
	P_5	$\langle 2, 4 \rangle$	0.37	0.38
N_1	P_1	$\langle 25, 0 \rangle$	0.96	0.84
	P_2	$\langle 8, 3 \rangle$	0.69	0.5
	P_3	$\langle 2, 5 \rangle$	0.33	0.42
	P_4	$\langle 8, 0 \rangle$	0.9	0.67
	P_5	$\langle 3, 2 \rangle$	0.57	0.33
N_2	P_1	$\langle 8, 5 \rangle$	0.6	0.54
	P_2	$\langle 9, 5 \rangle$	0.62	0.51
	P_3	$\langle 5, 5 \rangle$	0.50	0.44
	P_4	$\langle 11, 6 \rangle$	0.63	0.55
	P_5	$\langle 3, 4 \rangle$	0.44	0.38
N_3	P_1	$\langle 13, 2 \rangle$	0.82	0.62
	P_2	$\langle 2, 6 \rangle$	0.3	0.46
	P_3	$\langle 4, 7 \rangle$	0.38	0.47
	P_4	$\langle 20, 5 \rangle$	0.77	0.65
	P_5	$\langle 1, 11 \rangle$	0.15	0.64
N_4	P_1	$\langle 4, 11 \rangle$	0.29	0.55
	P_2	$\langle 6, 4 \rangle$	0.58	0.45
	P_3	$\langle 13, 5 \rangle$	0.7	0.57
	P_4	$\langle 5, 9 \rangle$	0.37	0.51
	P_5	$\langle 10, 6 \rangle$	0.61	0.53
N_5	P_1	$\langle 2, 0 \rangle$	0.75	0.38
	P_2	$\langle 1, 0 \rangle$	0.66	0.25
	P_3	$\langle 1, 2 \rangle$	0.4	0.27
	P_4	$\langle 1, 0 \rangle$	0.66	0.25
	P_5	$\langle 0, 1 \rangle$	0.33	0.25

To calculate the competency degree of neighbors, C_1 would analyze their average dishonesty. Through $\overline{U}_{(N_k)}$, it also examines the adequacy of their ratings. Afterwards, using Equation (5), it is able to calculate their $Comp_{(N_k)}$, resulting in detection of particular neighbors with consistent deceptive attitudes and those with few experiences. Here, a risk-averse C_1 selects the neighbors $\{N_1, N_2, N_3\}$ whose competency values $Comp_{(N_k)}$ surpass $\mu = 0.65$ and filters out the rest (Table 2). Next, in the second layer, C_1 requests detailed descriptions of their negotiated criteria with the selected set of providers so as to identify the behavioral characteristics of advisers. Table 3 articulates personal ratings of each participant through the $\langle weight, value \rangle$ pair related to each criterion regarding the selected providers.

Table 2: Calculating the competency level of the neighbors

N_i	$\overline{d}_{(N_k)}$	$\overline{U}_{(N_k)}$	$Comp_{(N_k)}$
N_1	0.1	0.08	0.81
N_2	0.19	0.12	0.70
N_3	0.19	0.11	0.71
N_4	0.38	0.12	0.54
N_5	0.13	0.3	0.59

As can be perceived, in the first layer, the disposition of consumers has not been reflected in the evaluation of the competency degree of the neighbors. To observe the influence of this factor in the second layer, we introduce a risk-taking C_2 in addition to C_1 and examine their approaches in evaluating the same advisers. Finally, as indicated in Tables 4 and 5, the behavioral patterns of participants could serve as determinant factors in evaluating their trustworthiness. We notice that, consumer agents with similar deviation ($\overline{Diff}_{(C_1,N_1)} = \overline{Diff}_{(C_2,N_1)}$) and the same $\beta = 0.15, \epsilon = 0.07$ and $\lambda = 0.8$ could predict different credibility values for the same adviser. Note that, in Table 4, we use the notation $Tend_{(C_1,N_k)}$ for $Tendency_{(C_1,N_k)}$.

Table 3: The negotiated criteria of participants with selected providers

Participants	P_i	$Criteria\langle w, v \rangle$				T
		Cri_1	Cri_2	Cri_3	Cri_4	
C_1	P_1	$\langle 6, 7 \rangle$	$\langle 10, 9 \rangle$	$\langle 10, 8 \rangle$	$\langle 5, 10 \rangle$	2
	P_3	$\langle 10, 4 \rangle$	$\langle 7, 2 \rangle$	$\langle 5, 5 \rangle$	$\langle 10, 3 \rangle$	2
	P_4	$\langle 6, 5 \rangle$	$\langle 3, 10 \rangle$	$\langle 10, 10 \rangle$	$\langle 8, 6 \rangle$	1
C_2	P_1	$\langle 3, 8 \rangle$	$\langle 10, 9 \rangle$	$\langle 9, 7 \rangle$	$\langle 8, 9 \rangle$	3
	P_3	$\langle 9, 3 \rangle$	$\langle 8, 4 \rangle$	$\langle 10, 5 \rangle$	$\langle 7, 4 \rangle$	1
	P_4	$\langle 2, 8 \rangle$	$\langle 10, 10 \rangle$	$\langle 8, 10 \rangle$	$\langle 6, 6 \rangle$	1
N_1	P_1	$\langle 4, 9 \rangle$	$\langle 10, 10 \rangle$	$\langle 10, 8 \rangle$	$\langle 6, 10 \rangle$	3
	P_3	$\langle 10, 5 \rangle$	$\langle 6, 2 \rangle$	$\langle 4, 6 \rangle$	$\langle 10, 5 \rangle$	3
	P_4	$\langle 3, 8 \rangle$	$\langle 10, 10 \rangle$	$\langle 10, 10 \rangle$	$\langle 6, 7 \rangle$	7
N_2	P_1	$\langle 7, 5 \rangle$	$\langle 10, 7 \rangle$	$\langle 10, 4 \rangle$	$\langle 4, 6 \rangle$	4
	P_3	$\langle 10, 6 \rangle$	$\langle 5, 6 \rangle$	$\langle 2, 7 \rangle$	$\langle 10, 6 \rangle$	3
	P_4	$\langle 10, 3 \rangle$	$\langle 5, 7 \rangle$	$\langle 10, 6 \rangle$	$\langle 7, 4 \rangle$	2
N_3	P_1	$\langle 10, 6 \rangle$	$\langle 10, 8 \rangle$	$\langle 10, 7 \rangle$	$\langle 5, 9 \rangle$	4
	P_3	$\langle 10, 3 \rangle$	$\langle 7, 2 \rangle$	$\langle 5, 4 \rangle$	$\langle 10, 2 \rangle$	3
	P_4	$\langle 8, 5 \rangle$	$\langle 3, 10 \rangle$	$\langle 10, 9 \rangle$	$\langle 7, 5 \rangle$	1

Table 4: Calculating tendency of neighbors and their deviation degree based on C_1 and C_2's experiences

N_k	$\overline{Diff}_{(C_1,N_k)}$	$Tend_{(C_1,N_k)}$	$\overline{Diff}_{(C_2,N_k)}$	$Tend_{(C_2,N_k)}$
N_1	0.18	0.13	0.18	0.14
N_2	0.60	-0.22	0.53	-0.22
N_3	0.17	-0.14	0.18	-0.16

Table 5: Behavioral pattern and credibility degree of neighbors determined by C_1 and C_2

Consumer	N_k	$BP_{(C,N_k)}$	$CR_{(C,N_k)}$	θ	σ
C_1	N_1	Optimistic	0.79	0.18	N/A
	N_2	Deceitful	0	N/A	N/A
	N_3	Pessimistic	0.82	N/A	1.08
C_2	N_1	Optimistic	0.82	0.02	N/A
	N_2	Deceitful	0	N/A	N/A
	N_3	Pessimistic	0.76	N/A	0.34

4. EXPERIMENTAL RESULTS

Our approach models the trustworthiness of advisers, not only based on their honesty degree but also by examining their competency level. That is, an honest adviser but with insufficient experiences is not qualified to provide advice. Furthermore, we claim that having a good comprehension of the adviser's disposition leads to a more adaptive credibility assessment. For this purpose, we have conducted two classes of experiments. The first class is designed to indicate the effectiveness of the proposed model in detecting

malicious neighbors as well as newcomers with insufficient experiences. In the second class of experiments we put the second layer to the test and observe how the *same* advisers could have *different* credibility values according to different consumers. We also estimate the accuracy of our prediction by comparing it with the actual trustworthiness value of advisers, obtained by averaging over multiple experiences.

The first series of experiments evaluates the competency level of an intrinsically *honest* neighbor having different numbers of experiences. It involves one consumer C asking a neighbor N_k about its common experiences with 2 and 50 providers. N_k provides percentages (ranging from 0% to 100%) indicating the level of difference between the number of experiences for C and the number of its own experiences. The results indicate that the competency of even an honest neighbor degrades as its number of experiences decreases (Figure 1). We also observe that C can effectively evaluate the competency level of advisers even with a limited set of providers. Figure 2 illustrates the experiment in which N_k

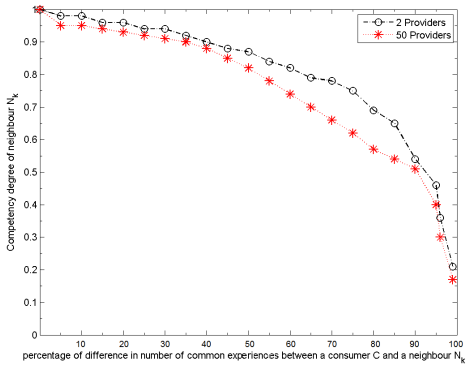


Figure 1: The competency degradation of N_k having different percentages of common experiences

provides different percentages (0% to 100%) of unfair ratings. Given similar conditions as in Figure 1, we observe that as the number of unfair ratings increases, the competency level of N_k decreases. It also indicates that the competency level of N_k drops more significantly if it provides unfair ratings (Figure 2) in comparison with the situation where it has insufficient ratings(Figure 1). Note that, in both experiments, it is noticeable that C can effectively evaluate the competency level of N_k with a few providers - e.g., 2.

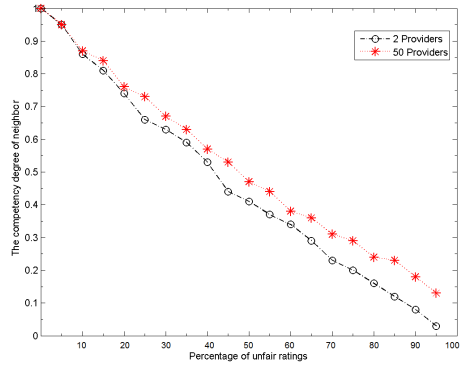


Figure 2: The competency degradation of N_k having different percentages of unfair ratings

The next class of experiment targets the second layer of the filtering algorithm. It involves 80 providers, 4 advisers and 2 consumers. The consumers and the advisers rate 50 randomly selected providers. We assume that the advisers have passed the first layer and are qualified to play the role of advisers. We model the credibility ratings the consumers have of participating advisers and compare them with the actual credibility value of advisers. More explicitly, we examine how the consumers C_1 and C_2 with different dispositions (risk-averse and risk-taking, respectively) evaluate the set of advisers A_1 , A_2 , A_3 and A_4. Note that A_1 and A_3 have a tendency to overrate the providers while A_2 and A_4 have a tendency to underrate the providers. These advisers have different credibility values from 0.0 to 1.0. Also, in order to examine the effect of the recency factor in prediction of the trust value, we assume that A_1 and A_2 provide ratings in the same window with consumers ($T_A - T_C = 0$) while the other advisers provide ratings in different time windows, differing by at most 3 time intervals ($T_A - T_C \leq 3$). Figures 3 and 4 illustrate the trustworthiness of advisers predicted by C_1 and C_2, respectively. Adjusting the threshold values and the forgetting parameter to $\beta = 0.25$, $\epsilon = 0.75$ and $\lambda = 0.9$, we can observe how C_1 and C_2 evaluate the credibility of advisers differently.

As shown in Figure 3, C_1 identifies the behavioral model of advisers and evaluates their credibility adaptively. Results indicate that C_1 assigns higher credibility to the pessimistic adviser A_4 (with $T_{A_4} - T_{C_1} = 3$) when compared with the optimistic adviser A_1 (with $T_{A_1} - T_{C_1} = 0$). Similarly, C_2 considers the old opinion of the optimistic Adviser A_3 more valuable than a recent opinion of pessimistic adviser A_2 (Figure 4).

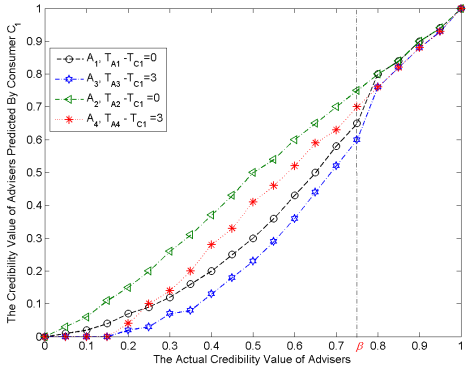


Figure 3: The predicted credibility of advisers by C_1 in comparison with their actual credibility

Table 6 measures the deviation (i.e., Mean-Square-Error and Mean-Absolute Percentage error) between advisers' actual and predicted credibility values determined by C_1 and C_2 across different values of $T_A - T_C$.

To examine how C_1 and C_2 adaptively calculate the coefficients θ and σ, Table 7 depicts the values of these coefficients across various percentages of advisers' dishonesty. That is, the advisers provide different percentages (0% to 100%) of unfair ratings. We observe that consumer agents with different characteristics take different approaches in computing such coefficients, resulting in different evaluations of the credibility degrees of the same advisers.

The final experiment examines the effect of the recency factor $T_{(C,A_k)_{P_j}}$ in evaluating the credibility of advisers.

Table 6: Calculating the error parameters for C_1 and C_2 having various time difference

Agent	Adviser's Pattern	Erorr	$T_A - T_C = 0$	$T_A - T_C = 1$	$T_A - T_C = 2$	$T_A - T_C = 3$	$T_A - T_C = 4$	$T_A - T_C = 5$
C_1	*Optimistic Adviser*	**MSE**	0.06	0.068	0.07	0.079	0.086	0.092
		MAPE	1.43%	1.56%	1.72%	1.58%	2.00%	2.15%
	Pessimistic Adviser	**MSE**	0.048	0.049	0.052	0.057	0.06	0.068
		MAPE	0.58%	0.80%	0.99%	1.22%	1.39%	1.59%
C_2	*Optimistic Adviser*	**MSE**	0.048	0.049	0.052	0.059	0.061	0.067
		MAPE	0.58%	0.8%	0.99%	1.19%	1.39%	1.59%
	Pessimistic Adviser	**MSE**	0.063	0.068	0.073	0.079	0.085	0.092
		MAPE	1.42%	1.57%	1.72%	1.86%	1.99%	2.14%

Table 7: The coefficients parameters calculated by consumers C_1 and C_2

Buyer's Disposition	Coefficient	Percentage of Unfair Ratings										
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Risk-Averse Consumer C_1	θ	0	0.36	0.46	0.56	0.66	0.76	0.86	0.96	1.06	1.16	1.26
	σ	0	0.001	0.002	0.04	0.14	0.24	0.34	0.44	0.54	0.64	0.74
Risk-Taking Consumer C_2	θ	0	0.001	0.002	0.04	0.14	0.24	0.34	0.44	0.54	0.64	0.74
	σ	0	0.36	0.46	0.56	0.66	0.76	0.86	0.96	1.06	1.16	1.26

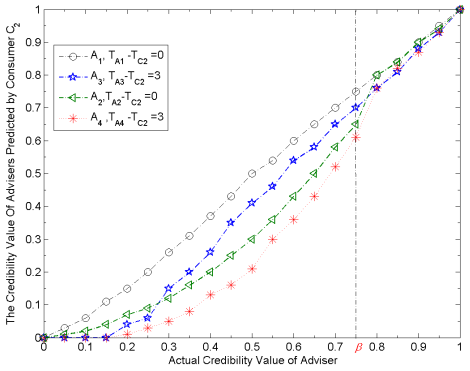


Figure 4: The predicted credibility of advisers by C_2 in comparison with their actual credibility

That is, we define a consumer C and adviser A regardless of the behavioral patterns. We also assume that A has successfully passed the first layer. Adjusting $\beta = 1$ and $\epsilon = 0$, we observe that A with $CR_{(A)} = 0.95$ loses its credibility as the differences between their time window $(T_A - T_C)$ increases. Figure 5 illustrates this by initializing the forgetting factor λ from 0.0 to 1.0.

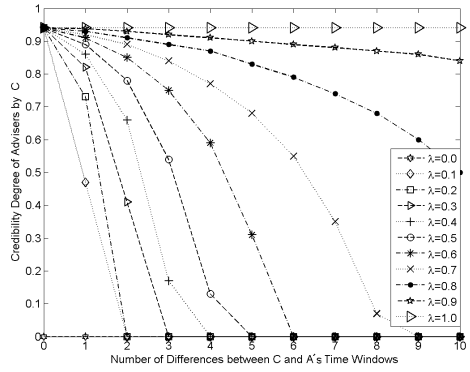


Figure 5: Aging the credibility value of A as time passes

5. RELATED WORK

Several reputation systems and mechanisms have been proposed for modeling the trustworthiness of advisers and coping with the problem of unfair ratings in multi-agent online environments.

In the beta reputation system (BRS) proposed by Jøsang and Ismail [9], which is based on a beta distribution, the agents can only provide binary ratings for each other. He further extends the proposed BRS to adopt a multinomial rating model that computes reputation scores by statistically updating the Dirichlet Probability Density Function (PDF) [10, 7]. In this context, participating agents are allowed to rate each other within any level from a set of predefined rating levels. To handle unfair feedback provided by adviser agents, Whitby et al.[13] use the endogenous discounting method to exclude advisers whose probability distributions of ratings significantly deviate from the overall reputation scores of the target agent. That is, it dynamically determines upper and lower bound thresholds in order to adjust the iterated filtering algorithm's sensitivity tailored to different environmental circumstances. For instance, if the majority of participants act deceitfully in the environment, the lower bound would be set to a higher value so as to increase the sensitivity of the BRS, which can lead to the exclusion of more unfair raters.

Teacy et al. [11] proposed TRAVOS, which is a probabilistic trust and reputation system for agent-based virtual organizations. To derive a measure of trust, this model relies heavily on its direct experiences and refuses to combine others' opinions unless it is not confident about the adequacy of its personal experiences. In such conditions, advisers share the history of their interactions in a tuple that contains the frequency of successful and unsuccessful interaction results. To evaluate the credibility of advisers, it uses a beta distribution and calculates the probability that a particular adviser provides accurate reports given its past opinions and proportionately adjusts the influence of its current observation afterwards.

PeerTrust [14] is a coherent dynamic trust model for peer-to-peer e-commerce communities. To evaluate the quality of the feedback provider, it proposes a *personalized similarity measures* mechanism to compute a feedback similarity rate between the evaluating peer and advising peer over a common set of peers with whom they have had previous interactions. Particularly, this model calculates the root-mean-error or standard deviation of the two feedback vectors to compute the feedback similarity. Through this principle, the evaluating peer discounts the future feedback released by feedback providers.

Yu and Singh[15] have proposed a decentralized reputation management model to locate the rightful advisers in multi-agent systems. In fact, one of the major concerns of this model is detecting malicious agents who deliberately disseminate misinformation through a network. The proposed model considers three types of deceptions: *complementary*, *exaggerative positive* and *exaggerative negative*. It defines

an exaggeration coefficient to differentiate between exaggerative and complementary deceptive agents. This model uses the same credibility measure to calculate the trustworthiness of different kinds of advisers by considering how much their ratings deviate from the actual value experienced by a consumer agent. Note that, in this model, all the advisers have an initial credibility of 1 and as a consumer agent interacts with more provider agents, its credibility will be updated.

Zhang and Cohen [16] proposed a personalized approach for handling unfair ratings in centralized reputation systems. It provides a public and private reputation approach to evaluate the trustworthiness of advisers. In this model, advisers share their subjective opinions over a common set of providers. To estimate the credibility of advisers, it exploits a probabilistic approach and calculates the expected value of advisers' trustworthiness based on their provided ratings.

Our work differs in a number of ways. Unlike other models, which mainly evaluate the credibility of advisers based on the percentage of unfair ratings they provided, this model takes the steps to aggregate several parameters in deriving the trustworthiness of advisers. That is, in addition to the similarity degree of advisers' opinions, we aggregate their behavioral characteristics and evaluate the adequacy of their reputation information in our credibility measure. In this model, every consumer with different behavioral characteristics is able to objectively evaluate the similarity degree of advisers through a multi-criterion rating approach. Also, consumer agents could adaptively predict the trustworthiness of advisers using different credibility measures well-suited for various kinds of advisers.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a two-layered filtering algorithm that cognitively elicits the behavioral characteristics of the participating agents in an e-marketplace. The principles of the two-layer filtering algorithm mainly target malicious agents with complementary rating patterns, agents with insufficient experiences and fraudulent participants who retain a minimum level of trust to cheat opportunistically.

In the first layer, consumer agents take a probabilistic approach and narrow a circle of neighbors by expelling those with significant deceptive patterns, as well as those with an inadequate number of experiences. The basis of the second layer provides mechanisms to cognitively derive the actual intentions of the surviving agents of the previous layer. Here, consumer agents conduct additional evaluations and objectively estimate the similarity degree of advisers through a multi-criterion rating model. Thereafter, they classify their behavioral characteristics based upon their own attitudes. Our model articulates that consumers could have different credibility degrees for the same advisers. Also, it enables consumer agents to include more participants as advisers through a variety of credibility assessment measures. This matter is mostly practical in an environment where the majority of participants are unfair. In order to articulate the effectiveness of our approach in dealing with a community where a majority of participants are unfair, in future work, we will conduct extensive experiments to compare our model with others in identifying honest participants in such situations. Another avenue for future work is to propose a mechanism to dynamically adjust the presented thresholds of the layers based on the environmental conditions and the quality of participants.

7. REFERENCES

[1] K. Barber, Karen Fullam, and Joonoo Kim. Challenges for trust, fraud and deception research in multi-agent systems. In *Trust, Reputation, and Security: Theories and Practice*, volume 2631, pages 167–174. 2003.

[2] Cristiano Castelfranchi, Rino Falcone, and Michele Piunti. Agents with anticipatory behaviors: To be cautious in a risky environment. In *ECAI*, 2006.

[3] Rino Falcone and Cristiano Castelfranchi. Generalizing trust: Inferencing trustworthiness from categories. In *AAMAS-TRUST*, pages 65–80, 2008.

[4] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. An adaptive probabilistic trust model and its evaluation. In *AAMAS (3)*, pages 1485–1488, 2008.

[5] T. D Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[6] Reid Kerr and Robin Cohen. Smart cheaters do prosper: defeating trust and reputation systems. In *AAMAS (2)*, pages 993–1000, 2009.

[7] Zeinab Noorian and Mihaela Ulieru. The state of the art in trust and reputation systems:a framework for comparison. *J. Theor. Appl. Electron. Commer.*, 2010.

[8] Marsh S. Optimism and pessimism in trust. *Proceedings of the Ibero-American Conference on Artificial Inteligence, McGraw-Hill*, 1994.

[9] Audun Jøsang and Roslan Ismail. The Beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

[10] Audun Jøsang and Walter Quattrociocchi. Advanced features in bayesian reputation systems. In *TrustBus*, pages 105–114, 2009.

[11] W. T. L Teacy, J Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Journal of Autonomous Agents and Multi-Agent Systems*, 12(2), 2006.

[12] Yonghong Wang and Munindar P. Singh. Formal trust model for multiagent systems. In *IJCAI*, 2007.

[13] Andrew Whitby, Audun Jøsang, and Jadwiga Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of 7th International Workshop on Trust in Agent Societies*, 2004.

[14] Li Xiong and Ling Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[15] Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 73–80, New York, NY, USA, 2003. ACM.

[16] Jie Zhang and Robin Cohen. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, 7(3), 2008.

Argumentation-based reasoning in agents with varying degrees of trust

Simon Parsons
Brooklyn College
City University of New York

Yuqing Tang
Graduate Center
City University of New York

Elizabeth Sklar
Brooklyn College
City University of New York

Peter McBurney
Department of Informatics
King's College London

Kai Cai
Graduate Center
City University of New York

ABSTRACT

In any group of agents, trust plays an important role. The degree to which agents trust one another will inform what they believe, and, as a result the reasoning that they perform and the conclusions that they come to when that involves information from other agents. In this paper we consider a group of agents with varying degrees of trust of each other, and examine the combinations of trust with the argumentation-based reasoning that they can carry out. The question we seek to answer is "What is the relationship between the trust one agent has in another and the conclusions that it can draw using information from that agent?", and show that there are a range of answers depending upon the way that the agents deal with trust.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence & co-ordination; languages & structures; multiagent systems.*

General Terms

Language, theory.

Keywords

Argumentation; Logic-based approaches and methods; Trust, reliability and reputation.

1. INTRODUCTION

Trust is an approach for measuring and managing the uncertainty about autonomous entities and the information they deal with. As a result trust can play an important role in any decentralized system. As computer systems have become increasingly distributed, and control in those systems has become more decentralized, trust has steadily become more important in computer science [5, 11].

Thus, for example, we see work on trust in peer-to-peer networks, including the EigenTrust algorithm [15] — a variant of PageRank [19] where downloads from a source play the role of outgoing hyperlinks and which is effective in excluding peers who want to disrupt the network — and the work in [1] that prevents peers manipulating their trust values to get preferential downloads. Zhong *et al.* [29] are concerned with slightly different issues in mobile ad-hoc networks, looking to prevent nodes from getting others to transmit their messages while refusing to transmit the messages of others, thus enforcing trustworthy behavior.

The internet, as the largest distributed system of all, is naturally a target of much of the research on trust. There have, for example, been studies on the development of trust in ecommerce [22], on mechanisms to determine which sources to trust when faced with multiple conflicting sources [28], and mechanisms for identifying which individuals to trust based on their past activity [2]. One interesting development is the idea of having individuals indemnify each other by placing some form of financial guarantee on transactions that others enter into [7, 8].

Trust is an especially important issue from the perspective of autonomous agents and multiagent systems [26]. The premise behind the multiagent systems field is that of developing software agents that will work in the interests of their "owners", carrying out their owners' wishes while interacting with other entities. In such interactions, agents will have to reason about the degree to which they should trust those other entities, whether they are trusting those entities to carry out some task, or whether they are trusting those entities to not misuse crucial information. As a result we find much work on trust in agent-based systems [24].

In such work it is common to assume that agents maintain a *trust network* of their acquaintances, which includes ratings of how much those acquaintances are trusted, and how much those acquaintances trust their acquaintances, and so on. An important line of inquiry in this context is what inference is reasonable in such networks, and the propagation of trust and provenance — both the transitivity of trust relations [23, 27] and more complex relationships like "co-citation" [12] have been studied, and in some cases empirically validated [12, 16, 28].

In this paper we look at the use of trust in other aspects of the reasoning that agents carry out. Argumentation [6] is a model of reasoning that seems well-suited to agent-based systems — it is robust against inconsistency, handles decision-making under uncertainty, and supports inter-agent communication. [20] suggests that argumentation is a suitable mechanism for reasoning about trust, and [18] shows how argumentation can be used to track trust in acquaintances. Here we investigate the combination of trust measures on agents and the use of argumentation for reasoning about belief, combining an existing system for reasoning about trust and an existing system of argumentation.

Cite as: Argumentation-based reasoning in agents with varying degrees of trust, S. Parsons, Y. Tang, E. Sklar, P. McBurney and K. Cai, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 879-886.

2. FORMAL MODEL

This paper deals with combining two formal models — a model of trust and a model of argumentation — and we introduce both here. Though there is no standard for either kind of model, we built as generic a model of both trust and argumentation as we could, drawing from well-established models in the literature. As a result we have a combined model that has a number of features unspecified — in later sections we will examine various instantiations.

2.1 Trust

We are interested in a finite set of agents Ags and how these agents trust one another. Following the usual presentation (for example [16, 27, 23]), we start with a *trust relation*:

$$\tau \subseteq Ags \times Ags$$

which identifies which agents trust one another. If $\tau(Ag_i, Ag_j)$, where $Ag_i, Ag_j \in Ags$, then Ag_i trusts Ag_j. This is not a symmetric relation, so it is not necessarily the case that $\tau(Ag_i, Ag_j) \Rightarrow \tau(Ag_j, Ag_i)$. It is natural to represent this trust relation as a directed graph, and we have:

DEFINITION 1. *A* trust network *is a graph comprising, respectively, a set of nodes and a set of edges:*

$$\mathcal{T} = \langle Ags, \{\tau\}\rangle$$

where Ags is a set of agents and $\{\tau\}$ is the set of pairwise trust relations over Ags so that if $\tau(Ag_i, Ag_j)$ is in $\{\tau\}$ then $\{Ag_i, Ag_j\}$ is a directed arc from Ag_i to Ag_j in $\mathcal{T}$.

In this graph, the set of agents is the set of vertices, and the trust relations define the arcs. We are typically interested in *minimal* trust networks, which are connected — these thus capture the relationship between a set of agents all of whom, in one way or another are connected by a "web of trust". A directed path between agents in the trust network implies that one agent indirectly trusts another. For example if:

$$\langle Ag_1, Ag_2, \ldots Ag_n\rangle$$

is a path from agent Ag_1 to Ag_n, then we have:

$$\tau(Ag_1, Ag_2), \tau(Ag_2, Ag_3), \ldots, \tau(Ag_{n-1}, Ag_n)$$

and the path gives us a means to compute the trust that Ag_1 has in Ag_n. Below we will make use of the function $length(\cdot)$ which returns the number of agents in a path: $length(\langle Ag_1, Ag_2, \ldots Ag_n\rangle)$ is n.

The usual assumption in the literature is that we can place some measure on the trust that one agent has in another, so we have:

$$tr : Ags \times Ags \mapsto \Re$$

where tr gives a suitable trust value. In this paper, we take this value to be between 0, indicating no trust, and 1, indicating the greatest possible degree of trust. We assume that tr and τ are mutually consistent, so that:

$$tr(Ag_i, Ag_j) \neq 0 \quad \Leftrightarrow \quad (Ag_i, Ag_j) \in \tau$$
$$tr(Ag_i, Ag_j) = 0 \quad \Leftrightarrow \quad (Ag_i, Ag_j) \notin \tau$$

Now, this just deals with the direct trust relations encoded in τ. It is usual in work on trust to consider performing inference about trust by assuming that trust relations are transitive. This is easily captured in the notion of a trust network:

DEFINITION 2. *If, in the trust network $\mathcal{T}$, Ag_i is connected to Ag_j by a directed path $\langle Ag_i, Ag_{i+1}, \ldots Ag_j\rangle$ then Ag_i* trusts *Ag_j* according to *$\mathcal{T}$*

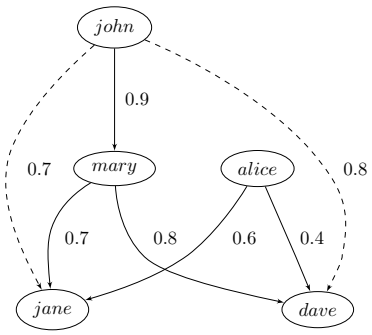


Figure 1: Example trust graph

The notion of trust embodied here is exactly Jøsang's "indirect trust" or "derived trust" [14] and the process of inference is what [12] calls "direct propagation". If we have a function tr, then we can compute:

$$tr(Ag_i, Ag_j) = tr(Ag_i, Ag_{i+1}) \otimes^{tr} tr(Ag_{i+1}, Ag_{i+2}) \otimes^{tr} \ldots \otimes^{tr} tr(Ag_{j-1}, Ag_j) \quad (1)$$

for some operation $\otimes^{tr}$. Here we follow [27] in using the symbol $\otimes$, to stand for this generic operation[1]. Sometimes it is the case that there are two or more paths through the trust network between Ag_i and Ag_j indicating that Ag_i has several opinions about the trustworthiness of Ag_j. If these two paths are

$$\langle Ag_i, Ag'_{i+1}, \ldots Ag_j\rangle \quad \text{and} \quad \langle Ag_i, Ag''_{i+1}, \ldots Ag_j\rangle$$

then the overall degree of trust that Ag_i has in Ag_j is:

$$tr(Ag_i, Ag_j) = tr(Ag_i, Ag_j)' \oplus^{tr} tr(Ag_i, Ag_j)'' \quad (2)$$

Again we use the standard notation $\oplus$ for a function that combines trust measures along two paths [27]. Clearly we can extend this to handle the combination of more than two paths.

Now, given this kind of propagation, we can define an order over the set of agents based on trust values. Since the trust measure we are using is relative to one agent, Ag_i, the order is necessarily relative that agent also. We have:

DEFINITION 3. *For an agent Ag_i, a trust network $\mathcal{T}$ and a trust measure tr, we can define an order over agents $\succeq_i^{tr}$ such that $Ag_j \succeq_i^{tr} Ag_k$ iff $tr(Ag_i, Ag_j) \geq tr(Ag_i, Ag_k)$. If this is the case, we say that Ag_i* considers *Ag_j* at least as trustworthy as *Ag_k*.

We further define $=_i^{tr}$ and $\succ_1^{tr}$ in the usual way. $Ag_j =_i^{tr} Ag_k$ iff $Ag_j \succeq_i^{tr} Ag_k$ and $Ag_k \succeq_i^{tr} Ag_j$. $Ag_j \succ_i^{tr} Ag_k$ iff $Ag_j \succeq_i^{tr} Ag_k$ and $Ag_k \not\succeq_i^{tr} Ag_j$. In addition we extend all these relations to operate over a set of agents: $Ags \succeq_i^{tr} Ags'$ iff Ag_i considers every $Ag \in Ags$ at least as trustworthy as every $Ag' \in Ags'$.

As an example of a trust graph, consider Figure 1 (a) which shows the trust relationship between John, Mary, Alice, Jane and Dave. This is adapted from the example in [16] normalizing the values to lie between 0 and 1. The solid lines are direct trust relationships, the dotted lines are indirect links derived from the direct links. Thus John trusts Jane and Dave because he trusts Mary and Mary trusts Jane and Dave. However, John does not, even indirectly, trust Alice.

[1] [12, 16, 23, 27], among others, provide different possible instantiations of this operation some of which we investigate below.

2.2 Argumentation

From the many formal argumentation systems in the literature, we take as our starting point the system from [21]. An agent $Ag_i \in Ags$ maintains a knowledge base, Σ_i, containing a possibly inconsistent set of formulae of a propositional language $\mathcal{L}$. Agent i also maintains the set of its past utterances, called the "commitment store", CS_i. We refer to this as an agent's "public knowledge", since it contains information that is shared with other agents. In contrast, the contents of Σ_i are "private" to Ag_i.

Note that in the description that follows, we assume that $\vdash$ is the classical inference relation, that $\equiv$ stands for logical equivalence, and we use Δ to denote all the information available to an agent. Thus in an interaction between two agents Ag_i and Ag_j, $\Delta_i = \Sigma_i \cup CS_i \cup CS_j$, so the commitment store CS_i can be loosely thought of as a subset of Δ_i consisting of the assertions that have been made public by Ag_i. In some dialogue games, such as those in [21] anything in CS_i is either in Σ_i or can be derived from it. In other dialogue games, such as those in [4], CS_i may contain things that cannot be derived from Σ_i.

DEFINITION 4. *An* argument A *is a pair* (S, p) *where* p *is a formula of* $\mathcal{L}$ *and* S *a subset of* Δ *such that: (i)* S *is consistent; (ii)* $S \vdash p$*; and (iii)* S *is minimal, so no proper subset of* S *satisfying both (i) and (ii) exists.*

S *is called the* support *of A, written* S *= Support(A) and* p *is the* conclusion *of A, written* p *= Conclusion(A). Thus we talk of* p *being* supported *by the argument* (S, p).

In general, since Δ may be inconsistent, arguments in $\mathcal{A}(\Delta)$, the set of all arguments which can be made from Δ, may conflict, and we make this idea precise with the notion of *undercutting*:

DEFINITION 5. *Let* A_1 *and* A_2 *be arguments in* $\mathcal{A}(\Delta)$. A_1 undercuts A_2 *iff there is some* $\neg p \in Support(A_2)$ *such that* $p \equiv Conclusion(A_1)$.

In other words, an argument is undercut if and only if there is another argument which has as its conclusion the negation of an element of the support for the first argument.

It will be typical for an agent Ag_i to have different degrees of belief $bel_i(\cdot)$ for the formulae in Δ_i, and in this paper we will assume that these belief values (like those in the much of the uncertainty handling literature) are between 0 and 1. Then, if there is some argument $A = (S, p)$ and $A \in \mathcal{A}(\Delta_i)$ we can compute the belief in an argument from the belief in the formulae in the support of the argument:

$$bel_i(A) = bel_i(s_1) \otimes^{bel} bel(s_2) \otimes^{bel} \ldots \otimes^{bel} bel(s_n) \qquad (3)$$

where $S = \{s_1, \ldots, s_n\}$. Where we need to establish the belief in the conclusion p of A we will set $bel_i(p)$ to be $bel_i(A)$. From these values we can then establish an order over arguments.

DEFINITION 6. *For an agent* Ag_i *and a set of belief values for arguments* $bel_i(\cdot)$*, we can define an order over arguments* $\succeq_i^{bel}$ *such that* $A_1 \succeq_i^{bel} A_2$ *iff* $bel_i(A_1) \geq bel_i(A_2)$*. If this is the case, we say that* Ag_i believes A_1 at least as much as A_2.

In addition we say that $A_1 =_i^{bel} A_2$ iff $A_1 \succeq_i^{bel} A_2$ and $A_2 \succeq_i^{bel} A_1$ and $A_1 \succ_i^{bel} A_2$ iff $A_1 \succeq_i^{bel} A_2$ and $A_2 \not\succeq_i^{bel} A_1$. As with the notion of belief on which they are grounded, we will use these relations between the conclusions of arguments when they hold for the arguments themselves.

We can now define the argumentation system we will use:

DEFINITION 7. *An* argumentation system *is a triple:*

$$\langle \mathcal{A}(\Delta_i), Undercut, \succ_i^{arg} \rangle$$

where $\mathcal{A}(\Delta)$ *is as defined as above,* $\succ_i^{arg}$ *is a preference order over arguments, and* $Undercut$ *is a binary relation collecting all pairs of arguments* A_1 *and* A_2 *such that* A_1 *undercuts* A_2.

Note that for now we don't define exactly where $\succ_i^{arg}$ comes from — later we discuss how it can be established from $\succ_i^{bel}$. We say that A_1 is *stronger than* A_2 iff $A_1 \succ_i^{arg} A_2$.

The preference order makes it possible to distinguish different types of relations between arguments:

DEFINITION 8. *Let* A_1*,* A_2 *be two arguments of* $\mathcal{A}(\Delta)$.

- *If* A_2 *undercuts* A_1 *then* A_1 defends itself *against* A_2 *iff* $A_1 \succ_i^{arg} A_2$*. Otherwise,* A_1 does not defend itself.
- *A set of arguments* $\mathcal{A}$ defends A_1 *iff for every* A_2 *that undercuts* A_1*, where* A_1 *does not defend itself against* A_2*, then there is some* $A_3 \in \mathcal{A}$ *such that* A_3 *undercuts* A_2 *and* A_2 *does not defend itself against* A_3.

If A_1 is undercut by A_2 and either does not defend itself, or is not defended by another set of arguments, we say that A_1 is *successfully undercut* and A_2 is a successful undercutter. We write $\mathcal{A}_{Undercut, \succ_i^{arg}}$ to denote the set of all arguments that are not successfully undercut (which includes those that are not undercut at all). The set $\underline{\mathcal{A}}(\Delta)$ of acceptable arguments of the argumentation system $\langle \mathcal{A}(\Delta), Undercut, \succ_i^{arg} \rangle$ is [3] the least fixpoint of a function $\mathcal{F}$:

$$\begin{aligned} \mathcal{A} &\subseteq \mathcal{A}(\Delta) \\ \mathcal{F}(\mathcal{A}) &= \{(S, p) \in \mathcal{A}(\Delta) \mid (S, p) \text{ is defended by } \mathcal{A}\} \end{aligned}$$

DEFINITION 9. *The set of* acceptable *arguments for an argumentation system* $\langle \mathcal{A}(\Delta), Undercut, \succ_i^{arg} \rangle$ *is recursively defined as:*

$$\begin{aligned} \underline{\mathcal{A}}(\Delta) &= \bigcup \mathcal{F}_{i \geq 0}(\emptyset) \\ &= A_{Undercut, \succ_i^{arg}} \cup \left[\bigcup \mathcal{F}_{i \geq 1}(A_{Undercut, \succ_i^{arg}}) \right] \end{aligned}$$

An argument is acceptable *if it is a member of the acceptable set, and a formula is* acceptable *if it is the conclusion of an acceptable argument.*

An acceptable argument is one which is, in some sense, proven since all the arguments which might undermine it are themselves undermined. If there is an acceptable argument for a formula p, then the *status* of p is *accepted*, while if there is not an acceptable argument for p, the status of p is *not accepted*.

3. ARGUMENTATION AND TRUST

In this paper we are concerned with the following question. If an agent makes use of information that it gets from an acquaintance, how should the degree of trust the agent has in its acquaintance inform the way it uses the information? In particular, if an agent constructs arguments using this information, what, in general terms, is it reasonable for the agent to conclude? For example, we might want to specify that if an agent is given information that it doesn't trust very highly, then it should not allow conclusions derived from this information to over-rule conclusions derived from information provided by more trustworthy sources. However it is not immediately clear how to capture principles like this in formal models we introduced above.

3.1 Combining trust and argumentation

To use our models of trust and argumentation to analyze this question, we first need to consider how to combine them. We opt for a very simple approach, adding a trust network to our existing definition of an argumentation system, so that a *trust argumentation system* is:

$$\langle Ags, \mathcal{A}(\Delta_i), Undercut, \succ_i^{arg}, \mathcal{T} \rangle$$

A trust argumentation system, then is specific to a given agent, Ag_i in the system above, and explicitly includes a set of agents Ags that corresponds to the trust network $\mathcal{T}$, and which are the agents whose commitment stores are combined with Σ_i to make up Δ_i.

The argumentation system from the previous section allows Ag_i to construct arguments from:

$$\Delta_i = \Sigma_i \cup \{ \bigcup_{j=1\ldots n} CS_j \}$$

and now, thanks to the trust network, Ag_i can assign a trust value to each of the other agents[2] and hence to their commitment store. In addition, the argumentation model assumes that every formulae in Δ_i can be assigned a belief value, and that there is a preference order $\succ_i^{arg}$ over arguments that identifies the relative strength of arguments.

This model, as introduced, is deliberately vague about a number of issues, allowing us to define a whole family of trust argumentation systems, each of which includes a particular instantiation of the elements we have not specified. First, we need to know what functions to use for $\otimes^{tr}$and $\oplus^{tr}$in order to propagate trust values through the trust network in (1) and (2). Second we need to know how to use the trust value $tr(Ag_i, Ag_j)$ that Ag_i puts on Ag_j to determine the belief that i places in information from CS_j. We can express that as a function $ttb(\cdot)$ such that for some $p \in CS_j$

$$bel_i(p) = ttb(tr(Ag_i, Ag_j)) \quad (4)$$

Third, we need to specify how the belief values $bel_i(\cdot)$ are combined using (3) to establish the belief in an argument from the belief in individual formulae and hence the order $\succ_i^{bel}$. Fourth, we need to know how the preference order $\succ_i^{arg}$, which is used to determine acceptability, is established from $\succ_i^{bel}$.

The main aim of this paper is to explore some of these instantiations — different instantiations will give us different behaviors, and we will use the behaviors to evaluate the instantiations. Before we select instantiations we identify a number of desiderata which we want the instantiated trust argumentation system to adhere to.

3.2 Desirable properties

The properties we use are extracted from the literature, and our aim is to identify which make sense when used in combination with argumentation. Golbeck *et al.* [10] suggests that trust should follow the standard rules on network capacity, so that along any given path the maximum amount of trust between a source and a sink will be no larger than the smallest capacity along the path. In terms of propagating trust through a trust graph, this can be interpreted as saying that the trust that some agent Ag_i has in Ag_j is no greater than the minimum trust value along the path between them:

PROPERTY 1. *If Ag_i is connected to Ag_{i+n} by a directed path $\langle Ag_i, Ag_{i+1}, \ldots, Ag_{i+n} \rangle$ in a trust network where arcs are labelled with values $tr(\cdot)$, then:*

$$tr(Ag_i, Ag_{i+n}) \leq min_{j=0,\ldots n-1} tr(Ag_{i+j}, Ag_{i+j+1})$$

[2]If there is no directed path between the two agents, then the value is 0.

[10] also suggest that the length of the path between two agents is relevant in assessing the trust between the agents, and [13] suggests that "the weakening of trust through long transitive paths should result in a reduced confidence level". We will consider two different ways to interpret this. One says that a longer path will never lead to a stronger trust relation than a shorter path:

PROPERTY 2. *If Ag_i is connected to Ag_j and Ag_k by two directed paths in a trust network, then $tr(Ag_i, Ag_j) \leq tr(Ag_i, Ag_k)$ iff $length(Ag_i, Ag_j) \geq length(Ag_i, Ag_k)$.*

The other interpretation says that trust values are monotonically non-increasing over paths:

PROPERTY 3. *Given the directed path $\langle Ag_i, \ldots Ag_j, \ldots Ag_k \rangle$ then $tr(Ag_i, Ag_k) \leq (Ag_i, Ag_j)$*

The above properties relate to $\otimes^{tr}$. There are also properties relating to $\oplus^{tr}$. The first comes from [13] which suggests that "combination of parallel trust paths should result in an increased confidence level". In other words:

PROPERTY 4. *If Ag_i and Ag_j are linked by two paths in the trust network $\mathcal{T}$, and the trust computed along these paths are $tr(Ag_i, Ag_j)'$ and $tr(Ag_i, Ag_j)''$, then the overall trust of Ag_i in Ag_j,*

$$tr(Ag_i, Ag_j) \geq max\left(tr(Ag_i, Ag_j)', tr(Ag_i, Ag_j)''\right)$$

The authors like to think of this as encoding the idea that having two letters of recommendation for a potential PhD student that say the student is excellent is no worse than having one. However, there is another desideratum that we might enforce here. If we have a potential PhD student with a multitude of recommendation letters that suggest they are a mediocre student, does this make them more highly recommended than a student with just a couple of letters suggesting that they are very good? The authors feel not, and so we also consider the property that combining two parallel trust paths does not cause the overall trust value to exceed the value defined by either path (which is one way to stop the many poor recommendations outweighing a few good ones for a different student).

PROPERTY 5. *If Ag_i and Ag_j are linked by two paths in the trust network $\mathcal{T}$, and the trust computed along these paths are $tr(Ag_i, Ag_j)'$ and $tr(Ag_i, Ag_j)''$, then the overall trust of Ag_i in Ag_j,*

$$tr(Ag_i, Ag_j) \leq max\left(tr(Ag_i, Ag_j)', tr(Ag_i, Ag_j)''\right)$$

In different situations, either of these properties may be appropriate.

We can extend several of these ideas to deal with beliefs and their role in argumentation, in essence placing constraints on the the operation $\otimes^{bel}$. Thinking of an argument as a chain of inferences that make use of formulae from Δ_i then an extension of Property 1 is that the conclusion of an argument should be believed no more than the minimum of the degrees of belief of all of the steps in the argument. This gives us:

PROPERTY 6. *If Ag_i has an argument (S, p), and the support $S = \{s_i, \ldots, s_m\}$, then:*

$$bel_i(p) \leq min_{j=1,\ldots m} bel_i(s_j)$$

We can also extend Properties 2 and 3 to argumentation. This extension suggests that an argument that requires a larger support (and so in some sense is "longer") than another is less believable, and there are two obvious ways that we might capture this:

PROPERTY 7. *If Ag_i has two arguments (S, p) and (S', p'), then $bel_i(p) \leq bel_i(p')$ iff $|S| \geq |S'|$.*

which is analogous to P2 in saying that larger support never means a greater degree of belief, and:

PROPERTY 8. *If Ag_i has two arguments (S, p) and (S', p'), then $bel_i(p) \leq bel_i(p')$ if $S \supseteq S'$.*

which is analogous to P3 in saying that adding additional formulae to a support cannot increase belief and is essentially Loui's [17] "directness" defeater.

The final property that we will consider here deals with the behavior of the combined trust and argumentation system, capturing one reading of the principle we outlined at the start of this section — the strength of an agent's arguments should reflect the trustworthiness of the agents from whom the support of those arguments was obtained. To capture this idea we need first to define:

DEFINITION 10. *Given a set of agents $Ags = \{Ag_1, \ldots Ag_n\}$ where each Ag_j has a commitment store CS_j, then a set of formulae S* corresponds *to the set of agents Ags' iff*

$$Ags' = \{Ag_j | s \in S \text{ and } s \in CS_j\}$$

so that a set of formulae corresponds to the set of agents from whose commitment stores the formulae are drawn. Then we have:

PROPERTY 9. *If Ag_i has two arguments (S, p) and (S', p'), where the supports have corresponding sets of agents Ag and Ag' then (S, p) is stronger than (S', p') only if Ag_i considers Ag to be more trustworthy than Ag'.*

If this property is obeyed, then arguments grounded in information from less trustworthy sources will not be able to defeat arguments whose grounds are drawn from more trustworthy sources. In turn this means that:

PROPOSITION 1. *In a trust argumentation system:*

$$\langle Ags, \mathcal{A}(\Delta_i), Undercut, \succ_i^{arg}, \mathcal{T} \rangle$$

If an argument (S, p), with corresponding set of agents Ag, is acceptable, then, given Property 9, a new argument (S', p') with corresponding set of agents Ag' if Ag_i cannot make (S, p) not acceptable if Ag_i considers Ag' to be less trustworthy than Ag.

PROOF. *If (S, p) is acceptable, then it is not successfully undercut, and so either (i) it is stronger than all its attackers, or (ii) it is defended by arguments that are stronger than those attackers that are stronger than it. Now consider that Ag_i learns enough information to create (S', p') which undercuts (S, p). To make (S, p) not acceptable (S', p') either has to successfully undercut (S, p) or one of (S, p)'s defenders. However, by Property 9, since (S', p')'s corresponding set of agents is less trustworthy than those of (S, p) it is not stronger than (S, p) and so cannot successfully undercut it. Furthermore, since the defenders in (ii) are also stronger than (S, p), (S', p') cannot undercut them either, and so it will fail to make (S, p) not acceptable.* □

This result shows the importance of Property 9 — when it holds, it prevents arguments based on less trustworthy agents from making otherwise acceptable arguments unacceptable, and thus altering what Ag_i takes as being proven.

Note that the desiderata are not independent:

PROPOSITION 2. *Property 2 implies Property 3 and Property 7 implies Property 8.*

PROOF. *P2 requires that given paths from Ag_i to Ag_j and Ag_k, then $tr(Ag_i, Ag_j) \leq tr(Ag_i, Ag_k)$ if and only if $length(Ag_i, Ag_j)$ is greater than or equal to $length(Ag_i, Ag_k)$. If this is the case, then given a path $\langle Ag_i, \ldots Ag_j, \ldots Ag_k \rangle$ it is clear that the path from Ag_i to Ag_k is longer than the path to Ag_j and so $tr(Ag_i, Ag_k)$ will be less than or equal to $tr(Ag_i, Ag_j)$, fulfilling P3.*

Similarly, P7 requires that if Ag_i has two arguments (S, p) and (S', p'), then $bel_i(p) \leq bel_i(p')$ iff $|S| \geq |S'|$. If $S \supseteq S'$ then this will imply that $|S| \geq |S'|$ and hence $bel_i(p) \leq bel_i(p')$, fulfilling P8. □

However these pairs of properties are distinct:

PROPOSITION 3. *Property 3 does not imply Property 2 and Property 8 does not imply Property 7.*

PROOF. *To prove that the first of each of these properties does not imply the second, it suffices to show a single instance where it is not the case. For P3 and P2 we do this by choosing a specific operator for $\otimes^{tr}$. If we use min, then P3 will hold for any assignment of trust values along the path $\langle Ag_i, \ldots Ag_j, \ldots Ag_k \rangle$, for example one with minimum value 0.5. However, with the same operator, we can construct a much longer path where the minimum trust value is 0.8, violating Property 2.*

The counter-example for the second pair of properties is analogous — combining beliefs with min means a small set of support can easily have a smaller belief value than a large set. □

4. TRUST ARGUMENTATION

Having identified a system of trust argumentation and some desiderata for it, in this section we explore its properties.

4.1 Properties of the system

We start by identifying which possible instantiations of the combined trust and argumentation model will satisfy the desiderata in the sense of guaranteeing that the properties will always hold. We begin with Properties 1–3 which depend upon the choice of $\otimes^{tr}$. Two such choices, suggested by Richardson *et al.*[23] are minimum and multiplication. We have:

PROPOSITION 4. *Combining trust values along a path in a trust network according to (1) with minimum or multiplication will satisfy Properties 1 and 3 but not Property 2.*

PROOF. *With associative operations like minimum and multiplication, combining trust values along a path in a trust network is exactly the same as combining a set of trust values. If we combine a set of trust values with minimum, then clearly the resulting value will be exactly the minimum of the values and satisfy Property 1. If we combine two sets of values S_1 and S_2 using minimum, and $S_1 \subseteq S_2$, then the minimum of S_1 will be no smaller than the minimum of S_2, and Property 3 holds. It is equally easy to prove Property 2 does not always hold. If we have two sets S_1 and S_2 and $S_1 \cap S_2 = \emptyset$, then even if S_2 is much larger than S_1, its minimum value can be larger than than that of S_1 — all the values in S_2 could be 0.8 and all those in S_1 could be 0.3.*

Combining a set of values that are no larger than 1 with multiplication will give a value that no larger than any of them, satisfying Property 1. Similarly, if we take the result of multiplying the values in S_1 and then multiply by the values in $S_2 - S_1$ for $S_1 \subseteq S_2$, the value we have won't increase, satisfying Property 3. However, with two unconnected sets S_1 and S_2 there is no necessary relationship between the product of the values in the sets and so Property 2 will not always hold. □

The issue with satisfying Property 2 is that both minimum and multiplication are applied link by link so there is no way to they can meet a criterion that applies to the whole path. If we stretch the definition of computing trust values along a path to allow trust values to be combined by functions that take the whole path as arguments, then we can easily show that:

PROPOSITION 5. *Combining trust values along a path in a trust network in such a way that the trust value is inversely proportional to the length of the path will satisfy Properties 2 and 3 but not Property 1:*

PROOF. *Property 2 requires* $tr(Ag_i, Ag_j) \leq tr(Ag_i, Ag_k)$ *iff* $length(Ag_i, Ag_j) \geq length(Ag_i, Ag_k)$ *which is obviously true for this combination. By Proposition 2, Property 2 implies Property 3, so Property 3 holds as well. The last part of the result is just as easy to show — since the combination depends only on the length of the path, not on the trust values labelling the arcs, there is no reason why the trust along a path should have any particular relationship with those values.* □

The problem with this approach to propagation, and the problem with Property 2, is that it ignores the values of the individual links. As a result it is easy to construct examples which conflict with intuition — a path with very high valued links creates less trust than a marginally shorter path with very low valued links, and any attempt to bring in the values of the links creates situations in which Property 2 can easily be violated.

Now we consider options for $\oplus^{tr}$. Richardson *et al.* [23] suggest maximum and Golbeck *et al.*[10] suggest average[3], while addition seems a suitable dual operation to consider for the options we considered for $\otimes^{tr}$— addition is the dual operation to multiplication for probability theory, and some variants of possibility theory use it as a dual for minimum [9]. Considering all three of these operations, we have:

PROPOSITION 6. *Combining trust values over multiple paths in a trust network according to (2) with maximum satisfies Properties 4 and 5, combining using addition satisfies Property 4 but does not satisfy Property 5, and combining using average satisfies Property 5 but does not satisfy Property 4.*

PROOF. *Since Property 4 specifies that the combination must be greater than or equal to the maximum of the values and Property 5 specifies that it must be less than or equal to the maximum, maximum satisfies both (and will be the only operation to). Adding the two values will clearly give something no smaller than the larger, satisfying Property 4 but won't in general satisfy Property 5 (it will only satisfy it when one value is 0). Average will give something no larger than the larger value, satisfying Property 5, but will only satisfy Property 4 when the values are the same.* □

So addition meets our formulation of Jøsang's property, average obeys the property that we introduced, and maximum meets both.

The third set of properties are those for combining beliefs with $\otimes^{bel}$. In our combined trust and argumentation system, we are assuming that the belief values of propositions in Δ_i are affected by trust values (and we discuss some ways in which this could be achieved below) but to consider the properties, all we assume for now is that there is some distribution of values:

$$m_i : \Delta_i \mapsto [0, 1]$$

from which we can establish a belief value $bel_i(\cdot)$, between 1 and 0, for any formula in Δ_i[4]. These values are then combined to establish beliefs in the conclusions of arguments. Here we consider multiplication and minimum as possible operations for this combination, following the conjunction operations in probability theory and possibility theory respectively [9]. Given Proposition 4 and the origin of Property 1 it is no surprise to find that:

PROPOSITION 7. *Combining belief values according to (3) with minimum or multiplication will satisfy Properties 6 and 8 but not Property 7.*

PROOF. *The proof is the same as for Proposition 4.* □

In order to satisfy Property 7 we need to combine beliefs in a way that depends on the size of the set of support, for example:

PROPOSITION 8. *Consider an argument* $A = (S, p)$ *where* $S = \{s_1, \ldots, s_n\}$*. Setting* $bel(p) = \frac{1}{|S|}$ *will satisfy Properties 7 and 8 but not Property 6.*

PROOF. *The proof is close to that for Proposition 5. The definition of the belief computation means it clearly satisfies Property 7 and by Proposition 2, Property 8 holds as well. The last part of the result is just as easy to show — since the belief in an argument depends only on the size of the support, not on the belief values of formulae in the support, there is no reason why the overall belief should have any particular relationship with the beliefs of the formulae.* □

Thus we have ways of handling trust and belief which will satisfy the various properties we identified, but we have no set of operations that will simultaneously satisfy all the properties.

The final desiderata that we laid down is Property 9, which relates trust values to the conclusions of arguments. To reason about the conditions under which this will hold, we first need to decide how to convert the trust that an agent Ag_i has in agent Ag_j into the belief that Ag_i has in formulae from CS_j. In order to obtain priorities over an agent's knowledge — which is the role played by beliefs in our argumentation — [16] simply imports trust values as the priorities, and here we propose the same method, defining the function ttb from (4) as:

$$ttb(tr(Ag_i, Ag_j)) = tr(Ag_i, Ag_j) \cdot bel\_limit_i$$

where bel_limit_i is a scaling factor that, given belief and trust values are between 0 and 1 limits the maximum belief that a trust value can map to. There are two obvious ways to set this:

L1 $bel\_limit_i = 1$

L2 $bel\_limit_i = min_j\{bel_i(s_j) | s_j \in \Sigma_i\}$

so that we either scale the trust values compared to the maximum possible value for beliefs, so that information with a trust value of 1 is considered as believable as anything, or we scale beliefs so that everything in Σ_i is at least as believable as anything Ag_i is told by another agent.

We also need to determine how $\succ_i^{arg}$ depends on $\succ_i^{bel}$, and there are two obvious ways to do this:

O1 $(S, p) \succ_i^{arg} (S', p')$ iff $(S, p) \succ_i^{bel} (S', p')$

O2 $(S, p) \succ_i^{arg} (S', p')$ iff $(S, p) \succ_i^{bel} (S', p')$ and $Ag \succ_i^{tr} Ag'$ for all Ag corresponding to S and Ag' corresponding to S'.

[3] Average is not usually considered as a binary operation, but it can be expressed in such a form, see, for example [25].

[4] The reason for describing the allocation of belief values in this indirect way is that it is required by some approaches to handling uncertainty, including possibility theeory [9] which we will make use of below.

With these aspects of the model instantiated, we can consider which combinations of the various features of the model satisfy Property 9. We have:

PROPOSITION 9. *A trust argumentation system that uses minimum for $\otimes^{tr}$, maximum for $\oplus^{tr}$, minimum for $\otimes^{bel}$ and adopts L2 and O1 satisfies Property 9.*

PROOF. *Property 9 requires the strength of an argument to be determined by the trust Ag_i has in the corresponding agents so that arguments with less trustworthy corresponding agents are weaker. L2 means that no formulae from any CS_j can be believed more than one from Σ_i, and using minimum to combine belief values means that the strength of any argument will be determined by the trustworthiness of the corresponding agents (a low belief from Σ_i cannot hide an argument's dependency on an untrustworthy agent).* □

Examining the proof, it is clear why we need to have bel_limit_i in the model — without it, there is nothing to stop a highly trusted source supplying information that ends up supporting a weak argument by virtue of another piece of the support which comes from Ag_i itself having a low degree of belief. This, in turn might lead to an argument supported by information from a less trusted source being stronger than an argument based on information from a more trusted source. Exactly this line of reasoning leads us to:

PROPOSITION 10. *A trust argumentation system that that uses minimum for $\otimes^{tr}$, maximum for $\oplus^{tr}$, minimum for $\otimes^{bel}$ and adopts L1 and O1 does not satisfy Property 9 unless $bel(s) = 1$ for every $s \in \Sigma_i$.*

PROOF. *Immediate from the proof of Proposition 9.* □

so not adopting L2[5] doesn't prevent a trust argumentation system meeting our benchmark of performance, Property 9, but means it can only do so under rather restricted circumstances.

Proposition 9 and Proposition 1 tell us that using possibility-style maximum and minimum operations for trust and argumentation — an instantiation of our trust-argumentation system that we will call TA_1 — can guarantee what we have argued is desirable behavior. What about using multiplication, which as we have remarked above, fits more naturally with a probabilistic interpretation of belief? It turns out that:

PROPOSITION 11. *A trust argumentation system that uses minimum for $\otimes^{tr}$, maximum for $\oplus^{tr}$, multiplication for $\otimes^{bel}$ and adopts L2 and O1 does not satisfy Property 9*

PROOF. *Since the result is only that the system does not satisfy the property, a counter example will suffice. Consider all propositions in Σ_i have belief 1. (S, p) includes just one formula that isn't from Σ_i, it comes from CS_j, and $tr(Ag_i, Ag_j) = 0.7$. $bel_i(S, p)$ is thus 0.7. (S', p') includes just two formulae that aren't from Σ_i. These formulae come from CS_k and CS_l, and $tr(Ag_i, Ag_k) = tr(Ag_i, Ag_l) = 0.8$. Thus $bel_i(p') = 0.64$ and the argument is not as strong as the argument which depends on information from a less-trusted source.* □

As the proof shows, the reason that this second trust argumentation system fails to satisfy Property 9 is because multiplying belief values will generate arguments with low beliefs and with O1 determining the order over arguments, this means weak arguments can be generated using information from highly trusted agents. One way to prevent this is to use O2 to determine the order over arguments. We have:

[5]Or, of course, some other mechanism for preventing the kind of interaction between belief and trust sketched in the proof of Proposition 9.

PROPOSITION 12. *A trust argumentation system that uses minimum for $\otimes^{tr}$, maximum for $\oplus^{tr}$, multiplication for $\otimes^{bel}$ and adopts L2 and O2 satisfies Property 9.*

PROOF. *Immediate from the definition of 02.* □

The disadvantage of adopting O2 is that it will only produce a partial order for $\succ_i^{arg}$, and given the role $\succ_i^{arg}$ plays in defining the acceptability, this will affect the reasoning the agents can carry out.

4.2 Trust thresholds

Let's look at one way we can use TA_1. Consider that Ag_i has a *trust threshold* of α, a trust value for agents below which it wishes not to use information from them. If we give arguments whose status is unaffected by information from agents whose trust value is below the threshold α the name *α-safe* then:

PROPOSITION 13. *If Ag_i has a TA_1 argumentation system:*

$$\langle Ags, \mathcal{A}(\Delta_i), Undercut, \succ_i^{arg}, \mathcal{T} \rangle$$

where all formulae in Σ_i have belief value 1, and Ag_i has a trust threshold α, then all arguments with a level of belief above α are α-safe.

PROOF. *Setting the belief of all formulae in Σ_i to 1 ensures that the belief values of arguments directly reflect their trust values making the belief value equal to the threshold easy to establish*[6]*. If an argument A is acceptable, and has a belief value above α, then — as we recall from the proof of Proposition 1 — any undercutters that aren't weaker than A (and so may be below the trust threshold but not affecting the status of A) must, since A is acceptable, be successfully undercut by stronger arguments. Because of the way that trust is converted into belief and belief values are combined with minimum, none of these arguments can be based on information that comes from an agent trusted less than α. So not only A, but all of the arguments that determine its status, must be α-safe.*

If an argument A' is not acceptable and it is above the trust threshold, but was successfully defeated, then that defeat must have been by an argument that is above the trust threshold which (since that defeater is successful) means that in the same way as A, this defeater is α-safe, and hence so is A'. □

This result is helpful because it shows us that for TA_1 information from agents below the trust threshold has limited impact — it won't change the acceptability or otherwise of arguments above the threshold.

5. CONCLUSION

In this paper we presented a formal model that provides a simple combination of argumentation and trust. We examined some of the properties of different instantiations of the model, and showed that the system we called TA_1 has the ability to ensure that arguments grounded in information from untrustworthy agents cannot overrule arguments grounded by more trustworthy agents and under certain conditions can deal with trust thresholds.

This work is distinct from, and complementary to, other existing work on trust and argumentation. The work of Matt *et al.* [18] for example looks at constructing arguments *for* trusting other agents — it is a way to compute the tr values that we assume. In contrast, here we are concerned with computing arguments *with* trust. Similar remarks hold for [20] which looks to construct arguments about the trust that one agent has in another.

[6]The proof can be altered to deal with formulae in Σ_i having smaller belief values, it would mean replacing the trust threshold in the proof with $\alpha.min_j\{bel_i(s_j)|s_j \in \Sigma_i\}$

Though the system we define is simple, there is more to say about it. Our future work will address aspects of the system that we have not had space to discuss here. We are working on a more extensive analysis of operators for the trust argumentation systems, as well as expanding the notion of trust threshold to what we call the *trust budget* — if an agent is prepared to tolerate a certain overall amount of distrust in all the information it uses in all of its arguments, how does this affect what it finds acceptable? Other topics of interest are combining what we have here with the use of argumentation to establish trust values, and the use of more complex methods of representing trust than the simple numerical approach we adopt here.

Acknowledgement

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

We thank the reviewers for their helpful comments.

6. REFERENCES

[1] Z. Abrams, R. McGrew, and S. Plotkin. Keeping peers honest in eigentrust. In *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[2] B. T. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *Proceedings of the 16th International World Wide Web Conference*, pages 261–270, Banff, Alberta, May 2007.

[3] L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation framework. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

[4] L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In *Proceedings of the Fourteenth European Conference on Artificial Intelligence*, 2000.

[5] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Web Semantics*, 5(2):58–71, June 2007.

[6] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.

[7] P. Dandekar, A. Goel, R. Govindan, and I. Post. Liquidity in credit networks: A little trust goes a long way. Technical report, Department of Management Science and Engineering, Stanford University, 2010.

[8] D. B. DeFigueiredo and E. T. Barr. TrustDavis: A non-explotable online reputation system. In *Proceedings of the 7th IEEE International Conference on E-Commerce Technology*, 2005.

[9] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, NY, 1988.

[10] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. In *Proceedings of the 7th International Workshop on Cooperative Information Agents*, Helsinki, August 2003.

[11] T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 4(4):2–16, 2000.

[12] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on the World Wide Web*, 2004.

[13] A. Jøsang, E. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[14] A. Jøsang, C. Keser, and T. Dimitrakos. Can we manage trust? In *Proceedings of the 3rd International Conference on Trust Management*, Paris, May 2005.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th World Wide Web Conference*, May 2004.

[16] Y. Katz and J. Golbeck. Social network-based trust in prioritzed default logic. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[17] R. P. Loui. Defeat among arguments: a system of defeasible inference. *Computational Intelligence*, 3(3):100–106, 1987.

[18] P.-A. Matt, M. Morge, and F. Toni. Combining statistics and arguments to compute trust. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagents Systems*, 2010.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical Report 1999-66, Stanford InfoLab, 1999.

[20] S. Parsons, P. McBurney, and E. Sklar. Reasoning about trust using argumentation: A position paper. In *Proceedings of the Workshop on Argumentation in Multiagent Systems*, Toronto, Canada, May 2010.

[21] S. Parsons, M. Wooldridge, and L. Amgoud. On the outcomes of formal inter-agent dialogues. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[22] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. In M. R. Baye, editor, *The Economics of the Internet and E-Commerce*, pages 127–157. Elsevier Science, Amsterdam, 2002.

[23] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the 2nd International Semantic Web Conference*, 2003.

[24] J. Sabater and C. Sierra. Review on computational trustand reputation models. *AI Review*, 23(1):33–60, September 2005.

[25] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[26] W. T. L. Teacy, G. Chalkiadakis, A. Rogers, and N. R. Jennings. Sequential decision making with untrustworthy service providers. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[27] Y. Wang and M. P. Singh. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[28] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the Conference on Knowledge and Data Discovery*, 2007.

[29] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003.

A Particle Filter for Bid Estimation in Ad Auctions with Periodic Ranking Observations

David Pardoe and Peter Stone
Department of Computer Science
The University of Texas at Austin
{dpardoe, pstone}@cs.utexas.edu

ABSTRACT

Keyword auctions are becoming increasingly important in today's electronic marketplaces. One of their most challenging aspects is the limited amount of information revealed about other advertisers. In this paper, we present a particle filter that can be used to estimate the bids of other advertisers given a periodic ranking of their bids. This particle filter makes use of models of the bidding behavior of other advertisers, and so we also show how such models can be learned from past bidding data. In experiments in the Ad Auction scenario of the Trading Agent Competition, the combination of this particle filter and bidder modeling outperforms all other bid estimation methods tested.

Categories and Subject Descriptors

I.2 [**Computing Methods**]: Artificial Intelligence

General Terms

Algorithms, Experimentation, Economics

Keywords

agent modeling, learning, particle filters, trading agents, sponsored search, ad auctions

1. INTRODUCTION

Sponsored search [5] is one of the most important forms of Internet advertising available to businesses today. In sponsored search, an advertiser pays to have its advertisement displayed alongside search engine results whenever a user searches for a specific keyword or set of keywords. An advertiser can thereby target only those users who might be interested in the advertiser's products. Each of the major search engines (Google, Yahoo, and Microsoft) implements sponsored search in a slightly different way, but the overall idea is the same. For each keyword, a *keyword auction* [6] is run in which advertisers bid an amount that they are willing to pay each time their ad is clicked, and the order in which the ads are displayed is determined by the ranking of the bids (and possibly other factors). Having an ad in a higher position is generally considered to be more desirable.

Cite as: A Particle Filter for Bid Estimation in Ad Auctions with Periodic Ranking Observations, David Pardoe and Peter Stone, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 887-894.

Running a successful keyword advertising campaign can be difficult. An advertiser must choose the keywords of interest and its bids for each one based on an understanding of customer behavior, competitors' bidding patterns, and its own advertising constraints and needs, all of which can change over time. Complicating matters is the fact that advertisers receive very limited information about the actions taken by other advertisers. In particular, advertisers do not see the bids of other advertisers. Knowing the bids of other advertisers for a specific keyword would allow an advertiser to predict the ad position and cost per click for any amount it bid and use this information to choose the bid it expected to maximize profit. Search engines typically release some information concerning the position that an advertiser could expect for certain bids, but this information is generally incomplete and out of date. Alternately, an advertiser could experiment with different bids and observe the resulting positions, but such experimentation would be time consuming and costly.

In this paper, we present a particle filtering approach to estimating the bids of other advertisers in a single keyword auction. This particle filter relies on periodic observations of the rankings of all advertisers. In addition, it requires models of the bidding behavior of other advertisers, and we show how such models can be learned. We implement and test our particle filter in the context of the Ad Auction scenario of the Trading Agent Competition (TAC/AA) [3], a competition developed in 2009 to encourage research into keyword auction bidding within a carefully designed simulated environment. Nevertheless, the basic approach to particle filtering described here should generalize to any standard ad auction setting.

The remainder of this paper is organized as follows. After formally specifying the auction setting we consider in Section 2, we present the design of the particle filter in Section 3. Section 4 describes our experimental domain, TAC/AA, and explains how our particle filter can be applied in this domain. Our particle filter requires bid transition models for all other advertisers, and in Section 5, we present a machine learning approach to building these models. Finally, Section 6 contains experimental results comparing the accuracy of our particle filter to other bid estimation methods in three different TAC/AA settings.

2. AUCTION SETTING

We begin by formally specifying our auction setting, which

has been chosen to be as general as possible while still capturing the basic elements of a keyword auction. We are interested in estimating the bids of N other advertisers for a single keyword for which we are advertising. Each advertiser has a standing bid indicating the amount it is willing to pay each time its ad is clicked, and this bid may occasionally be revised. This bid must be above a known reserve price *reserve*. When a user searches for the keyword, the advertisers' bids are ranked in descending order, and their ads are shown in this order. If more than M bids are above the reserve, then only the top M ads are shown. When a user clicks on our ad, our *cost per click* (*cpc*) is the minimum amount we could have bid and still had our ad shown in its current position. In other words, our *cpc* is equal to the amount of the bid ranked below ours, or to *reserve*. At some regular interval, we receive a report containing the following information: i) the bid ranking at time t (for advertisers whose ads are being shown, i.e., at most the top M), and ii) our own *cpc* at time t. Our goal is to estimate the bids of the other advertisers at time t. Depending on the nature of the auction, advertisers may be able to revise their bids more frequently than this reporting interval; however, we will only attempt to estimate bids at this interval, and our models of advertiser behavior will only model changes in bids at this interval (e.g., from time $t-1$ to time t).

As this model is an abstraction of the keyword auctions used in the real world, there are a number of complicating factors it does not include, but we believe it to be a useful model for study. One issue faced in real keyword auctions is that ad positions are often not determined by bid rank alone, but by a combination of bid rank and other factors such as clickthrough rate. This is not a problem, however, as in this case we could simply attempt to estimate the amount we would need to bid to achieve a higher position than each other advertiser, instead of the true bid of each advertiser, and use the same particle filtering approach.

A larger concern is that search engines do not actually provide advertisers with periodic reports of the bid rankings of other advertisers. Of course, it is possible to simply repeatedly search for the keyword and observe the order of the ads displayed, but for a large advertising campaign this process would need to be automated (using a "screen scraper"), and search engines generally take measures to prevent this type of activity. Nevertheless, a number of services offer to collect this type of information for subscribers, so the assumption of these periodic reports is not necessarily unrealistic.

Finally, we note that this auction setting is in fact an instance of a repeated generalized second price auction, and that our particle filter could be applied to any such auction given periodic ranking observations. Generalized second price auctions are most commonly used in keyword auctions, but they have been considered in other areas such as electricity auctions [10].

3. PARTICLE FILTER

We now describe our particle filter for estimating the bids of other advertisers given periodic reports. For now, we assume that we have a model of each advertiser that gives us a probability distribution over their next bid given a history of their estimated bids and rankings. Developing these models will be the subject of Section 5. Again, we emphasize that we only concern ourselves with the bids at the reporting interval — by "next bid" we mean the bid at the time of the next report, and likewise our history only reflects the auction state at the times of past reports.

Given these advertiser models and reports, we estimate the joint distribution over the bids of all other advertisers using a particle filter. A particle filter is a sequential Monte Carlo method that tracks the changing state of a system by using a set of weighted samples (called particles) to estimate a posterior density function over the possible states. The weight of each particle represents its relative probability, and particles and weights are revised each time an observation (conditioned on the current state) is received. In this case, the reports represent our observations, and each particle represents an estimate of the bids of all advertisers at the time of the last report. Additionally, each particle stores all of its past bid estimates, so each particle can be seen as a full bidding history of all advertisers. Particle filters are a fitting solution to this problem because they require no assumptions about the types of distributions involved (unlike Kalman filters), they can be used efficiently in high-dimensional spaces (unlike grid-based methods that discretize the state space), and particles are a convenient data structure for storing bidding histories. We estimate the joint distribution over bids instead of estimating each advertiser's bid independently due to the fact that our estimate for each advertiser is completely dependent on our estimate for all other advertisers. (In Section 6.2 we describe a method of estimating bids independently, but this method relies on several unrealistic simplifying assumptions.)

For the experiments of this paper, the implementation of our particle filter makes use of a discretized set of bids b^1 ... b^B, and so we describe our particle filter in terms of discrete probability distributions over these bids; however, continuous probability distributions could also be used in our particle filter if they can be dealt with analytically.

3.1 SIS Particle Filter

The simplest particle filter, and the one from which more complicated variations are derived, is the *Sequential Importance Sampling* (SIS) filter [1]. A SIS filter can be implemented for our bid estimation problem as follows. Each particle p contains a current estimate for the bids of all N advertisers, as well as bid estimates for each past time step. An initial set of particles P is chosen to reflect a possible distribution over bids when no reports have yet been received — essentially our prior. The number of particles $|P|$ should be chosen to give an acceptable tradeoff between accuracy and speed. Each particle p receives initial weight $w_p = 1/|P|$. Each time we receive a report, we update P by generating and weighting a new set of particles. For each existing particle p, we sample a new particle p' (i.e., we copy the bidding history contained in p and then sample a new set of current bids). Finally, we reweight the particles.

The sampling and weighting procedures depend on our choice of *proposal distribution* from which we sample new particles: $\pi(p'|p, report)$. π may be any distribution we choose. The weighting procedure then follows from the choice of π such that the set of weighted particles approximate the true posterior distribution. If particle p had weight w_p, then particle p' receives weight

$$w_{p'} = w_p \frac{Pr(report|p')Pr(p'|p)}{\pi(p'|p, report)} \quad (1)$$

Finally, the weights of all new particles are normalized so

that they sum to one.

3.2 Choice of Proposal Distribution

The choice of proposal distribution can significantly affect the performance of the particle filter. The distribution $\pi(p'|p, report) = Pr(p'|p, report)$ is what is known as the *optimal proposal distribution* and results in a weighting of $w_{p'} = w_p Pr(report|p)$. This proposal distribution is called optimal because it results in the least variance between particle weights — $w_{p'}$ is independent of p', and so it will be the same regardless of which p' is sampled. However, the optimal proposal distribution is often not used in practice because it can be difficult to sample from this distribution and perform weight calculations. Instead, the proposal distribution that would typically be used is $\pi(p'|p, report) = Pr(p'|p)$. The resulting weighting is $w_{p'} = w_p Pr(report|p')$.

The typical proposal distribution is indeed much easier to work with in our bid estimation problem, but is has a serious flaw: $Pr(report|p')$ may frequently be zero. If too few particles receive any weight, then the filter may eventually become degenerate, with mostly identical particles. To see why $Pr(report|p')$ might be zero, recall that the report contains a ranking and our *cpc*. If the current bids represented by p' are inconsistent with this ranking, then the likelihood of p' will be zero. The fraction of inconsistent particles will depend on advertiser behavior; in the worst case of random bids, only $1/N!$ of the particles would be expected to be consistent with the ranking, as any of the $N!$ possible rankings would be equally likely. Even in less extreme cases, we would still expect there to be occasional improbable rankings. Furthermore, even if p' is consistent with the rankings, it will likely not be consistent with our observed *cpc*.

We therefore use the optimal proposal distribution in our particle filter. Particles drawn from this distribution are guaranteed to be consistent with the report. Thus, we need methods of sampling from $Pr(p'|p, report)$ and computing $Pr(report|p)$. These methods are described in the following two subsections.

3.3 Computing Pr(report | p)

For $n \in 1...N$, let a_n be the advertiser ranked nth, excluding ourselves (i.e., lower ranked advertisers have their rank increased by one). Unranked advertisers may be assigned to the remaining a values (those representing the lowest ranks) arbitrarily. For a given set of current bid estimates, let c_n indicate that a_n's bid is consistent with (i.e., not higher than) the bids of advertisers a_1 ... a_{n-1} and with our own *bid* and *cpc*. Then $Pr(report|p) = Pr(c_1 \cap c_2 \cap ... \cap c_N|p)$. That is, the probability of particle p from the previous time step leading to a new particle consistent with *report* is equal to the probability that for each other advertisers, that advertiser's new bid estimate does not exceed the bid estimate of a higher ranked advertiser or conflict with *bid* or *cpc*. Furthermore, $Pr(c_1 \cap c_2 \cap ... \cap c_N|p) = Pr(c_N|p, c_1 ... \cap c_{N-1}) \cdot ... \cdot Pr(c_2|p, c_1) Pr(c_1|p)$. Below, we show how each of these N probabilities can be computed.

For each $n \in 1$ ... N, we would like to compute $Pr(c_n|p, c_1$ ... $c_{n-1})$, that is, the probability that particle p leads to a new bid for advertiser a_n that is consistent with our *bid* and *cpc* and the new bids of advertisers a_1 ... a_{n-1}, *given that these bids are already known to be consistent*. This probability is computed differently for each of five different cases. Let f_n be the probability mass function for a_n's next bid given the information in p, as determined by our advertiser model for a_n. In each case, we will determine f'_n, the probability mass function for a_n's next bid given p *and* c_1 ... c_{n-1}, as well as the corresponding cumulative distribution function F'_n giving the probability that the new bid is less than (but *not* equal to, as is usual in a CDF) a given value. We begin by setting F'_0 to be 0 everywhere.

- **Case 1:** *a_n has a higher rank than us.* Because the advertiser is ranked, its bid will be consistent with the bids of a_1 ... a_{n-1} so long as its bid is no greater than the bid of a_{n-1}. Because the advertiser is ranked above us, its bid must be no less than *bid*. Therefore,

$$Pr(c_n|p, c_1...c_{n-1}) = \sum_{x=bid}^{b^B} f_n(x)[1 - F'_{n-1}(x)] \quad (2)$$

  Similarly, we can define

$$f'_n(x) = f_n(x)[1 - F'_{n-1}(x)]Z \quad (3)$$

  where f'_n has support between *bid* and b^B and Z is a normalizing constant.

- **Case 2:** *a_n is ranked one below us.* Our *cpc* is determined by the advertiser ranked below us, so we know the bid of a_n.

$$Pr(c_n|p, c_1...c_{n-1}) = f_n(cpc) \quad (4)$$

  and we define F'_n to be 0 at or below *cpc* and 1 elsewhere.

- **Case 3:** *a_n is ranked at least two below us.* As in Case 1, we need the bid of a_n to be no greater than the bid of a_{n-1}. Because the advertiser is ranked below us, its bid must be between $reserve$ and *cpc*. Therefore,

$$Pr(c_n|p, c_1...c_{n-1}) = \sum_{x=reserve}^{cpc} f_n(x)[1 - F'_{n-1}(x)] \quad (5)$$

  and

$$f'_n(x) = f_n(x)[1 - F'_{n-1}(x)]Z \quad (6)$$

  where f'_n has support between $reserve$ and *cpc*.

- **Case 4:** *a_n is unranked and there are M ranked advertisers.* Because the maximum of M advertisers were ranked, we do not know if a_n placed a bid or not. We only know that a_n's bid is no greater than the bid of a_k, where a_k is the advertiser ranked M.

$$Pr(c_n|p, c_1...c_{n-1}) = \sum_{x=0}^{cpc} f_n(x)[1 - F'_k(x)] \quad (7)$$

  and

$$f'_n(x) = f_n(x)[1 - F'_k(x)]Z \quad (8)$$

  where f'_n has support between 0 and *cpc* .

- **Case 5:** *a_n is unranked and there are fewer than M ranked advertisers.* a_n did not bid or else it would have been ranked. We treat any non-bid (or bid below the reserve) as a bid of 0, so

$$Pr(c_n|p, c_1...c_{n-1}) = F_n(0) \quad (9)$$

  and $f'_n(0) = 1$.

By proceeding through the advertisers in order, we can determine $Pr(c_n|p, c_1 \ ... \ c_{n-1})$ for each $n \in 1 \ ... \ N$ and take the product to get $Pr(report|p)$. We repeat this process for each $p \in P$ and normalize the results to obtain the distribution from which we sample when generating new particles.

3.4 Sampling from Pr(p′ | p, report)

Now given a particle p and the report, we would like to sample a new particle p'. This involves choosing a new bid b_n for each advertiser a_n, and so $Pr(p'|p, report) = Pr(b_1 \cap b_2 \cap \ ... \ b_N|p, report) = Pr(b_1|p, report, b_2 \ ... \ b_N) \cdot \ ... \ \cdot Pr(b_N|p, report)$.

Observe that for advertiser a_N, the function f'_N generated above is in fact the same as $Pr(b_N|p, report)$ because it represented the distribution over b_N given that the bids of all other advertisers were consistent with *report*. For any other advertiser a_n, if bids $b_{n+1}...b_N$ are known, then we can compute $Pr(b_n|p, report, b_{n+1}...b_N)$ by taking the highest bid of any lower ranked advertiser (if any) and normalizing the portion of f'_n above that bid. Thus, by starting with b_N and working backwards, we can sample all bids in such a way that the bids are consistent with *report* and the probability of the resulting particle p' is $Pr(p'|p, report)$.

3.5 Example

We now use an example to illustrate particle filters using both the typical and optimal proposal distributions. Suppose that there are two advertisers x and y in addition to ourselves, and that according to our advertiser models, at each time step each either increases or decrease its bid by 1, with probability 0.5 in each case. We receive a report for time $t + 1$ indicating that y had the highest bid, x had the second bid, and we had the lowest bid of 0.25. Now consider a particle p that has the following bid estimates for time t: $b_x = 2$ and $b_y = 1.5$.

For the typical proposal distribution, to sample a new particle p' reflecting time $t + 1$ we would sample new bids for each bidder according to our advertiser models. However, of the four possible outcomes, only one, $b_x = 1$ and $b_y = 2.5$, is consistent with the bid ranking. If we sampled a different set of bids for p', then the weight of p' would be set to zero.

For the optimal proposal distribution, we let $a_1 = y$ and $a_2 = x$ and follow the procedure described above. First, we determine $Pr(report|p)$. We have $f_1(0.5) = f_1(2.5) = 0.5$ and $f_2(1) = f_2(3) = 0.5$, with both functions zero elsewhere. For a_1, we follow Case 1. $Pr(c_1|p) = 1$ and $f'_1 = f_1$ because F'_0 is zero and either possible bid is above our bid of 0.25. For a_2, we follow Case 1 again. $Pr(c_2|p, c_1) = 0.25$ and $f'_2(1) = 1$, because $1 - F'_1(3) = 0$, $1 - F'_1(1) = 0.5$, and either possible bid is above our bid of 0.25. Thus $Pr(report|p) = Pr(c_2|p, c_1)Pr(c_1|p) = 0.25$, which we know is correct.

To sample a new particle p', we first sample b_2 from f'_2 and get 1, the only possibility. Then we sample b_1 from the portion of f'_1 that is above 1, and we get 2.5, again the only possibility. So we are guaranteed to sample $b_1 = 2.5$ and $b_2 = 1$, the only possibility for p' given *report*.

3.6 Resampling

This section has described an implementation of an SIS particle filter using the optimal proposal distribution. A commonly used extension of an SIS filter is the *Sampling Importance Resampling* (SIR) filter, which occasionally resamples the set of particles to prevent the weights of some particles from approaching zero. Our implemented particle filter is an SIR filter, and we resample the particles in P after each update by replacing P with $|P|$ particles sampled (with replacement) according to the weights, then setting all weights to $1/|P|$.

4. TAC/AA

We now briefly describe the experimental domain in which we test our particle filter, TAC/AA [3]. For full details, see the game specification [2]. In each TAC/AA game, eight agents compete as advertisers to see who can make the most profit from selling a limited range of home entertainment products over 60 simulated game days, each lasting 10 seconds. Products are classified by manufacturer (3) and by component (3) for a total of nine products. Search engine users, the potential customers, submit queries consisting of a manufacturer and a component, although either or both may be missing. There are thus 16 total query types. Each day, for each of the 16 query types, a keyword auction is run. For each auction, an advertiser submits i) a (real, non-negative) bid indicating the amount it is willing to pay per click, and ii) a daily spending limit (optional). The top five bidders have their ads shown in order, but if an advertiser hits its spending limit (as a result of having its ad clicked enough times), its ad is not shown for the rest of the day, and all advertisers with lower bids have their ads move up one position. Bids must exceed a small reserve price. For each query type, advertisers receive a daily report providing limited information about the results of their actions and the actions of other advertisers. Reports include the advertiser's average cpc and the average position of each other advertiser. Note that these positions, and thus an advertiser's cpc, can change throughout the day due to spending limits.

TAC/AA differs from the auction model described in Section 2 in a number of ways. First, the daily reports provide the average positions of other advertisers instead of a ranking of their bids. Fortunately, it is possible to transform average positions into bid rankings with fairly high accuracy as described in [8]. Second, in TAC/AA the reserve price is unknown, but we can obtain a reasonably accurate estimate and use this in our particle filter. Third, we are only given an average cpc. If the agent ranked one spot below us hits its spending limit before we do, the average cpc will not equal the bid of that agent, as was assumed in Case 2 above. Once again, we can use the reported average positions to determine if this was the case, and if so we can apply Case 3 instead for that advertiser. Finally, as mentioned in Section 2, in TAC/AA ad positions are determined by a combination of bid rankings and clickthrough rates. In our experiments, we address this issue by adjusting each bid of each other advertiser to be the amount we would have needed to bid to achieve a higher position than that advertiser.

The use of spending limits in general represents another significant difference. We avoid dealing with spending limits by using our particle filter to estimate each advertiser's bid at the start of the day, before spending limits cause any advertiser to drop out of the bidding. Estimating the spending limits of other advertisers can be treated as a separate problem, as in [7].

The application of our particle filter to a single query type in a TAC/AA game can therefore be summarized as follows.

On each day d, we receive a report that includes our average cpc and the average position of all advertisers on day $d-1$. We transform the average positions into the bid rankings, and we determine whether our average cpc does in fact equal the bid of the advertiser ranked below us. Then, using this information, we update our particle filter, and the result is an estimated distribution over the bids of all advertisers at the start of day $d-1$.

In our experiments, without loss of generality we consider only the nine query types in which both a manufacturer and component are specified. In any one game, a number of random factors affect the value of advertising for any particular query type, and thus the bidding behavior of the agents. The distributions from which these factors are drawn are the same for all nine query types, however, and so this bidding behavior is the same in expectation for any query type in any game. As our particle filter is designed for a single keyword auction, in our experiments we treat each game as if it provides us with nine independent and identically distributed episodes, each representing a 60-day bidding history for a single keyword.

5. ADVERTISER MODELS

In Section 3, we assumed that we had a bidding model for each advertiser so that we could determine the distribution over the advertiser's next bid given its bid history. We now describe a method of generating such a model using machine learning. While the details of this section are specific to TAC/AA, the general approach could be used in any situation in which sufficient bidding data is available for use in learning. Note that precise knowledge of bids, as is available here, is not necessary to be able to build and make use of bidder models. Estimates based on information released by search engines could be used, and it might be possible to bootstrap by alternating model building and particle filtering stages to obtain increasingly accurate estimates.

The problem we are trying to solve is a conditional density estimation problem. While a number of parametric approaches to solving these problems exist, we choose to use a nonparametric approach. The bidding behavior of advertisers can be quite complex, and we would prefer to make as few assumptions about this behavior as possible. Methods of nonparametric conditional density estimation have been used in previous TAC domains to solve problems such as predicting future hotel prices [9] and predicting the probability of an offer to a customer resulting in an order [4]. The approach we take is to learn a model that takes as input both a bid amount b and a set of features representing the current state, and outputs the probability that the advertiser's next bid is less than or equal to b. Thus by evaluating this model for different values of b, we can build the cumulative distribution function for the advertiser's next bid for any given state. This approach is similar to the one used in [9] except that rather than including a price as an input to the model, there the space of prices is discretized and the model outputs a separate probability prediction for each price.

For a given advertiser, we assume that we have access to the logs from a number of TAC/AA games in which both that advertiser and our own agent participated. From these logs, we can determine the actual bids of the advertiser as well as the reports that would have been available to our own agent at any point in time. For each day $d > 0$ and any given bid b (for which we wish to make a prediction) we generate a feature vector containing the following:

- b,
- d,
- the last five bids: b_{d-1} ... b_{d-5},
- five bid differences: $b - b_{d-1}$ ... $b - b_{d-5}$,
- the last average position: ap_{d-1},
- five average position differences: $ap_{d-1} - ap_{d-2}$ ... $ap_{d-1} - ap_{d-6}$,
- the maximum and minimum bids so far: max and min,
- the differences $b - max$ and $b - min$,
- the maximum and minimum bids over the last ten days: max_{10} and min_{10}, and
- the differences $b - max_{10}$ and $b - min_{10}$

Any reference to a day before the first day is replaced with the corresponding reference to the first day. Finally, each vector is labeled with a 1 or 0 to indicate whether the advertiser's bid on day d was less than or equal to b. We note that a five-day history is used because five-day cycles can be observed in the bid series of some advertisers (for reasons specific to the TAC/AA rules). In real auctions, a 24-hour or 7-day cycle might be more likely to occur, and an appropriate bid history could be used.

Observe that any choice of b results in a unique feature vector. To generate a set of training data from game logs, we need to choose one or more values of b to use for each bid observed. If on day d the advertiser's bid was b_d, we generate 14 training instances by using 14 different values of b. The first two values are b_d and $b_d + 0.01$. The next two values are 0 and $\hat{b}$, where $\hat{b}$ is 1.1 times the highest bid ever observed for the advertiser. Next, we divide the interval $[0, b_d]$ in fifths and choose one bid uniformly randomly from each fifth. Finally, we do the same with the interval $[b_d + 0.01, \hat{b}]$. These choices give good coverage of the range of possible bids. The number 14 was chosen to give a reasonable tradeoff between model accuracy and keeping the size of the training set manageable.

Now that we have a training set, we need to choose a learning algorithm to build our model. We experimented with the learning algorithms available in the WEKA machine learning toolkit [11] and found that M5P model trees gave the best performance both in terms of probability prediction accuracy on the data set and bid estimation accuracy of the complete particle filter. Note that our learning problem can be treated as either a regression problem (treating the probability as a number to predict) or a binary classification problem (predicting the probability of belonging to the class '1'), and so both types of algorithms were tested.

One final issue that must be dealt with is the fact that we wish to use our model to produce a cumulative distribution function, but the output of our model may not in fact satisfy the requirements. For a given state, as the bid b increases from 0 to $\hat{b}$, the output of our model should monotonically increase and reach a maximum of 1, but this will sometimes not be the case. We address this problem as follows. Let the function $g(b)$ represent the output of our model for bid b in the current state. In our particle filter, we work with

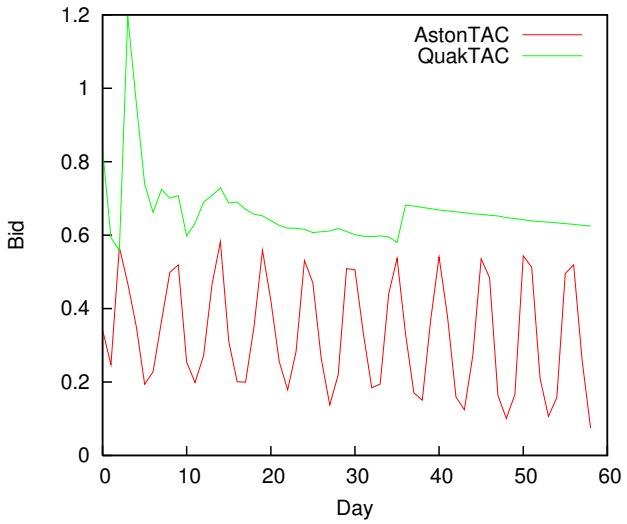


Figure 1: Daily bids of two advertisers

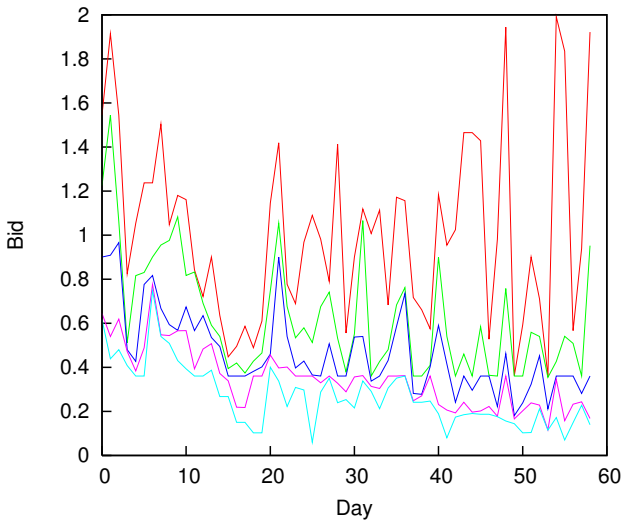


Figure 2: Top five daily bids

a discretized set of bids b^1 ... b^B. We define a probability mass function f over this set of bids:

$$f(b^i) = max(g(\frac{b^i + b^{i+1}}{2}) - g(\frac{b^i + b^{i-1}}{2}), \epsilon)Z \qquad (10)$$

for a normalizing constant Z and some small $\epsilon > 0$. The corresponding cumulative distribution function F is now strictly increasing, and $F(b^B) = 1$.

Each time the particle filter needs to draw a new particle p' based on an existing particle p, we generate f and F for each advertiser and then follow the procedure described in Section 3.1. Note that in this case, the advertiser's bid history used to generate the feature vector that is input to the model is based on the bid history stored in the particle p, and not on the (unknown) true bid history.

6. EXPERIMENTS

We now report on experiments that demonstrate the effectiveness of our particle filter for bid estimation. We begin by presenting the experimental setup and describing alternate bid estimation methods against which we compare our particle filter.

6.1 Setup

We evaluate our particle filter in three different settings. For each setting, we use our agent TacTex [7], a top-performing agent from the 2009 TAC/AA competition, as the advertiser we participate as (i.e., the agent whose observations we see and on whose behalf we are estimating bids). The other seven advertiser agents are chosen from the TAC Agent Repository[1], a collection of agent binaries. Different sets of agents are used for each of the three settings. For each setting, we run 50 games. 40 games are used to generate training data, and the remaining 10 are used for testing. For each advertiser, we train a model as described in Section 5.

For testing, we run our particle filter independently for each of the 90 60-day bidding episodes (nine independent episodes per game, as described in Section 4) contained in the test games. In each episode, we initialize each particle by drawing a bid for each advertiser from a histogram of that advertiser's initial bids. Then each day, we update our bid estimates by giving the particle filter the bid rankings, average positions, and TacTex's bid and cpc for that day and performing the update procedure described in Section 3.1.

In our particle filter implementation, we use 2000 particles and did not observe an increase in accuracy from increasing this number. We discretize the bid space into intervals of 0.01, with a maximum bid of 1.1 times the highest bid observed from any advertiser, and we set $\epsilon = 0.0001$.

Our goal in estimating bids is not to track the behavior of a specific advertiser but to get an idea of how much we would need to bid to reach a certain position. We therefore evaluate the performance of our particle filter by comparing our estimate of the nth ranked bid (based on the weighted mean of the particles) with the actual bid for each relevant value of n.

The first setting we consider involves a set of seven different advertiser agents: AstonTAC, QuakTAC, epflagent, MetroClick, Merlion, and two different versions of Schlemazl. Bidding strategies differ considerably between agents. For example, Figure 1 shows the bids of AstonTAC and QuakTAC for one particular episode. Here AstonTAC's bids tend to drift only slightly from day to day, while QuakTAC's bids take larger jumps but show a clear cyclical pattern. Figure 2 shows how the top five bids change each day, illustrating the difficulty of the bid estimation problem. For our second setting we run TacTex against seven copies of QuakTAC, and for our third setting we run TacTex against seven copies of AstonTAC.

6.2 Alternate Bid Estimation Methods

To evaluate the effectiveness of our particle filter, we need to compare its accuracy to other bid estimation methods. The first method we consider is a simple baseline of always estimating the *nth* ranked bid to be the average *nth* bid over the training set.

The second method was used in TacTex, so we will call it the TT estimator. Like the particle filter described in this paper, this method is also a form of sequential Bayesian filtering, but there are several important differences. First,

[1] `http://www.sics.se/tac/showagents.php`

instead of using a set of particles to represent a distribution over bids, the TT estimator is a grid-based method, meaning that it explicitly computes a probability mass function over a set of discrete bids. Second, the TT estimator maintains this function independently for each advertiser, instead of estimating a joint distribution over all bids, which requires a number of simplifying assumptions. Third, the TT estimator makes use of a much simpler bidding model than the models learned in Section 5.

The simple bidding model assumes that bids change in one of three ways. First, with probability 0.1, the bid jumps to a random bid. This case covers sudden jumps that are difficult to model. Next, with probability 0.5, the bid changes only slightly from the previous bid. This case reflects the behavior of AstonTAC in Figure 1. The change in bids is modeled under the assumption that the difference in logarithms of successive bids is distributed normally with zero mean. Finally, with probability 0.4, the bid changes according to a similar distribution, but the change is with respect to the bid 5 days ago. This case reflects the behavior of QuakTAC in Figure 1. This model is used for all advertisers.

The TT estimator performs a two step update each day. First, it updates the distribution over each advertiser's bid using the simple bidding model. Second, it multiplies the probability of each bid by the probability that the other advertisers' bids would be consistent with the observed bid ranking given that bid (assuming that the estimated distributions over their bids are correct) and then normalizes. Full details are available in [7].

The third alternate estimator we test is to use our particle filter with the simple bidding model from the TT estimator. We also considered the opposite combination — using the bidding models described in Section 5 with the TT estimator. However, because the TT estimator maintains only bid distributions, and not particles representing bid histories, we do not have the information required to use these bidding models. We tried using the mean of each advertiser's bid on each previous day as the bid history, but results were poor.

6.3 Estimation Results

Table 1 shows the results for all three settings for all bid estimators. For each of the 90 episodes from the 10 test games, we found the root mean squared error of the estimates, and the average RMS error is displayed. We ignored the first five game days in computing these errors so that the errors would not be skewed by start-game effects. (The method of simply using the average bid was especially inaccurate during this period.) In settings 1 and 2, we show the errors of the estimates for the top five bids, since there were nearly always at least five ranked bidders in these settings. In setting 3, however, there were often only three ranked bidders, and so we show three errors.

For all bid estimators, errors were highest on the top ranked bid and generally decreased as the rank increased. This result is expected since the top bid is essentially unbounded above and can fluctuate significantly (as in Figure 2), while lower bids tend to be grouped more tightly. Errors on settings 2 and 3 were much lower than on setting 1. Both QuakTAC and AstonTAC have somewhat predictable bidding patterns and avoid particularly high bids.

Our particle filter with the learned bidder models consistently gave the lowest error of any estimator. In all but one case (setting 3 rank 3) the difference between this error and all other errors was statistically significant ($p < 0.05$) according to a Wilcoxon matched-pairs signed-ranks test. Not surprisingly, using the average bid was worst overall. The performance of the particle filter using the simple bidder model and the TT estimator (which uses the same model) was mixed, with neither clearly outperforming the other. This result is somewhat surprising, since the particle filter is in theory a more principled approach. It may be the case that the deficiencies of the simple bidder model affect each approach differently and that in some cases the TT estimator is more robust.

6.4 Application to Bidding

Finally, while the focus of this paper has been on estimating bids accurately, the goal of this estimation is ultimately to allow an advertiser to set its own bids effectively. We now briefly explore the usefulness of our particle filter when utilized by a full bidding agent. For each setting, we ran 50 games using the original TacTex (which uses the TT estimator to estimate other advertiser's bids and then optimizes with respect to these estimates), then repeated these games using the particle filter with the learned bidder models. Surprisingly, TacTex's score did not improve in setting 1, apparently due to issues with the estimation of other advertisers' spending limits that are beyond the scope of this paper. Fortunately, QuakTAC and AstonTAC do not make significant use of spending limits, so this problem does not impact settings 2 and 3. In setting 2, using the particle filter improved TacTex's score by 452 (from 78,177), and in setting 3, the score improved by 926 (from 82,424). In both cases, the increase was statistically significant ($p < 0.05$) according to a Wilcoxon matched-pairs signed-ranks test. For reference, we ran each set of games again and fed TacTex the true bids of the other advertisers, and the scores in settings 2 and 3 increased by 847 and 1582, respectively, compared to the scores when the TT estimator was used. Thus, the use of our particle filter appears to provide us with a large portion of the gain to be had from improving bid estimation accuracy.

7. CONCLUSION

In this paper we have introduced a particle filter that can be used to estimate the bids of other advertisers in keyword auctions given a periodic ranking of their bids. The key to this particle filter is a method of sampling new particles (representing an updated set of bids) in such a way that the samples are consistent with the observed bid ranking. Additionally, we have described a learning approach to modeling the bidding behavior of other advertisers. In experiments in the TAC/AA domain, the combination of this particle filter and bidder modeling outperforms all other bid estimation methods tested, including the method that was used in the 2009 TAC/AA champion.

There are several areas in which future work is possible. The results show the importance of using accurate bidder models, and there are a number of additional conditional density estimation approaches we could try. Also, we currently only consider the problem of estimating past bids. The next step is to predict future bids, perhaps by using the bidder models to propagate the estimates forward. Testing our bid estimation approach with real world data is another necessary step. Finally, bid estimation is only one of many subproblems faced in designing a successful bidding agent,

Bid Estimator	*Average RMS error per bid estimate for each bid rank*												
	Setting 1					*Setting 2*					*Setting 3*		
	1	2	3	4	5	1	2	3	4	5	1	2	3
average bid	0.678	0.302	0.228	0.176	0.155	0.191	0.135	0.106	0.086	0.079	0.234	0.127	0.178
TT estimator	0.685	0.289	0.190	0.119	0.110	0.203	0.110	0.096	0.085	0.095	0.185	0.198	0.185
PF simple model	0.603	0.304	0.187	0.115	0.089	0.163	0.089	0.080	0.098	0.118	0.206	0.127	0.095
PF learned models	**0.459**	**0.255**	**0.135**	**0.082**	**0.066**	**0.112**	**0.060**	**0.055**	**0.049**	**0.052**	**0.135**	**0.102**	0.092

Table 1: Bid estimate errors for all estimators and settings. Significantly lowest errors in bold.

and fully integrating the methods presented here with other agent components (such as estimating clickthrough rates) remains an important challenge, as does scaling up to handle many simultaneous auctions.

8. ACKNOWLEDGEMENTS

We would like to thank the TAC/AA development team and all who contributed agents to the agent repository. We also thank Doran Chakraborty for assisting with the development of the TacTex agent. This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-0615104 and IIS-0917122), ONR (N00014-09-1-0658), DARPA (FA8650-08-C-7812), and the Federal Highway Administration (DTFH61-07-H-00030).

9. REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.

[2] P. Jordan, B. Cassell, L. Callender, and M. Wellman. The Ad Auctions Game for the 2009 Trading Agent Competition. Technical report, 2009.

[3] P. Jordan and M. Wellman. Designing the ad auctions game for the trading agent competition. In *IJCAI 2009 Workshop on Trading Agent Design and Analysis (TADA)*, Pasadena, California, 2009.

[4] C. Kiekintveld, J. Miller, P. R. Jordan, L. F. Callender, and M. P. Wellman. Forecasting market prices in a supply chain game. *Electronic Commerce Research Applications*, 8(2):63–77, 2009.

[5] S. Lahaie, D. Pennock, A. Saberi, and R. Vohra. Sponsored search auctions. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[6] D. Liu, J. Chen, and A. Whinston. Current issues in keyword auctions. In G. Adomavicius and A. Gupta, editors, *Handbooks in Information Systems: Business Computing*. Emerald, 2009.

[7] D. Pardoe, D. Chakraborty, and P. Stone. TacTex09: A champion bidding agent for ad auctions. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, May 2010.

[8] D. Pardoe, D. Chakraborty, and P. Stone. TacTex09: Champion of the first Trading Agent Competition on AdAuctions. Technical Report AI-10-01, Department of Computer Science, The University of Texas, 2010.

[9] R. E. Schapire, P. Stone, D. McAllester, M. L. Littman, and J. A. Csirik. Modeling auction price uncertainty using boosting-based conditional density estimation. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002.

[10] S. Schone. *Auctions In the Electricity Market : Bidding When Production Capacity Is Constrained*. Springer, Berlin, 2009.

[11] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kauffmann, 1999.

Conviviality Measures

Patrice Caire
FNRS Research Fellow
PReCISE Research Center
University of Namur
Belgium
patrice.caire@fundp.ac.be

Baptiste Alcalde,
Leendert van der Torre
CSC
University of Luxembourg
Luxembourg
baptiste.alcalde@laposte.net
leendert@vandertorre.com

Chattrakul Sombattheera
Faculty of Informatics
Mahasarakham University
Thailand
chattrakul.s@msu.ac.th

ABSTRACT

Conviviality has been introduced as a social science concept for multiagent systems to highlight soft qualitative requirements like user friendliness of systems. In this paper we introduce formal conviviality measures for dependence networks using a coalitional game theoretic framework, which we contrast with more traditional efficiency and stability measures. Roughly, more opportunities to work with other people increases the conviviality, whereas larger coalitions may decrease the efficiency or stability of these involved coalitions. We first introduce assumptions and requirements, then we introduce a classification, and finally we introduce the conviviality measures. We use a running example from robotics to illustrate the measures.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence —*Multiagent Systems*

General Terms

Theory, Measurement, Human Factors

Keywords

Agent Societies and Societal Issues, Artificial Social Systems, Dependence Networks

1. INTRODUCTION

Computer systems have to be user friendly and convivial, a concept from the social sciences defined by Illich as "individual freedom realized in personal interdependence" [10]. Multiagent systems technology can be used to realize tools for conviviality when we interpret "freedom" as choice [5]. For example, if there is only one supply store in your building, then you depend on it for your supplies, but if there are several stores, then you do not depend on a single store. We say that there is more choice, and thus it is more convivial. The challenge of measuring conviviality breaks down into the following research questions:

1. How to define conviviality measures?
2. How to classify conviviality?
3. What are the assumptions and requirements?
4. Why do we need a new measure?
5. How to use the measures in multiagent systems?

We measure conviviality by counting the possible ways to cooperate, indicating degree of choice or freedom to engage in coalitions. Our coalitional theory is based on dependence networks [6, 19], labeled directed graphs where the nodes are agents, and each labeled edge represents that the former agent depends on the latter one to achieve some goal.

To explain the need for the conviviality measures, we show the difference with stability and efficiency measures. Tools for conviviality are concerned in particular with dynamic aspects of conviviality, such as the emergence of conviviality from the sharing of properties or behaviors whereby each member's perception is that their personal needs are taken care of [10]. In such dynamic circumstances, the stability of the coalitions is an important criterion. Moreover, traditional coalition formation and game theoretic methods have been focused on the efficiency of coalitions.

The focus on dependence networks and more specifically on their cycles, is a reasonable way of formalizing conviviality as something related to the freedom of choice of individuals plus the subsidiary relations –interdependence for task achievement– among fellow members of a social system. However, this freedom of choice view is not the only view of conviviality, not even the most pertinent one. For example, in earlier work we define conviviality masks based on Taylor's idea that conviviality "masks the power relationships and social structures that govern societies." [20] A conviviality mask is a transformation of social dependencies by hiding power relations and social structures to facilitate social interactions, and conviviality mask measures can be defined to measure these transformations.

In this paper we do not consider Polanyi's notion of empathy, which needs trust, shared commitments and mutual efforts to build up and maintain conviviality, or the many definitions and relations with other social concepts discussed in the conviviality literature, referring to qualities such as trust, privacy and community identity.

The layout of this paper is as follows. In Section 2 we introduce a running example from coalition formation in robotics, in Section 3 we discuss stability and efficiency measures for dependence networks, in Section 4 we discuss the assumptions and requirements of conviviality measures, in Section 5 we introduce a conviviality classification, and in Section 6 we introduce the conviviality measures.

Cite as: Conviviality Measures, P. Caire, B. Alcalde, L. van der Torre, C. Sombattheera, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 895-902.

2. RUNNING EXAMPLE: NAO ROBOTS

We shall now give a scenario where we can discuss how our system works. In an office building, there are assistant robots to human being workers. The workers need office materials, which are scare and are to be shared, in order to accomplish their jobs. A worker may need materials which are not currently available at their desks, e.g., someone else in the building is using it. It is considered waste of time and unproductive for a worker to ring everyone else to find out where the needed materials are and leave his desk to collect those materials himself. Instead, the worker can submit a request to the robots to get and/or deliver the needed materials for him while he can continue with other works at his desk. We shall refer to a request submitted to the robots as a main task, which can be split into a number of tasks.

The communication between the workers and the robots can be done via a simple web-based application, which will transmit the request of the worker to the robots as well as keeping track of their status. However, the robots have limited computational resources. They only keep track of what they have done recently. They rely on each other to provide information about finding the location of a material. Basically, the last robot which dealt with it will know. We assume the existence of such an application as well as the communication network is stable and reliable. In general, the robots then travel from place to place during the working hours. A depiction of this scenario is presented in Figure 1.

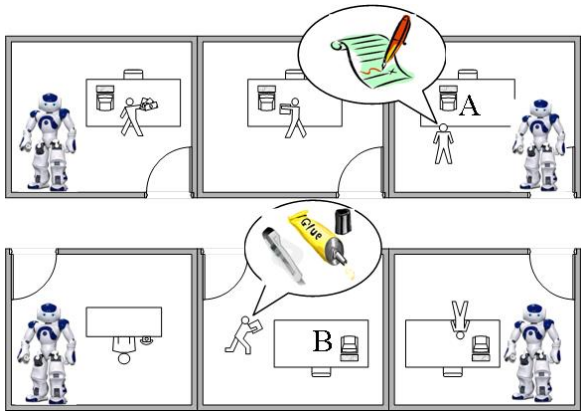


Figure 1: A depicted scenario of robots in office building.

In our example here, we assume that there is a set of 4 Nao robots, $N = \{n_1, n_2, n_3, n_4\}$ and there are two main tasks: $T_A = \{t_1, t_2\}$ and $T_B = \{t_3, t_4\}$, where t_1 is to deliver a pen to desk A, t_2 is to deliver a piece of paper to desk A, t_3 is to deliver a tube of glue to desk B, and t_4 is to deliver a cutter to desk B. Note that executing a task involves detailed actions, such as grabbing and dropping the pen, which are beyond the scope of this paper. These tasks are given to robots, which need to complete them in minimal use of power. Therefore, they need to minimize their travel time. A robot has incentives to perform as many tasks as possible as well as to save its battery life. [1].

Upon receiving the tasks, robots need to form coalitions to finish them. Due to limited resources in the robots, not all of the robots know about the tasks. There are multiple steps to carry out all the tasks from start to finish. First, the information known by each robot is who has the information about the sources and the destinations of the resources needed to accomplish the tasks. The actual coordinates, involving the present location of each material and the respective desk, are revealed only after an agreement on a coalition among the robots has been made. This involves interdependency among robots. Second, robots need to decide how they form coalitions, i.e., which ones will join to carry out each main task. Third, for each possible coalition, each robot needs to plan for their optimal route to carry out the assigned task.

At the start, robots get the information concerning the material locations and the distances between the materials and destinations. For example, robot n_1, regarding task t_1, knows i) nothing about the source of the pen, i.e., where it currently is, and ii) the destination of the pen, i.e., where it must be delivered. Regarding task t_2, robot n_1 knows where the paper is but knows nothing about its destination. Table 1 presents the knowledge of the robots about the tasks and the current distances among the robots, the materials and the destinations.

Table 1: Robots' knowledge (top); Distances (bottom).

Robot	n_1				n_2			
Task	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
Source		✓			✓			
Destination	✓		✓					✓

Robot	n_3				n_4			
Task	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
Source				✓			✓	
Destination		✓						

Distances among locations				
Robot	Pen	Paper	Glue	Cutter
n_1	10	15	9	12
n_2	14	8	11	13
n_3	12	14	10	7
n_4	9	12	15	11
Destination	Pen	Paper	Glue	Cutter
Desk A	11	16	9	8
Desk B	14	7	12	9

Upon receiving information about the tasks, robots form coalitions to execute them. We refer to a coalition as a group of robots executing a main task, i.e., either T_A or T_B. Robots joining the coalition are to execute the task, e.g., deliver the pen to desk A. For example to accomplish all the tasks t_1, t_2, t_3, t_4, the following coalitions may be formed: $C_0 : \{(n_1, t_3), (n_2, t_2), (n_3, t_4), (n_4, t_1)\}$, $C_1 : \{(n_1, t_1), (n_2, t_2), (n_3, t_3), (n_4, t_4)\}$ and $C_2 : \{(n_1, t_3), (n_2, t_4), (n_3, t_2) (n_4, t_1)\}$. For agents to execute their tasks, they need to know an optimal plan such that they can minimize their costs for executing the task. Given the knowledge, they are capable of computing for an optimal route for getting the assigned materials and for delivering it[2]. Therefore, the robots can generate plans for themselves after they have been given tasks. However, discussing the details about generating plans for the robots is out of the scope of this paper.

[1]This is common for goal-oriented agents. However, the model can also be applicable to other types of agents.

[2]Planning for an optimal route is a typical shortest path finding algorithm, whose implementations are available and can be deployed on the robots.

3. EFFICIENCY AND STABILITY

3.1 Definitions

There are many ways to define efficiency. Generally speaking, efficiency in a coalition is a relation between what agents can achieve as part of the organization compared to what they can do alone or in different coalitions. In this section and to give an illustration on our example, we recall two definitions of efficiency: the *cost efficiency* (Def. 3.1), and the *economic efficiency* (Def. 3.2).

DEFINITION 3.1 (COST EFFICIENCY). *Let $N = \{n_1, \ldots, n_j\}$ be the set of agents, T the set of tasks (or goals), and $C \subseteq N$ a coalition. Let $Cost : 2^N \rightarrow \mathbb{R}$ be the function that associates to a coalition the cost of achieving all tasks of T. Then, C is cost efficient iff $\forall\, n_i \in C$, $(Cost(n_i) - Cost(C)) > 0$.*

DEFINITION 3.2 (ECONOMIC EFFICIENCY). *A coalition is economically efficient iff i) no one can be made better off without making someone else worse off, ii) no additional output can be obtained without increasing the amount of inputs, iii) production proceeds at the lowest possible per-unit cost [14].*

Stability of coalitions is related to the potential gain in staying in the coalition or quitting the coalition for more profit (i.e., free riding). Hence, several elements come to play for the evaluation of a coalition's stability.

First, the coalition outcome should be greater than the individual ones cumulated. This is usually computed via a characteristic function such as proposed by [13]. Therefore, a necessary condition to stability is that the characteristic function is positive, i.e., acting as a group is overall more beneficial than acting individually.

Second, the distribution of benefits should be fair. Several functions, named *sharing rules* where proposed such as Shapley value [16], nucleolus [15], and Satisfactory Nucleolus [12]. The leading idea is to take the individual contribution and the free rider's value into account when sharing the benefits.

For the purpose of illustration, we introduce the concept of *core* to check the stability of a coalition. Indeed, it is relatively (computably) simple to check if a coalition is in the core. Informally, a coalition is in the core iff no sub-coalition is more profitable. Formally, the core follows Def. 3.3.

DEFINITION 3.3 (CORE). *Let $x \in \mathbb{R}^N$ be a pay-off allocation vector, $\nu : 2^N \rightarrow \mathbb{R}$ be the characteristic function (pay-off function), and $C \subseteq N$ a coalition. Then, x is in the core iff $\sum_{i \in N} x_i = \nu(N)$ and $\sum_{i \in C} x_i \geq \nu(C)$.*

3.2 Efficiency computation

Let us apply the above definitions to our example. From Table 1 of Sect. 2 we can compute the distance for each robot to do each task, as displayed on Table 2:

Using this table we can compute the cost of executing tasks in a given coalition by adding up the costs of each robot to the assigned task. For instance, the cost of $C_1 : \{(n_1, t_1), (n_2, t_2), (n_3, t_3), (n_4, t_4)\}$ is $Cost(C_1) = 87$, whereas the cost of $C_2 : \{(n_1, t_3), (n_2, t_4), (n_3, t_2)\ (n_4, t_1)\}$ is $Cost(C_2) = 93$, and the cost of $C_3 : \{(n_1, t_1), (n_1, t_3), (n_2, t_2), (n_4, t_4)\}$ is $Cost(C_3) = 86$.

Table 2: Distances between robots and their tasks.

	t_1	t_2	t_3	t_4
n_1	10+11=21	15+16=31	9+12=21	12+9=21
n_2	14+11=25	8+16=24	11+12=23	13+9=22
n_3	12+11=23	14+16=30	10+12=22	7+9=16
n_4	9+11=20	12+16=28	15+12=27	11+9=20

These costs have to be compared to the costs of each robot doing all tasks on their own, which are respectively 94 for n_1 and n_2, 91 for n_3, and 95 for n_4. As a conclusion, we can say that C_1 and C_3 seem efficient for all robots, whereas C_2 is a bad option with respect to efficiency for n_3 only.

We can see that C_3 is more cost efficient than C_1. However, we should note that C_1 is not economically efficient. Indeed, there is a coalition $C_0 : \{(n_1, t_3), (n_2, t_2), (n_3, t_4), (n_4, t_1)\}$ where at least one agent is better off without making anyone worse off (actually, this applies for all of them), all the rest been equal. If we compare $Cost(C_0) = 81$ to $Cost(C_3) = 86$, we conclude that C_0 is economically efficient and more cost efficient than C_3.

3.3 Stability computation

As explained earlier, we will check the stability of the coalitions according to the core definition (Def. 3.3).

We can see that C_1 is not in the core, hence not stable, because there exist at least a sub-coalition which is more profitable, e.g., C_3. Indeed, in the context of C_1 the robot n_1 can threaten n_3 to do the task t_3 for the same outcome but less cost. The two other robots agree since their respective pay-off is unchanged. The coalition C_2 is also not in the core, since n_2 can be threatened by all agents and n_3 can be threatened by n_2 and n_4.

In contrast, C_3 and C_0 are in the core. In fact, in C_3, even if n_4 has a lower cost than n_1 for the task t_1, neither n_2 nor n_4 can handle the task t_3 without decreasing their global pay-off, i.e., they are satisfied with this coalition.

The coalition C_0 is stable, according to the core definition, and it also involves all the robots, whereas C_3 leaves one robot idle (n_3) and gives additional work to another one (n_1). As a preliminary conclusion, for efficiency and stability, as well as for the sake of balancing the workload (which was also an objective of the main goal achievement), the coalition C_0 seems to be the best.

3.4 Need for other coalition measures

Efficiency and stability metrics are commonly used to evaluate coalitions. The former giving an assurance on the economical gain reached by being in the coalition, the later giving a certainty that the coalition is viable on the long term. Therefore, the positive evaluation of a coalition against these two metrics is often considered to be a prerequisite for the coalition formation.

However, depending on the application domain, other functional and non-functional requirements, e.g., security, user-friendliness or conviviality, may play an important role in the choice of a coalition. Requirements may be considered in a trade-off at the same level as efficiency and stability, or as a further filtering criterion, to select among otherwise efficient and stable coalitions. This highlights the need for further metrics, such as the proposed conviviality metrics.

4. ASSUMPTIONS AND REQUIREMENTS

According to [3], conviviality may be measured by the number of reciprocity based coalitions that can be formed. Some coalitions, however, provide more opportunities for their participants to cooperate with each other than others, being thereby more convivial. To represent the interdependencies among agents in the coalitions, we use dependence networks. First, we present definition 4.1 [3], illustrated with our running example. Then, we review our assumptions and requirements for the conviviality measures we define.

Recalling Section 2, two steps are needed to achieve each task. To each step, we associate a goal for a robot to reach. For example, to perform task t_1, *deliver pen to desk A*, robots must have the goals g_{1S}, *get the pen from its source*, and g_{1D} *deliver it to its destination*. Abstracting from tasks and plans we define a dependence network as in 4.1 [3]:

DEFINITION 4.1 (DEPENDENCE NETWORKS). *A dependence network is a tuple* $\langle A, G, dep, \geq \rangle$ *where:* A *is a set of agents,* G *is a set of goals,* $dep : A \times A \rightarrow 2^G$ *is a function that relates with each pair of agents, the sets of goals on which the first agent depends on the second, and* $\geq$*:* $A \rightarrow 2^G \times 2^G$ *is for each agent a total pre-order on sets of goals occurring in his dependencies:* $G_1 >_{(a)} G_2$.

In our example Section 3, robots form the coalitions C_0, C_1 and C_2. Let DN_0, DN_1 and DN_2, visualized in Figure 2 (a), (b) and (c), be three dependence networks respectively corresponding to these coalitions, where:
Nao robots $N = \{n_1, n_2, n_3, n_4\}$,
Goals $G = \{g_{1S}, g_{1D}, g_{2S}, g_{2D}, g_{3S}, g_{3D}, g_{4S}, g_{4D}\}$,
where dependencies are built from Table 1 and preferences are the following:

- for DN_0: $dep(n_1, n_4) = \{g_{3S}\}$, $dep(n_2, n_1) = \{g_{2S}\}$, $dep(n_2, n_3) = \{g_{2D}\}$, $dep(n_3, n_2) = \{g_{4D}\}$, $dep(n_4, n_1) = \{g_{1D}\}$, $dep(n_4, n_2) = \{g_{1S}\}$;
  Robot n_4 prefers to deliver pen to desk A than to get it : $\{g_{1D}\} >_{(n_2)} \{g_{1S}\}$;
- for DN_1: $dep(n_1, n_2) = \{g_{1S}\}$, $dep(n_2, n_1) = \{g_{2S}\}$, $dep(n_2, n_3) = \{g_{2D}\}$, $dep(n_3, n_4) = \{g_{3S}\}$, $dep(n_3, n_1) = \{g_{3D}\}$, $dep(n_4, n_3) = \{g_{4S}\}$, $dep(n_4, n_2) = \{g_{4D}\}$;
  Robot n_4 prefers to get cutter than deliver it to desk B: $\{g_{4S}\} >_{(n_2)} \{g_{4D}\}$, and n_3 prefers to get glue than deliver it to desk B: $\{g_{3S}\} >_{(n_1)} \{g_{3D}\}$;
- for DN_2: $dep(n_2, n_3) = \{g_{4S}\}$, $dep(n_1, n_4) = \{g_{3S}\}$, $dep(n_3, n_1) = \{g_{2S}\}$, $dep(n_4, n_1) = \{g_{1D}\}$, $dep(n_4, n_2) = \{g_{3S}\}$;
  Robot n_3 prefers to deliver pen to desk A than get glue: $\{g_{1D}\} >_{(n_2)} \{g_{3S}\}$.

4.1 Assumptions

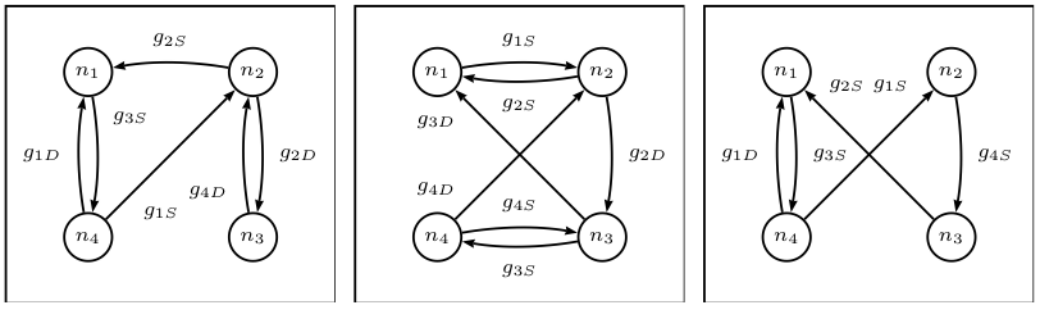
In this work, the cycles identified in a dependence network are considered as coalitions. These coalitions are used to evaluate conviviality in the network. Cycles denote the smallest graph topology expressing interdependence, i.e, conviviality, and are considered as atomic relations conveying interdependence. When referring to *cycles*, we are implicitly signifying *simple cycles* (as defined in [7]), also discarding self-loops. Moreover, when referring to conviviality, we always refer to potential interaction not actual interaction.

In our second assumption, we consider the conviviality of a dependence network to be evaluated in a bounded domain, i.e., over a $[min; max]$ interval. This allows to read the values obtained by any evaluation method.

4.2 Requirements

The first requirement for our conviviality measures concerns the size of coalitions. This requirement is captured by the statement that larger coalitions are more convivial than smaller ones. We express this requirement through the following two cases. First case, a dependence network DN_i with a coalition of size n is better for conviviality than a DN_j with coalition of size $m = (n - \alpha)$, where $m < n$. For example, consider a coalition for peace in the world. The more countries participate, the better it is. Second case, a dependence network DN_i with a coalition of size n is better for conviviality than a dependence network DN_j with two coalitions, one of size k and the other of size l, such as that $k + l \leq n$, all else being equal. This is motivated by the fact that having one large coalition eliminates the risk of being exposed to potential competition from other coalitions, which may be looking for the same resources.

Our second requirement concerns the number of coalitions. It is captured by the statement that the more coalitions in the dependence network, the higher the conviviality measure (all else being equal). This requirement is motivated by the fact that a large number of coalitions indicates more interactions among agents, which is positive in term of conviviality according to our definition based on interdependence.



(a) DN_0 (b) DN_1 (c) DN_2

Figure 2: Dependence networks DN_0, DN_1 and DN_2.

5. CONVIVIALITY CLASSIFICATION

Based on the requirements outlined in Section 4, we now propose a conviviality classification that allows an intuitive grasp of conviviality measures through a ranking of the dependence networks. First, we introduce the five definitions of conviviality classes, from the absolute best to the absolute worst convivial networks.

5.1 Definitions

DEFINITION 5.1 (P). *A dependence network DN is P convivial (most convivial), iff all agents in DN belong to all cycles, i.e.,* $\forall a_i \in A$ *and* $\forall c_k \in C, a_i$ *is s.t.* $a_i \in c_k$*, where* $C = \{c_1, \ldots, c_l\}$ *is the set of all cycles.*

DEFINITION 5.2 (APE). *A dependence network DN is APe convivial, iff all agents in DN belong to at least one cycle, i.e.,* $\forall a_i \in A, \exists c_k \notin C, s.t.\ a_i \in c_k$*, where* $C = \{c_1, \ldots, c_l\}$ *is the set of all cycles.*

DEFINITION 5.3 (N). *A dependence network DN is N convivial, iff there exists at least one cycle in DN, and there is at least one agent not in a cycle, i.e.,* $\exists a, b \in A$ *s.t.* $a, b \in c_k$*, where* $c_k \notin C$*, and* $\exists d \in A$ *s.t.* $d \notin c_i, \forall c_i \in C$*, where* $C = \{c_1, \ldots, c_l\}$ *is the set of all cycles.*

DEFINITION 5.4 (AWE). *A dependence network DN is AWe convivial, iff there is no cycle in DN, i.e.,* $C = \{\emptyset\}$*, and s.t.* $\exists dep(a, b) = \{g_i\}$*, where* $a, b \in A$ *and* $g_i \in G$*.*

DEFINITION 5.5 (W). *A dependence network DN is W convivial (worst convivial), iff there is no dependency between the agents in DN, i.e.,* $\nexists dep(a, b) = \{g_i\}$*, where* $a, b \in A$ *and* $g_i \in G$*.*

Figure 3, illustrates the different types of dependence networks that correspond to each conviviality class. The arrow on the top of the figure depicts the direction of increasing conviviality. The scale goes from the worst case (no conviviality) to the best case (maximal conviviality).

5.2 Examples

Consider the three dependence networks DN_0, DN_1, and DN_2 respectively corresponding to the robots coalitions C_0, C_1, and C_2 illustrated Figure 2. All robots belong to at least one cycle. Hence, from Definition 5.2, C_0, C_1, and C_2 belong to the *APe* conviviality class. They are said to be *Almost Perfectly* convivial. All robots are engaged in reciprocal dependence relations: each one gives to the coalition and receives from it. All robots are pursuing goals and cooperate with at least one other robot to achieve their tasks.

With a different initial knowledge, the potential coalitions formed may belong to other conviviality classes. For instance, if in the initial knowledge table, the destination of task t_2 is known by n_4 instead of n_3, then coalition C_{01} is represented by the dependence network DN_{01} depicted on Figure 4. We note that n_3 depends on another robot (n_2), but that this dependency is not reciprocated, leaving n_3 out of any coalition. Hence, n_3 being isolated, the corresponding coalition belongs to the N conviviality class.

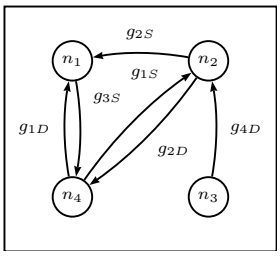


Figure 4: Conviviality class N.

Consider now that in C_1, each robot knows the information, i.e., source and destination, about one task only, and is assigned the task it knows about. Then, not a single robot depends on another, since each robot knows exactly what to do on its own. There is no cooperation among the robots, each is isolated. The corresponding network consists of four nodes and no dependencies. Therefore, this coalition belongs to the W conviviality class. Similarly, if all robots know all the information about all tasks, then any task assignment results in a coalition corresponding to a network of conviviality class W, as all robots may perform any task by themselves without having to cooperate with any other robots to obtain the information concerning the source and destination of the office supplies they have to move.

5.3 Preliminary distinctions among measures

Returning to the efficiency and stability measures presented in Section 3, we can already see a major distinction between conviviality and the two former metrics. Indeed, in order to evaluate conviviality, we need to perform an analysis of the dependencies between the agents, i.e., we must consider the topological aspects of the task (or goal) dependencies in the graph. This is not the case in efficiency and stability metrics, which only compare coalitions to subcoalitions or individuals in terms of global pay-off. Therefore, we cannot rely on similar functions to evaluate conviviality. Finally, conviviality is orthogonal to efficiency and stability, and trade-off situations are to be expected.

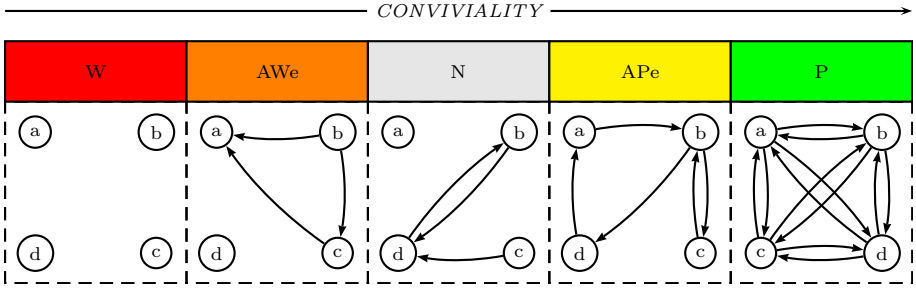


Figure 3: Conviviality classes.

6. CONVIVIALITY MEASURES

We now propose two indices based on graph properties and built on three measures; the number of: agents in the network, agents that belong to at least one cycle, and cycles.

We define the *in-the-loop* index ρ_{DN_i} as the ratio of the number of agents in cycles in relation to the total number of agents in the network. For example, computing the *in-the-loop* index for coalitions C_0 and C_{01}, respectively represented by DN_0 depicted Figure 3a and DN_{01} depicted Figure 4 yields: $\rho_{DN_0} = 1$ and $\rho_{DN_0} = 0.75$. Although useful this metric and its inverse ($\beta = 1-\rho$) do not allow to differentiate between the number of coalitions present in the network, e.g., we obtain the same value for the networks depicted Figure 3a, 3b and 3c: $\rho_{DN_0} = \rho_{DN_1} = \rho_{DN_2} = 1$.

We therefore set a second index, the *connectivity* index δ_{DN_i}, defined as the average length of cycles in the network, and reflecting when coalitions are larger, i.e. more convivial. Computing the *connectivity* index for coalitions C_0, C_{01}, C_1 and C_2 we obtain: $\delta_{DN_0} = 1.333$ and $\delta_{DN_{01}} = 1$, $\delta_{DN_1} = 1$ and $\delta_{DN_2} = 2$. Clearly, this result satisfies the requirement of Section 4 that the larger the cycle, the more convivial the network, all else being equal, however, it fails to distinguish between DN_{01} and DN_1 even though DN_1 contains more cycles. Moreover, intuitively, and per our classification Section 5, DN_{01} containing one isolated node is less convivial than DN_1, in which each node belongs to at least one cycle.

Combining the two indices, as well as defining other measures based on global graph properties, does not seem to create more accurate measures, i.e., satisfying our requirements, hence highlighting the need to capture the network topologies more precisely.

Therefore, we propose a conviviality measure constructed on our assumption Section 4 that conviviality measures are based on the coalitions the agents form with each other. As at least two agents are needed to form a coalition, the measure is based on pairs of agents. More specifically, what is measured is the number of coalitions to which any two given agents in the dependence network belong, the evaluation being performed over the whole network. Furthermore, to allow comparisons between dependence networks of various sizes and to increase its usefulness, the measure must be defined over a bounded space, such as $[0; 1]$.

6.1 Bounding evaluations

Our first step is to define a function that evaluates conviviality over one pair of agents – denoting a *partial* measure of conviviality. Let $coal_{DN_i}(a, b) \in \mathbb{N}$, be **the number of cycles that contain both** a **and** b in a dependence network DN_i, where $a, b \in A$ and $a \neq b$. Then, based on $coal(a, b)$, we construct a bounded conviviality measure. We start by determining the maximum number of cycles that contain any two agents. We note that the number of cycles containing two agents, $coal(a, b)$, can neither be more than the maximum number of cycles possible containing two (given) agents nor less than no cycle at all. Let Θ be the **maximum number of cycles between two agents**, we write:

$$0 \leq coal(a, b) \leq \Theta \tag{1}$$

In order to determine the maximum number of cycles, let us first assume that the set of goals is reduced to only one goal, i.e., $|G| = 1$, and the DN is a clique on all goals. We note that the maximal number of cycles is the summation of the maximal number of cycles for each cycle length. We call L the cycle length. In addition, as stated in Section 4, we do not consider self-loops in the evaluation. So, the smallest cycle to consider is $L = 2$, and that can happen iff the set of agents A has a cardinality greater than or equal to 2, i.e., $|A| \geq 2$. Trivially, when $|G| = 1$ there can be at most 1 cycle between two agents such that $L = 2$.

To have a cycle of length $L = 3$, we must have at least 3 agents in the DN, i.e., $|A| = 3$. We can already generalize, saying that the maximal cycle length L in a DN with $|A|$ number of agents is $L = |A|$.

Furthermore, given two agents $a, b \in A, a \neq b$, a cycle of length $L = 3$ is found if there is a agent $c \in A, c \neq a, c \neq b$ such that there is an edge from a (resp. b) to c and an edge from c to b (resp. a). The maximum number of cycle of length $L = 3$ is then obtained by choosing one agent c among the agents which are neither a nor b, without repetition and with order. Since there are $|A|-2$ such c agents, the maximal number of cycle of length $L = 3$ can be expressed by the permutation $P(|A|-2, 1)$, where $P(n, k)$ is the usual permutation defined in combinatorics by: $P(n, k) = \frac{n!}{(n-k)!}$, where n is the number of elements available for selection and k is the number of elements to be selected ($0 \leq k \leq n$)

For length $L \geq 3$, applying a similar reasoning, we obtain the maximal number of cycles of length L by choosing $L-2$ agents among $|A| - 2$, without repetition and with order, hence given by the expression $P(|A| - 2, L - 2)$.

Finally, as noted above, the maximum number of cycles is the summation of the maximal number of cycles for each cycle length. Hence for $|G| = 1$, the maximum number of cycles, $\Theta_{|G|=1}$, is:

$$\Theta_{|G|=1} = \sum_{L=2}^{L=|A|} P(|A| - 2, L - 2) \tag{2}$$

Now, for $|G| \geq 1$, we can choose for each edge one goal among $|G|$. Since the number of edges for a cycle is defined by its length L, we have a maximum of $|G|^L$ cycles of length L. Therefore, the maximum number of cycles, Θ, is expressed as follows:

$$\Theta = \sum_{L=2}^{L=|A|} P(|A| - 2, L - 2) \times |G|^L \tag{3}$$

6.2 Combining conviviality measures

In Equation 2 we obtain bounds for a pairwise evaluation. We now need to sum up all these pairwise evaluations. Let $\sum coal(a, b)$ be this summation. As there are $|A|(|A| - 1)$ pairs of agents to consider in the whole network:

$$0 \leq \sum coal(a, b) \leq |A|(|A| - 1) \times \Theta \tag{4}$$

If we want to bound our conviviality measure *conv* over [0;1], i.e., $0 \leq conv \leq 1$, then we get the following Equation 5:

$$0 \leq \frac{\sum coal(a, b)}{A(A-1) \times \Theta} \leq 1 \tag{5}$$

We can now write Equation 6 to express the pairwise conviviality measure of a dependence network DN:

$$conv(DN) = \frac{\sum coal(a, b)}{\Omega} \tag{6}$$

where we write $\Omega = A(A-1) \times \Theta$ for the sake of readability, for the remainder of the paper.

6.3 Conviviality computation

In Table 3, we present the conviviality evaluation for each dependence network, illustrated in Figure 2. As expected, the value for the maximum number of cycles is a large number, $\Omega = A(A-1) \times \Theta = 111360$. The evaluations are performed using the pairwise measure defined in Equation 6. The results return $conv(DN_1) = 0.000143 > conv(DN_2) = 0.000125 > conv(DN_0) = 0.0000897$, indicating that DN_1 is the most convivial network, followed by DN_2 and that DN_0 is the least convivial.

We observe that $conv(DN_1) > conv(DN_0)$, coincides with our intuition as clearly, DN_1 contains more cycles than DN_0. This result satisfies our requirements Section 4 namely, that the more coalitions in the dependence network, the higher the conviviality measure (all else being equal). DN_1 is more convivial than DN_0 as more cooperation may occur among the robots in coalition C_1. Similarly, the computation returns $conv(DN_1) > conv(DN_2)$ as DN_1 contains more cycles than DN_2. The result $conv(DN_2) > conv(DN_0)$ reflects the fact that DN_2 contains a cycle larger than the largest cycle in DN_0. In this *grand* coalition (n_1, g_{3D}, n_4, g_{1S}, n_2,g_{4S}, n_3, g_{2S}, n_1), all four robots may cooperate. As per our requirements Section 4, such a coalition is more convivial as the potential conflicts that may arise among several smaller coalitions is reduced.

Computing $conv(DN_{01})$ returns, as expected, the smaller value ($\frac{8}{\Omega} = 0.0000718$) highlighting the lesser conviviality of coalition DN_{01}.

In our running example, we measured conviviality by counting the possible ways for robots to cooperate, indicating the degree of choice or freedom to engage in coalitions. Indeed, the conviviality measures allow to compare the coalitions and select the most appropriate one(s) for the multiagent system. If a high level of cooperation is needed in the system, then coalitions involving the highest number of agents and cycles will be preferred. Of course, trade-offs must be made among the system requirements, including user-friendliness and conviviality as well as efficiency and stability. However, the conviviality measures allow to provide an indicator for the level of cooperation among the agents and their degree of choice to engage in coalitions. More opportunities to work together with other agents increases the conviviality. As stated by Bradshaw et al. [11], the success of future human-agent teams relies in such sophisticated interdependence among human-agent team members.

Table 3: Measures based on dependencies.

Fig.	Pairs in 1 cycle	Pairs in 2 cycles	Conviviality $\left(= \frac{\Sigma coal(a,b)}{\Omega}\right)$
DN_0	(n_1,n_2), (n_2,n_1), (n_2,n_3), (n_2,n_4), (n_3,n_2), (n_4,n_2)	(n_1,n_4), (n_4,n_1)	$\frac{6\times1+2\times2}{\Omega} = \frac{10}{\Omega}$
DN_1	(n_1,n_3), (n_2,n_4), (n_3,n_1), (n_4,n_2)	(n_1,n_2), (n_2,n_1), (n_2,n_3), (n_3,n_2), (n_3,n_4), (n_4,n_3)	$\frac{4\times1+6\times2}{\Omega} = \frac{16}{\Omega}$
DN_2	(n_1,n_2), (n_1,n_3), (n_2,n_1), (n_2,n_3), (n_2,n_4), (n_3,n_1), (n_3,n_2), (n_3,n_4), (n_4,n_2), (n_4,n_3)	(n_1,n_4), (n_4,n_1)	$\frac{10\times1+2\times2}{\Omega} = \frac{14}{\Omega}$

7. RELATED RESEARCH

This paper builds on our previous work, Caire et al. [5] in which, conviviality has been proposed as a social concept to develop multi-agent systems. Indeed, the intuitions behind the term conviviality are significant for social IT-enabled systems, and has been very little studied so far. However, conviviality is likely to become a core design feature for such systems in the future.

In "Conviviality Measure for Early Requirement Phase" [4], Caire and Van Der Torre introduce three conviviality models using dependence networks. First, temporal dependence networks model the evolution of dependence networks and conviviality over time. Second, epistemic dependence networks combine the viewpoints of stakeholders, and third normative dependence networks model the transformation of social dependencies by hiding power relations and social structures to facilitate social interactions. The authors show how to use these visual languages in design. The description level of the paper is methodologies and languages, and conviviality measures were not defined.

The approach we use in this paper brings novelty by operationalizing an elusive intuition and proposing a way to measure one type of conviviality. Furthermore, we provide an original approach to measuring one aspect of robustness of coalitions of agents. We present two kinds of measures: a conviviality classification that captures a hierarchical structure of the dependence networks, and a pairwise measure, based on the interdependencies among robots, that provide a total order on conviviality dependence networks.

This work builds on the notion of social dependence introduced by Castelfranchi along with concepts like groups and collectives [6]. Castelfranchi brings such concepts from social theory to agent theory to enrich agent theory and develop experimental, conceptual and theoretical new instruments for social sciences . The present work takes as a starting point an abstract notion of dependence graphs initially elaborated by Conte and Sichman [19]. The notions of dependence graphs and dependence networks were further developed by the authors [19], and with a more abstract representation similar to ours, in Boella et al. [1] and Caire et al. [5].

Dependence based coalition formation is analyzed by Sichman [18], while other approaches are developed in [17, 8, 2].

The clustering coefficient provides global and local measures in social networks to indicate respectively the overall clustering of the network and the embeddedness of single nodes. Although an interesting measure, the clustering coefficient was not used in our paper as it does not include the notion of cycle fundamental to our conviviality model. The literature concerning efficiency and stability in coalition is vast and referred to in Section 3. Particularly relevant to conviviality are the works related to the fairness of sharing the benefits of coalitions as in [14, 16, 15, 12].

Similarly to Grossi and Turrini [9], our approach brings together coalitional theory and dependence theory in the study of social cooperation within multiagent systems. However, our approach differs as it does not hinge on agreements.

Finally, works emphasizing agents' interdependence as a critical feature of multiagent systems, particularly for the design of systems involving joint interaction among human-agent systems such as in Johnson and Bradshaw et al. "coactive" design [11].

8. SUMMARY

Conviviality has been introduced as a social science concept for multiagent systems to highlight soft qualitative requirements like user friendliness of systems. In this paper we introduce formal conviviality measures for dependence networks using a coalitional game theoretic framework, which we contrast with more traditional efficiency and stability measures. Roughly, more opportunities to work with other people increases the conviviality.

We classify conviviality by five degrees of conviviality, from most convivial or fully connected to least convivial or unconnected. The assumptions of our conviviality measures are a bounded domain given by a [min; max] interval, and coalitions are represented by simple cycles. The requirements of our conviviality measures are that larger coalitions are more convivial than smaller ones, that coalitions based on mutual dependence are more convivial than coalitions based on reciprocal dependence, and that more possible coalitions indicate a higher conviviality, all else being equal. We need a new measure, since more traditional measures like efficiency or stability measures are different. More opportunities to work with other people increases the conviviality, whereas larger coalitions may decrease the efficiency or stability of these involved coalitions. Conviviality measures may be seen as a particular kind of robustness measures, since more convivial systems have more opportunities for agents to choose their partners, and therefore are also more robust when partnerships break up. However, in contrast to robustness measures, conviviality measures do not say anything about the stability of the coalitions. Note that intuitively, these measures may be related, for example that more stable coalitions may be more convivial, but in this paper we have disentangled these measures as much as possible. We illustrate how to use the conviviality measures in multiagent systems by discussing an example from robotics.

In further research we contemplate the need to come up with different notions of conviviality when one wants to say that a "goal-directed" system is convivial (e.g., a G2C portal) as opposed to when one claims that an "open interaction platform" is convivial (e.g., Facebook or LinkedIn). While in the first case there is an owner of the system (the city government or the tax authority) that imposes a certain way of doing things in order to reach some goals that may be convivial or dictatorial, in the second place one may think of functionalities that make the platform prone to a conviviality that is closer to the intuitions operationalized in this paper (e.g. artifacts that facilitate bringing friends into the platform and doing interesting things with them thanks to the platform). We will also look into the "conviviality as mask" intuition where conviviality appears to be more a matter of etiquette and discretion, than a matter of task interdependence. We expect that the proposed measures do not apply in a straightforward way, but that new measures will be needed to capture further views of conviviality.

Acknowledgements. This work was funded by the FNRS, with additional support of the B and B. We thank the anonymous referees for their helpful comments.

9. REFERENCES

[1] G. Boella, L. Sauro, and L. van der Torre. Power and dependence relations in groups of agents. In *International Conference on Intelligent Agent Technology*, p. 246–252, 2004.

[2] G. Boella, L. Sauro, and L. van der Torre. Algorithms for finding coalitions exploiting a new reciprocity condition. *Logic Journal of the IGPL*, 17(3):273–297, 2009.

[3] P. Caire. *New Tools for Conviviality: Masks, Norms, Ontology, Requirements and Measures*. PhD thesis, Luxembourg University, Luxembourg, 2010.

[4] P. Caire and L. van der Torre. Conviviality measure for early requirement phase. In *Normative Multi-Agent Systems. Dagstuhl Seminar Proceedings*, volume 09121, 2009.

[5] P. Caire, S. Villata, G. Boella, and L. van der Torre. Conviviality masks in multiagent systems. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, p. 1265–1268, 2008.

[6] C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[8] A. Gerber and M. Klusch. Forming dynamic coalitions of rational agents by use of the dcf-s scheme. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, p. 994–995, 2003.

[9] D. Grossi and P. Turrini. Dependence theory via game theory. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, p. 1147–1154, 2010.

[10] I. Illich. *Tools for Conviviality*. Marion Boyars Publishers, London, August 1974.

[11] M. Johnson, J.M. Bradshaw, P.J. Feltovich, C.M. Jonker, M. Sierhuis, and B. van Riemsdijk. Toward coactivity. In *International Conference on Human-Robot Interaction*, p. 101–102, 2010.

[12] L.G. Kronbak and M. Lindroos. Sharing rules and stability in coalition games with externalities. *Marine Resource Economics*, 22:137–154, 2007.

[13] M. Mesterton-Gibbons. *An Introduction to Game theoretic Modelling*. Addison-Wesley, CA, USA, 1992.

[14] A. O'Sullivan and S.M. Sheffrin. *Economics: Principles in Action*. Pearson Prentice Hall, 2006.

[15] D. Schmeidler. The nucleolus of a characteristic functional game. *SIAM journal of Applied Mathematics*, 17:1163–1170, 1969.

[16] L.S. Shapley. A value for n-person games. *Annals of Mathematical Studies*, 28:307–317, 1953.

[17] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.

[18] J.S. Sichman. Depint: Dependence-based coalition formation in an open multi-agent scenario. *J. Artificial Societies and Social Simulation*, 1(2), 1998.

[19] J.S. Sichman and R. Conte. Multi-agent dependence by dependence graphs. In *First International Joint Conference on Autonomous Agents & Multiagent Systems*, p. 483–490. ACM, 2002.

[20] M. Taylor. Oh no it isn't: Audience participation and community identity. *Trans, Internet journal for cultural sciences*, 1(15), 2004.

Author Index