AAMAS 2012

The 11th International Conference on Autonomous Agents and Multiagent Systems

June 4—8, 2012

Valencia, Spain

Proceedings

Volume III

IFAAMAS

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

ISBN-10: 0-9817381-3-3
ISBN-13: 978-0-9817381-3-0

Contents

Session 4B – Agent Societies

Session 5B – Teamwork II

Session 1C – Learning I

Session 2C – Learning II

Session 3C – Human-agent Interaction

Session 4C – Argumentation & Negotiation

Session 5C – Emergence

Session 1D – Social Choice I

Session 2D – Social Choice II

Session 3D – Economies & Markets I

Session 4D – Economies & Markets II

Session 5D – Auction & Mechanism Design

Session 1E – Game Theory I

Session 2E – Game Theory II

Session 3E – Game Theory III

Session 4E – Game Theory IV

Session 5E – Game & Agent Theories

Session 1F – Planning

Session 2F – Knowledge Representation & Reasoning

Session 3F – Agent-based Software Development

Session 4F – Logics for Agency

Session 5F – Logic and Verification

Main Program - Extended Abstracts

Innovative Applications

Virtual Agents

Robotics

Agent Reasoning

Agent Cooperation

Economic paradigms

Agent-based simulations

Agent societies and Societal issues

Learning and Adaptation

Agreement Technologies

Demonstrations

Main Program - Extended Abstracts

Emergence of Multi-generational Migration Behavior by Adaptiogenesis to Environmental Changes

(Extended Abstract)

Katsuya Suetsugu, Atsuko Mutoh, Shohei Kato, Hidenori Itoh
Dpt. of Intelligence and Computer Science, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi 466–8555
{suetsugu, atsuko, shohey }@juno.ics.nitech.ac.jp

ABSTRACT

The target of our study is the Monarch Butterfly, which is known for its multi-generational migration behavior: it migrates between southern Canada and Mexico over the course of one year within three to four generations. In spite of many reported studies, little is known about what influences their migration. We approach this subject by using an ecosystem model consisting of artificial agents and five areas. We simulate under the environmental condition that the average annual temperature rises every year, which is modeled on the current global temperature rise. Our agents emerge the migration behavior similar to the multi-generational migration of the actual Monarch. The migration process of the agents is discussed.

Categories and Subject Descriptors

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

General Terms

Agent design

Keywords

Artificial Life, Adaptive Behavior, Multi-agent Simulation

1. INTRODUCTION

The subject in our study is the Monarch Butterfly (*Danaus plexippus L., Nymphalidae, Lepidoptera*), which is known for its multi-generational migration behavior: it migrates between southern Canada and Mexico over the course of one year within three to four generations. In spite of many reported studies, little is known about what influences their migration. Our purpose of study is to reveal the reason why Monarchs migrate. It is believed that the gradual rise in air temperature is the triggers for the Monarch to migrate. In this study, we model an ecosystem consisting of artificial agents and five areas, and simulate Monarch behaviors over long periods of time.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Table 1: Sensory information.

Variable	Variable is True if
X_0	Is it diapausing? ($state_j$ = Dp)
X_1	Is it hungry?
X_2	Does a plant exist around there?
X_3	Does other agents exist around there?
X_4	Is daylight more than 12 hours?
X_5	Does it feel cold? ($s_j - ea_j < tmpr_i$)
X_6	Does it feel hot? $s_j + ea_j > tmpr_i$

Table 2: Actions.

	Behavior
W	Do not move
E	Move toward a plant and eat food.
R	Reproduce a new agent
D	Go into diapause / Stop diapausing
Mn	Migrate toward northern area ($area_i$ to $area_{i+1}$)
Ms	Migrate toward southern area ($area_i$ to $area_{i-1}$)

2. ECOSYSTEM MODEL

2.1 Agent

An agent can sense seven types of information in Table 1. By using sensory information, an agent decide its behavior only once in a day. In this paper, the action strategy is expressed by n-output binary decision diagram (n-BDD) [3], which is an extension of BDD. An agent $agent_j$ (j is identifier) has three genetic component and characterized as

$$agant_j(ea_j, cs_j, st_j), \tag{1}$$

where ea_j is a thermal sensitivity, cs_j is a cold resistance of the diapause agent, st_j is the action strategy. These genetic components are unique to each agent. An agent can reproduce a new agent by crossing, and genetic components of a child are generated from that of both parents by crossover and mutation. An agent is removed if it reaches its maximum life-span (200 days) or run out its energy.

An agent decides the action act_j by

$$act_j(t) \quad = \quad st_j(X_0(t), X_1(t), \cdots, X_6(t)), \tag{2}$$

where t is the number of steps. The variable X_m is true if the condition in Table 1 is met.

Six actions of an agent are shown in Table 2. After the action, the energy in_j is updated by

$$in_j(t) = in_j(t-1) + f(act_j(t), td), td = |s_j - tmpr_i|, \tag{3}$$

where function f is the update function of the energy level and is proportional to the difference in temperature between s_j and $tmpr_i$. E is an only action which increases its energy, and other actions decrease.

An agent has the state $state_j$ as its internal parameter. We defined three states — Cp, Dp, and Rp — which an agent can enter. The Cp state is the larval stage and is an initial state when it is first born. In the Cp state, only W and E are selectable actions and the agent can change its state to Rp after 30 days. The Rp state is an adult stage. In the Rp state, an agent can select all five actions. The Dp state is the reproductive diapause stage. Reproductive diapause is a period of rest or quiescence between phases of growth or reproduction. Diapausing Monarch halts reproductive development and is resistant to cold by reducing body temperature. In the Dp state, agent can select any action except R. By the D action, an agent changes its state from Rp to Dp or from Dp to Rp.

Diapausing agent has cold resistant and other states does not. The suitable temperature s_j is given by

$$s_j = \begin{cases} \mathrm{S_A} - cs_j, & \text{if } state_j = \mathrm{Dp} \\ \mathrm{S_A}, & \text{otherwise} \end{cases} \quad (4)$$

where S_A is the temperature suitable for nondiapausing agents.

2.2 Area

The ecosystem has five areas that we label as $area_0$ to $area_4$ from south to north. Each area is modeled on the area of North and Central America where the migration of Monarchs actually occurs. $area_i$ has three environmental parameters, which are temperature, day length, and foods. These three environmental factors have significant effects on the migration of the Monarch.

Temperature is decided by two kinds of environmental changes: long-term and short-term. A long-term change is an annual temperature rise. A short-term change is a daily temperature changes. Thus, we define a temperature $tmpr_i(y, d)$ in $area_i$ at year y and day d as

$$tmpr_i(y, d) = long_i(y) + short_i(d). \quad (5)$$

where $long_i(y)$ is a long-term change, $short_i(d)$ is a short-term change, y is a year ($y = t\ /\ 365$) and d is a day ($d = t \bmod 365$). To configure a short-term change, we used real data from the past 20 years in each original area (collected by [2]) and calculated the average annual data by trigonometric function. The day length is defined as the time difference from sunrise to sunset. We compute the time of sunrise and sunset by an approach in reference [1]. Food is a source of vital energy for the agents. Amount of foods is large if temperature is suitable for foods. A food is removed when it is eaten by an agent or reaches its maximum life-span.

3. EXPERIMENTS AND DISCUSSION

In this section, we present simulation results. We placed 200 agents with randomly generated genetic components in $area_0$. We simulated our proposed model under the environmental condition that the average annual temperatures of each area rise every year (Experiment 1). In Experiment 1, long-term environmental change $long_i(y)$ is given by

$$long_i(y) = 0.01 \times y, \quad (6)$$

Fig.1 shows the number of agents that stay in $area_0$ or migrate from $area_0$ to the others for 2000 years. Agents gradually expanded their migration range toward north with a temperature rise. In later simulations, 74.5 % of the agents migrated to $area_3$ or $area_4$, and agents migrate between $area_0$ and $area_4$ within 3.76 generations on an average. We can say that agents' migration is closely similar to actual Monarchs' migration because general migration route of Monarchs is from wintering places in Mexico (which is $area_0$ in our simulation) to areas located at a latitude of more than 40 degrees north (which are $area_3$ and $area_4$), and their one round-trip migration requires 3 to 4 generation.

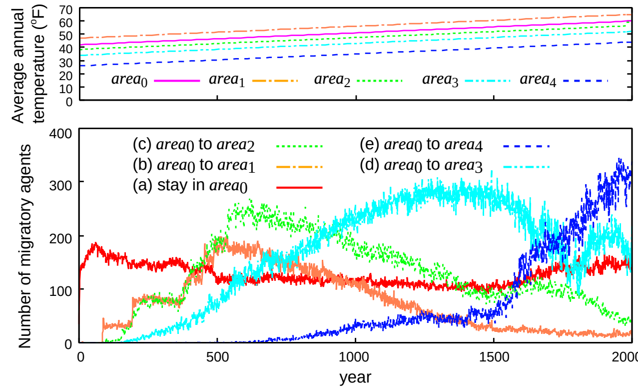


Figure 1: The number of agents that stay in $area_0$ or migrate from $area_0$ to $area_n$ for 2000 years. Data was obtained after 30 experimental runs.

To examine the relation between the migration and the temperature rise, we simulated under the condition that the average annual temperatures are constant (Experiment 2). We simulate with $long_i() = 1.0, 2.0, \cdots, 20.0$. As a result, we found that 21.5 % of the agents migrate to $area_1$, 25.8 % migrate to $area_2$ and 3.7 % migrate to $area_3$ when $long_i(y) = 7.0$, and no agents migrate from $area_0$ when $long_i(y) \geq 8.0$. These results show that Monarchs' migration pattern is not emerged under the environmental condition in which the average annual temperatures are constant. A comparison of Experiments 1 and 2 leads us to conclude that the average annual temperature rise is a trigger for the multi-generational migration of the Monarch.

4. CONCLUSION

We have designed the agent model to reveal the migration of the Monarch Butterfly in computer simulation. Agents emerged the multi-generational by adapting to temperature rises. We confirmed similarities between the agents' migration and the actual Monarchs' migration.

5. REFERENCES

[1] K. Nagasawa. *Computations of sunrise and sunset.* Chijin-Shoin, 2000. (in Japanese).

[2] National Climatic Data Center. NNDC Climate Data Online (Global Summary of the Day). `http://cdo.ncdc.noaa.gov/CDO/cdo`.

[3] T. Sawada, A. Mutoh, S. Kato, and H. Itoh. A model of biological differentiation in adaptiogenesis to the environment. *8th International Conference on the Simulation and Synthesis of Living Systems: Artificial Life VIII*, pages 93–96, 2002.

A cognitive architecture for emergency response

(Extended Abstract)

Felipe Meneguzzi
Jean Oh,
Nilanjan Chakraborty,
Katia Sycara
Carnegie Mellon University
Pittsburgh, PA 15213 – USA
{meneguzz,jeanoh,nilanjan,katia}
@cs.cmu.edu

Siddharth Mehrotra
James Tittle
Agent Dynamics Inc.
Pittsburgh, PA 15213 – USA
siddharthmehrotra11@
gmail.com
jtittle@
perceptive-cognition.com

Michael Lewis
University of Pittsburgh
Pittsburgh, PA 15260 – USA
ml@sis.pitt.edu

ABSTRACT

Plan recognition, cognitive workload estimation and human assistance have been extensively studied in the AI and human factors communities, but have seldom been integrated and evaluated as complete systems. In this paper, we develop an assistant agent architecture integrating plan recognition, current and future user information needs, workload estimation and adaptive information presentation to aid an emergency response manager in making high quality decisions under time stress, while avoiding cognitive overload. We describe its main components as well as results for en experiment simulating various possible executions of the emergency response plans used in the real world, comparing reaction time of an assisted versus an unassisted human.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Human Factors, Management

Keywords

Agent-based system development, Innovative Applications

1. INTRODUCTION

Planning for complex activities often involves consulting multiple information sources in order to reduce uncertainty associated with decision making. As humans interleave planning, execution and re-planning, managing information to meet the changing requirements becomes a cognitively demanding task. Consequently, users who must make time-critical decisions are cognitively overloaded due not only to the planning activities but also to the information requirements of the planning and re-planning. In this context, we develop the *Anytime Cognition* (ANTICO) concept to assist cognitively overloaded users through an assistant agent.

Our approach consists of recognizing a user's plan for future activities, allowing the agent to act proactively to help the user balance her workload over time. ANTICO uses predicted future plans for proactive information gathering and subsequent presentation in a suitable form that takes into consideration a user's cognitive workload available time.

We transition ANTICO to real-world scenario through a proof of concept application to assist a disaster response manager, which must deal with a chemical attack against a civilian facility in a major city, facing uncertainty throughout the response. Uncertainty stems primarily in the diagnosis and determination of the chemical used, and later from the various second-order effects. The agent assists the manager in making decisions under time-pressure, analyzing a stream of information arriving from various localized sources while keeping track of the big picture in order to coordinate multiple agencies performing activities around the affected areas. Information needed for decision making must be presented in ways that facilitate quick action, as response managers must make decisions within tight deadlines. Our contributions are threefold: first, we extend prior work on a proactive assistive agent architecture [4]; second, we deploy it in a concrete application domain; and third, we provide a simulation-based evaluation highlighting the circumstances in which gains could be obtained by our approach. We develop an emergency response scenario based on the standard disaster scenarios planning document [1], and present an application of ANTICO using this scenario, which has been fully implemented. Since potential gains from using ANTICO are closely associated to the accuracy of the agent in presenting relevant information, we evaluate the potential effectiveness of the approach through simulations of the assistance under various success rates for both intention prediction and information presentation.[1]

2. AGENT ARCHITECTURE

ANTICO architecture comprises multiple AI components including probabilistic plan recognition and intelligent information management, more details of which can be found in our previous work at [4]. Figure 1 shows a modularized view of the ANTICO components and how those components are interconnected. The rectangles represent the main components; the third-party components are drawn in dot-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1]ANTICO demo video: `http://goo.gl/o186E`

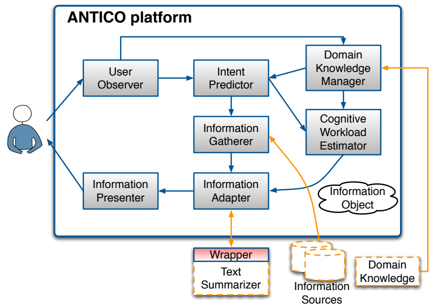


Figure 1: Architecture Overview

ted lines; a problem domain specification is provided as an input to the system; and the information object is the communication medium representing a user's information needs. Here, we specifically focus on the following two desiderata for the assistant agent. First, the agent must be able to recognize a user's activities. Second, the agent's interaction with the user must be unobtrusive and adaptive to user cognitive workload. The User Observer module is responsible for monitoring various parameters indicating a user's current activities and her environment. When a change is observed, the Intent Predictor module analyzes the new observation to identify the user's intention and makes predictions for the user's future activities according to a workflow model. Subsequently, the Information Gatherer communicates with a set of information sources to meet the information requirements relevant to the predicted future user activities. Concurrently, the agent maintains an estimated user cognitive workload based on observed temporal parameters in order to determine the appropriate level of detail in presentation.

3. EVALUATION

To demonstrate the applicability of the ANTICO approach to the real world, we developed a scenario [3] based on the National Planning Scenarios created by the Department of Homeland Security (DHS)[1].[2]In real disaster scenarios within the United States, emergency management is conducted by following an *Emergency Operations Plan* (EOP), and major urban centers in the US have them available to disaster managers. Intuitively, if the agent correctly infers the user's current intention and presents the right information in summarized form, a human user should see gains in terms of reaction time. Otherwise, if the agent displays incorrect information, a user must refer to the EOP document and suffer the time penalty of reading the irrelevant information. Given the difficulty in obtaining access to trained emergency management personnel, we have devised a simulation of a user managing an emergency scenario to evaluate the potential effectiveness of the ANTICO concept under various hypothetical error rates by the agent. The simulation consists of random walks through the workflow, following its transition probabilities, while accumulating the time taken by a human user to read the information needed to complete the task. Since we consider the amount of information actually read by the user during emergency management to be the main driver for the time spent carrying out a task, the main parameters of each step in the simulation

[2]`http://goo.gl/YtQfq`

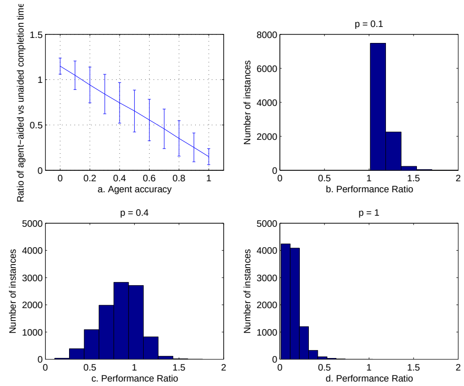


Figure 2: Simulation analysis

are the best and worst-case scenario for the number of pages required to be read to complete an activity. Each activity in the workflow is associated with a particular section (or chapter) within the EOP document[3], and the amount of information needed at each task varied from none (for tasks where the emergency manager is expected to know the information) to six pages. In order to estimate the expected time spent by a human user reading this information, we took the standard measures of reading rates obtained from the human factors literature [2]. Using the resulting times of our simulations, we calculated the performance ratio between the agent-assisted and the unaided user. These results are illustrated in the graph of Figure 2.a, which shows the various accuracy values along the X axis, as well as the performance ratios (with standard deviation) along the Y axis. Furthermore, we illustrate in more detail the specific number of samples associated with each performance ratio in the histograms of Figure 2.b-2.d for p equal to 0, 0.4 and 1. Bars to the left side of each histogram show samples in which the agent led to improved performance. Notice that at $p = 0.1$ the user's performance tends to be worse than the unaided user, but already at $p = 0.4$, most of the samples indicate an improvement in user performance.

Acknowledgement: This research was sponsored by the U.S. Army Research Laboratory (ARL) and the U.K. Ministry of Defence (MoD) and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, the U.S. Government, the MoD or the U.K. Government. The U.S. and U.K. governments are authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.

4. REFERENCES

[1] J. K. Benini. National planning scenarios, version 21.3. Homeland Security Council, March 2006.

[2] R. P. Carver. *Reading rate: A review of research and theory.* San Diego, CA, US: Academic Press, 1990.

[3] F. Meneguzzi, J. Oh, N. Chakraborty, K. Sycara, S. Mehrotra, and M. Lewis. Anytime cognition: An information agent for emergency response. In *The Fifth Annual Conference of the International Technology Alliance*, Adelphi, MD, USA, 2011.

[4] J. Oh, F. Meneguzzi, and K. Sycara. Probabilistic plan recognition for intelligent information agents. In *Proc. ICAART*, pages 281–287, 2011.

[3]We had access to the table of contents of the EOP for a major US city, from which data size estimates were derived.

An Adaptive System for Proactively Supporting Sustainability Goals

(Extended Abstract)

Sarah L. Hickmott
RMIT University
Melbourne, Australia
sarah.hickmott@rmit.edu.au

Liam Magee
RMIT University
Melbourne, Australia
liam.magee@rmit.edu.au

James A. Thom
RMIT University
Melbourne, Australia
james.thom@rmit.edu.au

Lin Padgham
RMIT University
Melbourne, Australia
lin.padgham@rmit.edu.au

Categories and Subject Descriptors

I.2.11 [**Distributed AI**]: Multi Agent Systems

General Terms

Management, Measurement, Design

Keywords

agents, sustainability indicators, design, holistic, proactivity, adaptivity, prototype

1. INTRODUCTION

We have developed a software application for a new, emerging approach to sustainability reporting, where a multi-agent system is an integral part of the overall architecture. The agent-oriented approach readily achieves the functionality required for this application, and the Belief Desire Intention (BDI) agent framework assisted in clarifying system behaviour across our heterogeneous, cross-disciplinary research team.

It has recently been suggested that current reporting practices are failing to capture the full picture of whether an organisation's practices are sustainable, e.g. [5, 1, 2]. This is due, in part, to the way economic issues are often considered independently from environmental and social issues, and vice versa. Moreover, sustainability reporting is typically addressed using either locally defined *or* external, standardised indicator sets. This forces a choice between measuring what is most relevant to the organisation, on the one hand, and what allows for comparability with other organisations, on the other. There is a need to support organisations to use a combination of indicators that capture specific local concerns *and* indicators which can be used to communicate and compare the organisation's performance externally.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Our research group of computer and social scientists has been developing a software system that aims to facilitate sustainability reporting in accordance with a new reporting framework that has recently been proposed to address these issues [5]. The software needs to support organisations to combine their own locally defined indicators with global standards, in a sustainability reporting template, and to look at their performance holistically. It should do this in a manner that both adapts to and guides the user. That is, the application ought to be responsive to end user preferences as to how to perform tasks and define indicators, but also proactive with respect to the goals of the underlying philosophy. For example, the system should allow users to define their own indicators, guide them to identify a set of indicators that holistically monitors their performance, and suggest the use of standard indicators when compatible with their needs.

The system we are developing currently exists as a continually evolving prototype (see Figure 1), being trialled with case study participants for ongoing feedback. The multi-agent component mediates between an interactive web interface and an extensible RDF-based data store for capturing information about sustainability projects, and the specific indicators used to measure them. Fundamental properties of the BDI agent paradigm have readily met our above-mentioned needs: the proactive, goal-oriented features enable us to easily guide and support the user; and, the context-sensitive manner in which agents achieve their goals allows us to build a system that can readily adapt to specific user needs. Also, the goal-plan framework lends itself to the easy addition of automated reasoning support, and allows quick adaptation of the prototype system in response to case studies.

Furthermore, the BDI framework, and the available design tools, have facilitated a highly interactive collaboration by providing an effective structure for communication between the computer and social scientists in our research team. We have found that the agent oriented entities of goals, events, plans and beliefs are sufficiently intuitive and jargon-free, that it is possible for non-technical team members to understand and contribute to the agent design much more directly than we believe could happen using a more traditional approach.

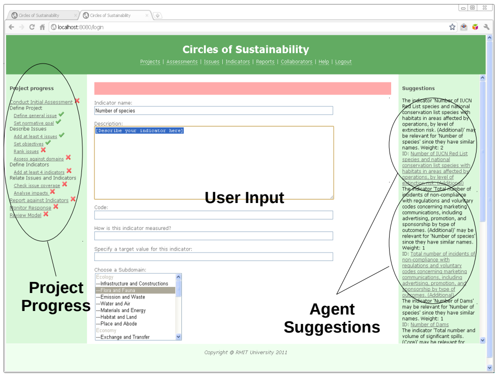


Figure 1: Screen shot of indicator suggestion.

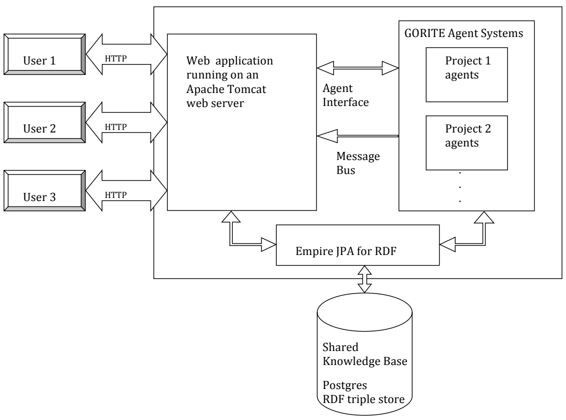


Figure 2: Overall architecture of system

2. SYSTEM DESCRIPTION

The system architecture needs to support multiple users accessing multiple sustainability projects simultaneously via the web. Though an individual user may be associated with multiple projects, in a single browser session they can only be connected to one project. This led us to design the four main components of our architecture as shown in Figure 2. **Users** access the system via any modern web browser; the **Web application** running on an Apache Tomcat web server handles session management (including authentication), interaction with users via HTTP, sending and receiving messages to and from the agent system, and storing and retrieving information in the knowledge base; the **Agent system** has a separate instantiation that manages each project; and the **Knowledge base** provides a persistent store for shared knowledge about sustainability indicator sets as well project-specific knowledge.

3. COMPARISON WITH OTHER SYSTEMS

We conclude by briefly comparing our application with other sustainability reporting systems, highlighting advantages offered by our use of agent systems.

Numerous commercial vendors have developed systems for environmental sustainability reporting, such as VERISAE, Cloudapps, TBL2, IHS and SIMPLIFi. Ecological Footprint calculators are the most widely used examples of such software, and can be used by individuals, enterprises, cities and countries to measure the embodied biocapacity to sustain production of goods and services [4]. While such tools are very useful in calculating specific measures, they are built using specific and inflexible assumptions, and are generally designed to measure only environmental (not social or economic) aspects of sustainability.

More closely related to our application is a widely-used online reporting tool, the MDG Dashboard [3], that supports visualising different sustainability data sets taken from the UN MDG indicator database. The MDG allows other datasets to be used, but requires specialist technical knowledge to prepare them, and once created, does not allow for multi-user editing. Our application is similarly holistic in philosophy to this system, but is far more customisable in terms of the types and relationships between indicators. It also differs in allowing multiple indicator sets to be applied to a single project, and for these to be edited by end users.

Web-based multi-user collaborative environments, such as wikis, blogs and social networking sites, have become increasingly popular for documenting and reporting on projects. Such systems, even where they support structured data sets (such as semantic wikis), still required considerable customisation for the specialised case of sustainability reporting. Hence, while *flexible*, they do not offer the *guidance* necessary for developing complex indicator reporting structures.

Existing systems support an impressive array of reporting approaches. However none sufficiently address the challenges of "bottom-up" sustainability reporting – supporting a high degree of flexibility without sacrificing context-specific guidance. While extensive qualitative and quantitative processes exist in the literature for "bottom-up" reporting, to date these have not made their way into supporting software. By using agent systems to provide unobtrusive support to the process of developing sustainability indicators, our application facilitates flexible collaboration and structured guidance in a novel way.

4. ACKNOWLEDGEMENTS

This work is funded by the Australian Research Council (ARC) grant LP0990509 and supported by the industry partners and other team members on this project.

5. REFERENCES

[1] S. Bell and S. Morse. *Sustainability indicators: Measuring the Immeasurable?* Earthscan, London, 2008.

[2] R. G. Eccles and M. Krzus. *One Report: Integrated Reporting for a Sustainable Strategy.* John Wiley and Sons, Inc., New York, 2010.

[3] European Commission Joint Research Centre. Millennium Development Goals Dashboard, 2010.

[4] B. Ewing, A. Reed, A. Galli, J. Kitzes, and M. Wackernagel. Calculation Methodology for the National Footprint Accounts, 2010 Edition. Technical report, Global Footprint Network, Oakland, 2010.

[5] A. Scerri and P. James. Accounting for sustainability: combining qualitative and quantitative research in developing 'indicators' of sustainability. *International Journal of Social Research Methodology*, 13(1):41–53, 2010.

Cooperative Virtual Power Plant Formation Using Scoring Rules

(Extended Abstract)

Valentin Robu[1] Ramachandra Kota[2] Georgios Chalkiadakis[3] Alex Rogers[1]
Nicholas R. Jennings[1]
{vr2,rck05r,gc2,acr,nrj}@ecs.soton.ac.uk

[1] Electronics and Computer Science, University of Southampton, UK

[2] Secure Meters Ltd., Winchester, UK

[3] Dept. of Electronic and Computer Engineering, Technical University of Crete, Greece

ABSTRACT

The growing focus on sustainable and environmentally friendly energy production has resulted in the proliferation of distributed energy resources (DERs), mainly based on renewable sources like wind and sunlight. However, their small size and the intermittent nature of their supply means that such generators cannot easily be assimilated into the current electricity network (Grid) like conventional generators. Against this background, Virtual Power Plants are fast emerging as a solution to this problem whereby a large number of small energy generators may be aggregated together such that they exhibit the characteristics like a traditional generator in terms of predictability and robustness. In this work, we propose a method to promote the formation of such "cooperative" VPPs (CVPPs) using multi-agent technology. In particular, we design a payment mechanism that encourages DERs to join CVPPs with large overall production. Our method is based on strictly proper scoring rules and elicits accurate probabilistic estimates of energy production from the CVPPs—and in turn, the member DERs—which aids in the planning of the supply schedule at the Grid.

We empirically evaluate our approach using the real-world setting of 16 commercial wind farms in the UK, and we show that our mechanism incentivises real DERs to form CVPPs and, moreover, it outperforms the current state of the art payment mechanism developed for this problem.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Economics, Experimentation

Keywords

energy and emissions, scoring rules, smart grid

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1. INTRODUCTION

In recent years, a number of strands in intelligent and multiagent systems research have taken up the challenge of creating smart and robust electricity supply networks, which can make efficient use of all available energy resources, thereby reducing dependence on carbon-intensive conventional generators [4]. One representative example for this effort is the research performed as part of the iDEaS project at the University of Southampton [4, 1, 2].

In this work, we consider the problem of integrating number of distributed energy resources (DERs) into existing electricity grid. In the last decade, *distributed energy resources (DERs)*—essentially small to medium capacity (2kW-2MW) renewable energy generators—have begun to appear in greater numbers in the network. Though their deployment could in principle reduce reliance on conventional power plants significantly, their integration into the Grid is problematic since the DERs, given their small size, are largely "invisible" to the Grid. This means they cannot readily be taken into account while planning production schedules, even if their total energy production represents a significant amount. Even if visible, the uncertainty and uncontrollability of renewable energy sources inhibits individual DERs from profitably dealing with the Grid directly, because they are often unable to meet the set generation targets. On the other hand, if individual DERs could be aggregated together to form larger energy generating entities, these entities would then have the opportunity to become economically sustainable by overcoming such invisibility and unreliability problems. This has led several researchers to propose the creation of *Virtual Power Plants (VPPs)*, consisting of large numbers of DERs, which can be viewed as the virtual equivalents of conventional power stations. In previous work (Chalkiadakis *et al.* [1]), we proposed a pricing mechanism that can be used by the Grid to promote the creation of *cooperatives* of DERs, and constitutes an alternative to feed-in tariffs. However, in that approach, each CVPP only reports to the Grid a point (mean) estimates of its production. An alternative that is more useful to the Grid is that production estimates are provided in the form of probability distributions, specifying the *confidence* individual entities place in their estimates.

In mechanism design literature, scoring rules with specific properties, have long been used to design payment mechanisms that incentivise agents to report private probabilistic predictions truthfully and to the best of their forecasting abilities [3]. Thus, in this paper we propose a novel pricing scheme that serves a dual goal: (i) Incentivising DERs to join forces to form CVPPs, (ii) Incentivis-

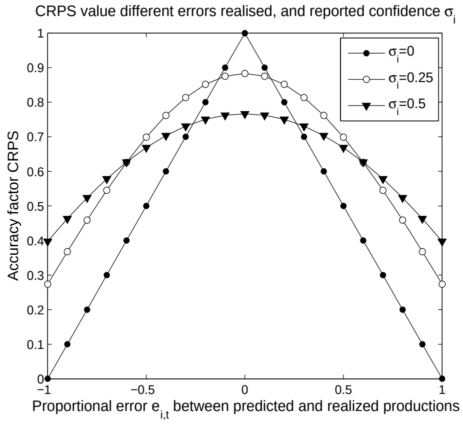


Figure 1: Example accuracy factors generated by using the scoring rule for different levels of prediction confidence

ing DERs to provide truthful probabilistic estimates of their future production, given the best information they have available.

2. SCORING RULES-BASED PAYMENTS

In our mechanism, for each half-hourly interval (called a clearing period in electricity markets), the DERs are asked to report not only an estimate of their production, but also an expected error, which reflects how accurate (in percentage terms) they expect their estimates to be. The confidence that an agent places in its own predictions is modeled through the standard deviation σ_i of its expected prediction error. Using scoring-rule based payments, the amount an agent receives from the Grid (or the CVPP if it has joined a CVPP), depends not only on how accurate the mean prediction at each clearing period t is, but also on the confidence the agent (representing a DER) reports in its predictions.

To illustrate this concept intuitively, Figure 1 exemplifies a scoring rule-based accuracy factor (which forms a part of our payment functions) for different values of σ_i vs. actual error $e_{i,t}$, for a clearing period t. What is interesting to observe here is how this error varies for different values of the reported standard deviation σ_i. If DER i is highly confident in its predictions (reporting $\sigma_i = 0$), the maximum reward for accuracy can be achieved, but only if the actual error is also close to 0. However, if the actual relative error is high, then reporting a higher σ_i (i.e., less confidence) provides a better reward.

In our formal analysis, we prove that, in all cases, our payment functions are *strictly proper*. This is a crucial property in this setting, which means that all agents will accurately declare their privately calculated distributions, reflecting their confidence in their own forecasts. Without a strictly proper payment mechanism in place, agents may be untruthful or simply not bother to provide the most accurate estimates they have available.

3. EXPERIMENTAL STUDY

We study the performance of the proposed pricing functions in a real-life, renewable electricity generation scenario. Specifically, we consider the setting of Ecotricity, one of the largest renewable generation and distribution companies in the UK[1]. Ecotricity owns 16 wind farms distributed across the UK, with installed nominal capacities ranging from 0.5 MW to 16 MW. The geographical distribution of these turbines is shown in Figure 2.

[1] www.ecotricity.com

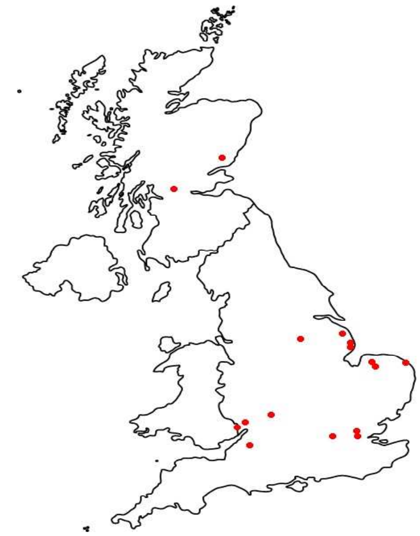


Figure 2: Map of Ecotricity farm locations

For our experiments, we collected half-hourly wind speed data for a 10-week period from 15 February to 30 April 2011. The data was collected from the website uk.weather.com. Both the actual and predicted wind data for each half hour were collected using the geographical locations of the 16 wind farms of Ecotricity. These were then converted into power outputs using the power curve formulas obtained from a large wind turbine manufacturer.

We conducted several sets of experiments, where we studied both the performance accross the 16 wind farms, and across the prediction horizon (i.e. number of hours in advance the wind prediction is made). We observe that, for all settings, our pricing functions incentivise DERs to join CVPPs, as their profit is always higher in a cooperative. Comparing the scoring rules-based method to our previous method [1] (which only requires DERs to declare pointwise estimates of their productions), we see that agents prefer the scoring rule based scheme. In fact, we found that DERs facing higher degree of uncertainty (such as those predicting their output with a longer prediction horizon) stand to benefit the most from the new mechanism. This is because allowing agents to report the uncertainty in their estimates allows them to avoid being harshly punished in settings with high uncertainty. Thus, our payment mechanism is especially well-suited for settings where predictions may be prone to large errors, such as wind-based energy generation.

4. REFERENCES

[1] G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, and N. R. Jennings. Cooperatives of distributed energy resources for efficient virtual power plants. In *Proc. of the 10th Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 787–794, 2011.

[2] E. H. Gerding, V. Robu, S. Stein, D. Parkes, A. Rogers, and N. R. Jennings. Online mechanism design for electric vehicle charging. In *Proc. 10th Int. Conf. Aut. Agents and Multi-Agent Systems (AAMAS'11)*, pages 811–818, 2011.

[3] T. Gneiting and A. Raftery. Strictly proper scoring rules, prediction and estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007.

[4] S. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings. Putting the "smarts" into the smart grid: A grand challenge for artificial intelligence. *Communications of the ACM*, 2012.

A Storage Pricing Mechanism for Learning Agents in Masdar City Smart Grid

(Extended Abstract)

Fatimah Ishowo-Oloko[1], Perukrishnen Vytelingum [2], Nick Jennings[2], Iyad Rahwan[1,3]
[1]Masdar Institute of Science & Technology, UAE
[1]University of Southampton, UK
[3]Massachusetts Institute of Technology, USA

ABSTRACT

Masdar City in the United Arab Emirates is designed to be the first modern city powered solely by renewable energy. However, the stochastic nature of renewable energy generators has remained a major challenge in their sole and large-scale deployment. Traditional approaches couple large-scale storage systems to renewable generators while more recent approaches also study how emerging technologies such as electric vehicles and micro-batteries can be used as consumer-side storage. Future smart grids are however likely to contain both forms of storage. We present a novel model of joint-storage management that allows both renewable energy suppliers and consumers to coordinate in a decentralized manner by gradually adopting storage abilities. For this model, we present a dynamic storage-pricing mechanism that makes use of the storage information from the renewable supplier to generate daily, real-time electricity prices which are communicated to the consumers.

Categories and Subject Descriptors

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

General Terms

Economics,Experimentation

Keywords

Energy and emissions, Simulation

1. INTRODUCTION

The growing threat of climate change and the depletion of non-renewable energy sources have led to the growth of sustainable development projects. In particular, sustainable urban development has been advocated as one of the factors in changing the way we produce and use energy. For example, urban planning in the future would not only involve designing buildings that minimize in-house energy use, it would also have to consider the effects of distributed energy resources like wind turbines and solar panels on land-use patterns. Thus, future cities would have to be designed in ways that are sustainable, attractive and commercially viable. Masdar City[1] is built to be a pioneering model for such future cities and it is currently fully powered by onsite renewable energy.

Given the above features of the Masdar city grid, there arises the challenge of balancing supply and demand on a constant basis. Previously, conventional energy suppliers ensured the matching of supply and demand by maintaining a generation capacity that was always much higher than demand. With renewable generation, maintaining excess capacity does not solve the problem as excess capacity is still subject to intermittency and cannot be dispatched at will. To address this challenge, electricity storage devices in the form of large utility-scale batteries [1] and small domestic batteries [3, 4] have been proposed for use with renewable energy generators.

Here, we propose the use of both utility and domestic batteries to form a decentralized energy storage-solution that can be coupled with renewable generators. Given the decentralized nature of the domestic storage and the different consumption patterns of houses, each storage unit is best represented as an autonomous agent that aims to maximize its own preferences. In line with the Abu Dhabi Economic Vision 2030 of increasing the penetration of renewable energy, we provide a novel mechanism by which renewable generators can determine the best price signal to send to their consumers giving their particular seasonal and daily patterns. This dynamic pricing mechanism improves the system efficiency and consumer savings by up to 23% and 35% respectively. Thus, it outperforms the existing fixed price mechanism.

2. MODELING THE MASDAR CITY GRID

The city is designed to be powered solely by renewable energy with a target residential population of about 40,000. We model the grid using real data as consisting of wind and solar generators and batteries at the supply end. At the consumption end, home models which may possess electricity storage capacity (either batteries or electric vehicles) are represented by agents that decide their behavior. In our models, we consider fixed time intervals consisting of

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1]www.masdarcity.ae

single days, each divided into a set of half-hourly intervals $I = \{1, 2, ..., 48\}$.

Specifically, the wind speed data was obtained from Masdar City Meteorological station for the period of August 2008 to June 2009. We modeled the stochastic process of the wind speed with the Weibull probability distribution [2] and the power outputs of the wind turbine at recorded speeds for each time interval were then obtained.

For the solar generator, the time data series of the power output from a test PV panel located at Masdar City PV contest site was used. The output was recorded every 5 minutes (288 readings per day) for the period of August 2008 to June 2009. The average of the six readings in each half-hour readings was then obtained for each time interval $i \in I$.

The utility-scale storage was modeled based on the Sodium Sulphur (NaS) deep-cycle batteries produced by NGK.[2] Our choice was based on its high power, energy density and capacity which makes them suitable for utility scale storage. Each generator and battery model has an associated levelized daily cost. Our objective was to determine the optimal supply configuration that will minimize total daily costs while ensuring that the demand of the consumers are fully met.

Finally, we built our consumer model upon the recent model for homes equipped with smart meters by Vytelingum et al [5]. Specifically, each agent $a \in A$ has a consumption profile defined as the actual amount of electricity used by agent a for time interval i during each day and a demand profile defined as the amount of electricity demanded (purchased) by the agent from the energy supplier for the corresponding time interval. In our model we assume that this consumption profile is fixed but an agent can minimize its costs by changing its demand profile.

3. THE STORAGE PRICING MECHANISM

The storage pricing mechanism (SPM) uses the availability of real-time storage information that is known to the supplier. For every generation interval, the electricity generated satisfies the demand of consumers while the excess is stored in the batteries. Thus the amount of electric charge in the utility batteries captures the amount of renewable generation that is available but not being demanded by the consumers. Using this information, the supplier can then determine when to decrease its electricity price to encourage more demand and also by how much it should decrease the price and vice versa. Therefore, our mechanism uses the correlation between the amount of charge (or discharge) and the excess (or deficit) generation.

The deviation from the previous day's price ϵ (in dollars/kWh) is given by the ratio of the cost of the batteries and the amount of electric energy that is charged into them. We define the ϵ at each interval as

$$\epsilon = \frac{c_b \times n_b}{\sum_{i \in I} P_i^{ch}} \tag{1}$$

where c_b is the levellized daily cost of each battery, n_b is the number of batteries installed and P_i^{dch} is the power output from the battery at time i. Thus, the supplier offers the consumer the incentive of savings in line with how much it also saves when it avoids using storage by reducing the price by ϵ. More formally, the price for the following day p_i^{t+1} is computed as ϵ less than the retail price of electricity i.e. $p_i^{retail} - \epsilon$ for some $i \in I$, i.e., the time periods when the amount demanded can be directly satisfied by the supplier from its generation. At all other times, the electricity is priced based on the retail price of electricity p_i^{retail}.

Given the above, a self-interested agent (with storage ability) that is interested in minimizing its cost responds by adapting its storage profile in line with changes in daily electricity prices. In more detail, the consumer agent adopts the day-ahead best-response adaptive strategy by [5]. As opposed to their model however, the agent does not need to predict the next day's price for each time slot as this is given by the supplier on a day-ahead basis. Via optimizations, the agent first computes the optimal storage capacity (maximum energy stored) required for it to minimize its cost and then it obtains the daily storage profile of energy.[3]

4. RESULTS IN BRIEF

We simulated the performance of the mechanism based on the Masdar City model and evaluated it in terms of the system efficiency and consumer benefits. The results showed that unlike the fixed pricing mechanism (currently in use in UAE) which achieves a system efficiency of 74%, the storage pricing mechanism achieved a system efficiency of up to 97.4% with all consumers having storage devices and smart meters installed in their homes. Moreover, the consumers with storage devices were able to make an average savings on their electricity bills of 35% when all the consumers are equipped with storage devices.

5. REFERENCES

[1] E. Jim and C. Garth. Energy storage for the electricity grid: Benefits and market potential assessment guide. Technical report, Sandia National Laboratories, 2010. Section 3.6.1.2, Energy Time-shift from Wind Generation.

[2] T.-Y. Lee. Operating schedule of battery energy storage system in a time-of-use rate industrial user with wind turbine generators: A multipass iteration particle swarm optimization approach. *Energy Conversion, IEEE Transactions on*, 22(3):774 –782, sept 2007.

[3] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Agent-based homeostatic control for green energy in the smart grid. *ACM Trans. Intell. Syst. Technol.*, 2:35:1–35:28, July 2011.

[4] T. Voice, P. Vytelingum, S. Ramchurn, A. Rogers, and N. Jennings. Decentralised control of micro-storage in the smart grid. In *AAAI Conference on Artificial Intelligence*, 2011.

[5] P. Vytelingum, T. D. Voice, S. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based micro-storage management for the smart grid . In *9th International Joint Conference on Autonomous Agents & Multi Agent Systems, AAMAS'2010, Toronto, Canada*, 2010.

[2] `http://www.ngk.co.jp/english/products/power/nas/index`

[3] We used IBM ILOG CPLEX 12.2 to implement and solve the optimization problem

MAS for manufacturing control: A layered case study (Extended Abstract)

Sindre Pedersen
Department of Engineering Cybernetics, NTNU
N-7491 Trondheim
+4732285568
sindre.pedersen@kongsberg.com

Bjarne Foss
Department of Engineering Cybernetics, NTNU
N-7491 Trondheim
+4792422004
bjarne.foss@itk.ntnu.no

Johannes Tjønnås
SINTEF ICT
N-7465 Trondheim
+4740825940
johannes.tjonnas@sintef.no

Ingrid Schjølberg
SINTEF ICT
N-7465 Trondheim
+4793066355
ingrid.schjolberg@sintef.no

ABSTRACT

The usage of multi-agent systems for manufacturing control seems to be sharply contrasted by classical mathematical control theory. This work emphasizes how views from both scientific fields can be combined to create the flexible and optimal manufacturing control systems of tomorrow.

Categories and Subject Descriptors

J.7 [**Computer-aided Engineering**]: Computer-aided manufacturing

General Terms

Performance, Design, Theory, Algorithms, Experimentation

Keywords

Multi-agent systems, manufacturing control, optimization, layered control system.

1. INTRODUCTION

The manufacturing industry in the western world is undergoing a paradigm shift from mass production to more specialized, customized production. In addition, the industry is experiencing increasingly diverse and volatile demands from the market [3]. The traditional control systems in the manufacturing industry are typically centralized and monolithic in structure [2]. Multi-agent manufacturing control is proposed as a new way of dealing with these challenges. Such control systems is said to have characteristics such as flexibility, agility and modularization which current rigid hierarchical control systems does not have. Some examples of such architectures can be found in [1].

In the field of control theory the notion of an agent is not very frequently used. However, multi-agent systems (MAS) is an architecture which is decentralized in nature, and as such it puts restrictions on the possible control algorithms which can be implemented. It is well known that the interconnection of locally optimal objectives does not necessarily give a globally optimal objective. As an example, if the agents are greedy non-cooperative game theory states that the total system will converge to a Nash equilibrium which need not be the same as the globally best solution [4]. Rawlings and Stuart [6] show that a network of optimal controllers can be suboptimal and in fact also unstable if not special care is taken.

If measuring the performance of a control system with some objective function J (to be maximized), at an instant T a centralized control structure, J_c, may be more optimal than a decentralized one, J_{dc}, such that $J_c(T) \geq J_{dc}(T)$. If the centralized structure implements some globally optimal solution, the difference $J_c(T) - J_{dc}(T)$ is said to be a *optimality gap* [5].

When considering a production plant, it may have thousands of measurements and control loops. The issue of plantwide control considers control system design with emphasis on the structure of the overall plant [7]. It is in the realm of plantwide control that the justification for the usage of MAS is found. MAS are architectures that implicates a decentralized control approach for plantwide control that aims to provide the system with a degree of robustness to variances. These variances can often be divided into operational variances, like rate of throughput, or external variances, like marked conditions. That is, the system should be able to function under the full range of operating conditions, internal and external, with- out the need for reconfiguration. The multi-agent community thus often empathizes that *decentralized decision making can make a control system more flexible when compared to a fully integrated (centralized) implementation.*

Although there may be a centralized control structure available that is specialized for the operating conditions *today*, it may be more beneficial to implement a decentralized structure that can also cope with the uncertainty of *tomorrow* with minimal need for expensive and time consuming reconfigurations. That is, over time the integral of the objective function may be larger for the decentralized control structure because it can handle larger variety of operating conditions, such that $J_{dc}(t)dt \geq J_c(t)dt$. The idea is illustrated in Figure 1.

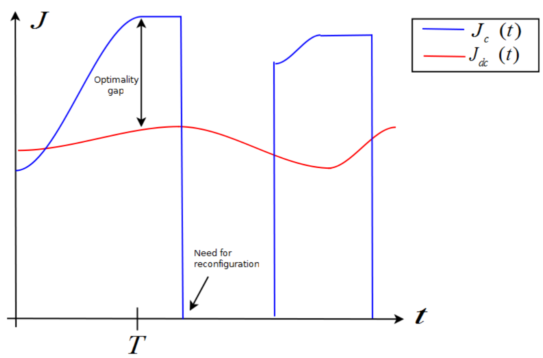


Figure 1. Optimality over time.

The multi-agent publications mentioned in the first paragraph all give excellent qualitative arguments for the use of MAS in manufacturing that follows in the lines of these arguments. More mathematical control oriented literature, on the other hand, often emphasize the optimality gap and thus argues the usage of centralized control structures. A simple idea is that *both* the optimality gap *and* the difference in accumulated difference in objective functions should be weighted, thus giving a good balance between **optimality now** and **flexibility later**. It seems to be of vital importance to explore how one can achieve such a balance. As most traditional control systems are hierarchical, examining a layered approach to multi-agent control can provide a more smooth transition into new multi-agent control systems.

2. CONTROL SYSTEM LAYERING

We consider a manufacturing plant where two control problems are to make production and distribution schedules at minimal cost. Two non-layered setups are proposed in addition to a layered setup for control of this plant.

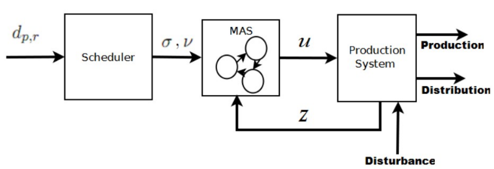


Figure 2. Multi-agent control with scheduling.

- **Single node:** The production and distribution schedule are applied directly to the simulator. The production system tries to follow this schedule strictly even in the event of disturbances.
- **Multi-agent control without scheduling:** A MAS produces and delivers orders without any scheduling layer. The orders are produced and delivered in a first-come-first-served fashion.
- **Multi-agent control with scheduling:** The schedule is fed into the MAS, as shown in Figure 2. Under normal operating conditions, the MAS follows the schedule strictly. However, in the event of disturbances, typically local interactions of agents algorithms will cause the system to deviate from the schedule.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4–8 June 2012, Valencia, Spain.

Simulations are being done on a computer software impleme nted as shown in Figure 3. Preliminary results show that a proposed MAS can cope with variances in a more flexible way than pure centralized control. Layering the MAS with a central node also improves coordination and reduces the optimality gap.

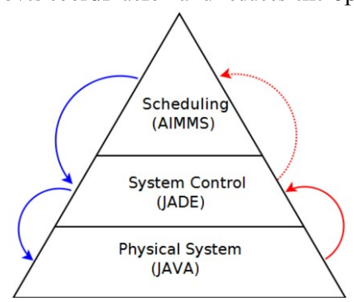


Figure 3. A layered control approach.

3. CONCLUSIONS

Many manufacturing control systems are hierarchical, and developing layered multi-agent control systems would provide the opportunity for a more smooth transition in implementation that can utilize the systems already in place. More work should also be done investigating possible performance benefits with such layered approaches, as simulation results show it can in fact improve system performance when compared to pure multi-agent control.

4. ACKNOWLEDGMENTS

This work is in part funded by FP7 NMP Grace project Contract n° NMPP2-SL-2010-246203.

5. REFERENCES

[1] Bussman, S., Schild, K. 2001. An agend-based approach to the control of flexible production systems. In Proceedings of the 8'th IEEE International Conferance on Emerging Technologies and Factory Automation, Antibes-Juan les Pins, France.

[2] Colombo, R. 2006. An agent-based intelligent control platform for industrial holonic manufacturing systems. IEEE Trans on Industrial Electronics. 53, 322-337.

[3] Marik, V., McFarlane, D. 2005. Industrial adoption of agent-based technologies. IEEE Intelligent Systems. 20, 27-35.

[4] Myerson, R. 1997. Game theory:analysis of conflict. Harvard University Press.

[5] Nocedal, J., Wright,S. 2000. Numerical Optimization. Springer.

[6] Rawling, J., Steward, B. 2008. Coordinating multiple optimization-based controllers: New opportunities and challenges. Journal of Process Control. 18, 839-845.

[7] Skogestad, S. 2000. The search for the self-optimizing control structure. Journal of process control. 10, 487-507.

Opinion Gathering Using a Multi-Agent Systems Approach to Policy Selection

(Extended Abstract)

Adam Wyner
Dept. of Computer Science
University of Liverpool, UK
a.z.wyner@liverpool.ac.uk

Katie Atkinson
Dept. of Computer Science
University of Liverpool, UK
k.m.atkinson@liverpool.ac.uk

Trevor Bench-Capon
Dept. of Computer Science
University of Liverpool, UK
tbc@csc.liv.ac.uk

ABSTRACT

An important aspect of e-democracy is consultation, in which policy proposals are presented and feedback from citizens is received and assimilated so that these proposals can be refined and made more acceptable to the citizens affected by them. We present an innovative web-based application that uses recent developments in multi-agent systems (MAS) to provide intelligent support for opinion gathering, eliciting a structured critique within a highly usable system.

Categories and Subject Descriptors

I.2.11 Multi Agent Systems

General Terms

Experimentation; Human Factors; Theory.

Keywords

e-Government and e-Democracy; argumentation.

1. INTRODUCTION

Current web technologies are fuelling an increase in the desire of members of the public to participate in democratic debate and decision making, and are also enabling governments to provide opportunities for them to do so. However, many issues arise when one considers how to analyse, evaluate and respond to the volume of data gathered.

From a developer's point of view, a key consideration in designing and building online tools for opinion gathering is the trade-off between the amount of structure provided by the tool and its ease of learning and use. Since the target audience is the general public, participation must be fostered by making the interactive system as straightforward to use as possible. If, however, the responses are to be meaningfully analysed in terms of their content, then considerable structure needs to be imposed on the data.

Clear separation of distinct issues is one problem with unstructured systems. A second important difficulty concerns how to assess and evaluate competing opinions; placing the requirement on the user to provide arguments that are sound and coherent yields no guarantee this will be accomplished. Forming coherent and well-expressed arguments is a rare skill, and people, including the highly educated, find it hard even to organise their thoughts into premises and a conclusion that follows validly from these premises. If, additionally, the arguments need to conform to, and be annotated with respect to, a structure requiring some minimum knowledge of argumentation theory, the difficulties are multiplied.

Thus, there is a clear need for online opinion gathering tools to be grounded on some solid semantic foundation whilst retaining their usability. To achieve this, we look to multi-agent systems, and in particular how the reasoning of the agents in a system can be support by a computational model of argument. In the next section we pinpoint three key developments from this field that can provide the backbone of support for a tool for online opinion gathering.

2. MAS ARGUMENTATION FOR POLICY

The first important development is computational modelling of argument [3], which has become increasingly important as a sub-field of AI in general and MAS in particular. From [3] we take the key notion that evaluating the status of an argument takes place in the context of an argumentation framework (AF), containing arguments in an attack relation, and where the status of an argument is *relative to a set of arguments* that either attack or defend it. Subsequent research on AFs has included methods for distinguishing between successful and unsuccessful attacks. The defeat relation is replaced by an attack relation, and then a preference relation on arguments is used to remove unsuccessful attacks leaving only successful attacks (i.e. defeats), so inducing a standard AF. Several kinds of preference have been suggested: we use an ordering on the social values promoted or demoted by acceptance of an argument which yields Value-based Argumentation Frameworks (VAFs) [2].

A second important development involves Argumentation Schemes, a notion imported from the study of argument in Informal Logic and Critical Thinking, but now widely used in MAS. Their importance from our perspective is that such schemes provide us with guidance on how to construct and how to attack arguments. The argumentation scheme mainly used in our tool for opinion gathering is the *Practical Reasoning* (PR) scheme for justifying the choice of an action as developed in [1]: **PR**: In the current circumstances (R), action ac should be performed, since this will bring about a new set of circumstances (S) in which a goal (g) is realised.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Realising g is desirable because it promotes a social value (v).

The third development is the study of the interaction between independent agents and how this interaction can be managed so that the system as a whole operates in as harmonious and effective a manner as possible. One semantical basis for modelling agents and their interactions, used in [1], is a transition system based on joint actions between agents (Action-based Alternating Transition System (AATS)).

We claim that these theoretical developments taken from agent-based studies of computational argumentation can support our opinion gathering task in the following ways:

- *Modelling the Domain.* The need to underpin the enterprise with an AATS determines the components that we need and structures the task of identifying them.
- *Producing Arguments.* Instantiations of the Argumentation Scheme now give us arguments that can justify various actions in the situation as modelled and various attacks on these arguments.
- *Selecting an Argument.* The arguments can now be organised into an Argumentation Framework (in particular a VAF). Choosing the best argument from those available requires us to make factual and preference assumptions, which can be modelled using agents.
- *Receiving Feedback* The chosen argument, and various possible ways of attacking it, can now be offered to the public as simple questions in a web-based survey tool.
- *Evaluating Feedback* Given the precise attacks that various people wish to make, and the relative numbers who wish to make the different attacks, we can record these in the agent system and so reconsider the factual and value assumptions in the light of what is believed and desired by the citizenry.

We focus especially on critiques about a particular proposal. Having constructed our AATS model, and generated a set of arguments and the objections to them, we evaluate the resulting VAF in accordance with our value preferences to choose a particular policy and justification. That argument can now be presented to the public for feedback using the web-based tool. We solicit feedback on the model, both disagreements and omissions, the assumptions made, and the ordering of values chosen. After an initial statement of the selected argument, participants who disagree are led through a series of screens to identify the particular points at which they disagree or want further justification.

- *Screen 1* invites the participant to agree or disagree with the proposed the circumstances. If there is disagreement, supporting evidence is presented. If the participant remains unconvinced, the argument for the circumstance can be critiqued.
- *Screen 2* offers the participant the selected policy action, which can be accepted or critiqued. Alternative actions can be selected by the participant. It can be justified why alternative actions were rejected.
- *Screen 3* asks whether participants agree or disagree with the proposed consequences of the action. Disagreement will lead to a justifying argument, and participants will either accept this and continue or be led through a critique of this further argument.
- *Screen 4* inquires whether the user agrees that the policy action promotes or demotes the value as specified in the original argument, e.g. raising taxes promotes equality. If the user disagrees, a justification is given.

3. DISCUSSION

Our opinion gathering tool brings improvements from a functional and a software engineering perspective. The improvements are the result of using the underlying AATS and the supporting agent system it enables. The tool is a significant advance on current systems [4] and an innovative and effective use of MAS techniques.

We have outlined a web-based application that deploys state of the art argumentation techniques taken from agent-based research to provide computational support for a particular stage of the policy making process - the production of a White Paper to solicit public feedback on a broadly expressed proposal. We shift the effort away from the construction of arguments to justify the proposal and the analysis of free form responses, and instead move to a precise and formal understanding of the problem and its relevant aspects. From this analysis, a model can then be created, from which arguments can be generated automatically and into which responses can be assimilated. The interactivity offered by the web is exploited by enabling the exact points of objection to be pinpointed so that disagreement can be specifically addressed by improved justifications, by modifications to the assumptions, or even by changes to the policy. The application illustrates how the full potential of the web and agent systems is achieved, not by supporting existing paper-based procedures, and so perpetuating the flaws in those processes, but rather by rethinking those procedures so that the opportunities offered can be grasped.

Acknowledgements

This work was partially supported by the FP7-ICT-2009-4 Programme, IMPACT Project, Grant Agreement No 247228 The views expressed are those of the authors and are not necessarily representative of the project.

4. REFERENCES

[1] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artif. Intel.*, 171(10-15):855–874, 2007.

[2] T. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–48, 2003.

[3] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[4] A. Wyner, K. Atkinson, and T. Bench-Capon. Towards a structured online consultation tool. In *Electronic Participation - Third International Conference*, number 6847 in LNCS, pages 286–297. Springer, 2011.

Lottery-based Resource Allocation for Plug-in Electric Vehicle Charging*

(Extended Abstract)

Matteo Vasirani
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
matteo.vasirani@urjc.es

Sascha Ossowski
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
sascha.ossowski@urjc.es

ABSTRACT

The near-future penetration of plug-in electric vehicles (PEV) is expected to be large enough to have a significant impact on the power grid. If PEVs were allowed to charge simultaneously at the maximum power rate, the distribution grid would face serious problems of stability. Therefore, mechanisms are needed to coordinate the charging processes of PEVs. In this paper, we propose an allocation policy inspired by lottery scheduling that aims at balancing fairness and selfishness, providing preferential treatment to the PEVs that have a high valuation of the electricity, while guaranteeing a non-zero share of the available power to all the PEVs to ensure fairness.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, multiagent systems*

General Terms

Algorithms, Experimentation

Keywords

Lottery scheduling, resource allocation, smart grids, plug-in electric vehicles

1. INTRODUCTION

Plug-in electric vehicles (PEVs) are expected to heavily penetrate the automotive market around the world. Thus the power grid could be greatly affected by the use of PEVs. Depending on when (and also where) the PEVs are plugged in, they could cause serious reliability problems to the local grid [1], since historically it has not been designed for that kind of intensive loads.

*Research supported by the project "AT" (CONSOLIDER CSD2007-0022, INGENIO 2010) and "OVAMAH" (TIN2009-13839-C03-02)

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

In this paper we present an allocation policy inspired by lottery scheduling that allows multiple PEVs to charge simultaneously at different charging rate.

2. ALLOCATION POLICY

We use a model of a local distribution grid composed of a substation and several charging spots. The substation converts the voltage from medium to low and feeds the charging spots where PEVs can be plugged in.

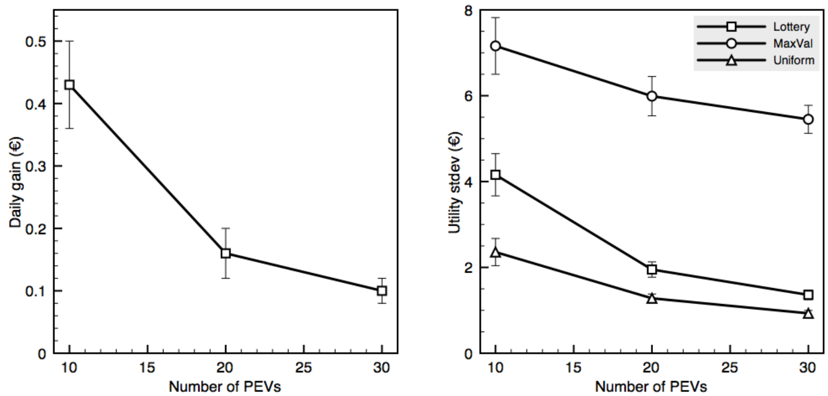
Due to the physical limitation of the distribution grid, a substation is able to provide a certain amount P of power (in kW) to the set of charging spots $\mathcal{V}$. The task of the substation agent is therefore allocating the available power P among the plugged PEVs by setting an appropriate power supply ω_i for each charging spot so as $\sum_{i \in \mathcal{V}} \omega_i \leq P$.

The substation allocates the available power P using a policy inspired by lottery scheduling, a randomised resource allocation mechanism that has been developed for operating systems [2]. Since in our problem the resource to be granted (i.e., the available power P) is infinitely divisible, the outcome of the allocation is *not a single winner*, but the determination of a share of the disputed resource, proportional to the number of tickets, to be granted to *each participant*.

Let g be the amount of base commodity owned by each PEV, x the amount of tickets issued by a PEV, and r the exchange rate that determines the worth of one ticket in terms of the base commodity ($x = r \cdot g$). To be eligible for receiving a share ω_i of the available power P, a PEV reports the amount of tickets issued by the PEV itself. As in lottery scheduling [2], the power supply that is provided to a charging spot with a plugged PEV is proportional to the worth of the amount of tickets issued by the PEV. This worth is given by x/r. The computation of the power supply is carried out according to Eq. 1.

$$\omega_i = \frac{\frac{x_i}{r_i} \cdot \zeta_i}{\sum_{j \in \mathcal{V}} \frac{x_j}{r_j} \cdot \zeta_j} \cdot P \tag{1}$$

Although the amount of issued tickets x is set by the PEV, the exchange rate r is set by the agent that controls the substation, which is built by the distribution grid operator. By *delaying* the update of the exchange rate r towards the "true" exchange rate x/g, the PEV is given the possibility of reporting an *inflated* amount of tickets. In this way, a PEV



(a) Daily gain (b) Inequality measure

Figure 1: Experimental results

may try to increase its share ω_i by inflating the worth of its tickets so as $x/r > g$. However, assuming that the PEVs behave rationally, all of them would report an inflated amount of tickets. In this case the outcome of the allocation policy would be that none of them would actually be able to increase its power supply. This undesired outcome is avoided if we put a limit to the overall inflation. When more than a fixed percentage of PEVs report an inflated amount of tickets, the power supply of the inflationary agents is reduced by the penalisation term ζ.

3. EXPERIMENTAL EVALUATION

The main objective of the evaluation is assessing the difference, in terms of average utility of PEVs, between our allocation policy (`Lottery`) and a uniform policy that equally distributes the available power P among the PEVs (`Uniform`). We refer to this difference with the term *daily gain*, expressed in €. A PEV is assumed to have an internal combustion engine that can supply driving force when the battery is depleted. The PEV's utility function is defined according to Eq. 2, where p^c is the price of fuel (in €/litre), γ^c is the internal combustion engine efficiency (in km/litre), and γ^e is the electric efficiency (in km/kWh).

$$u(b) = \frac{p^c}{\gamma^c} d - \frac{p^c}{\gamma^c}(d - b\gamma^e) = \frac{p^c}{\gamma^c}\gamma^e b \tag{2}$$

To assess how fair is our allocation policy, we further consider the outcome of another (theoretical) allocation policy that assigns all the available power P to the PEV with the highest valuation of one unit of electricity (`MaxVal`). The outcome of this policy is the same as that of an incentive-compatible auction that assigns the disputed resource to the PEV that submitted the highest bid (i.e., the agent with the highest valuation).

Fig. 1(a) shows the daily gain in a small neighbourhood, with 10 to 30 plugged PEVs. A PEV owner may gain from 10 to 40 cents of € per day, depending on the number of PEVs that compete for the available power P. In a year, this gain can account for more than 140 €. Due to the fact that different PEVs have different valuations of one unit of electricity, a uniform allocation does not reward those agents that value electricity the most. Our allocation policy instead enables the agents with higher valuations to increase their share of the available power P.

Even though the allocation policy meets the selfishness of the PEV owners, it also enforces fairness. To assess the inequality of the evaluated policies we compute the standard deviation of the utility that the PEVs obtain at the end of charging. Fig. 1(b) shows the inequality measure of the three allocation mechanisms. As expected, `Uniform` is the fairest policy that ensures the allocations with the smallest standard deviation. `MaxVal` is the most unfair policy, since the available power P is always allocated to the PEV with the highest valuation, at the expense of the PEVs that, albeit with a lower valuation, still have energy needs. `Lottery` falls in between and tends to approach `Uniform` when the number of PEVs in the system grows.

4. CONCLUSIONS

In this paper we put forward an allocation policy inspired by lottery scheduling to automatically coordinate the simultaneous charging of several PEVs. We demonstrated how our allocation policy is capable of balancing fairness and selfishness: it provides preferential treatment to those PEVs that value the electricity the most (they can report an inflated amount of lottery tickets so as to increase their share of the available power P), while guaranteeing a non-zero share of P to all the PEVs. The experimental evaluation showed that our allocation policy always ensures a utility gain compared to a straightforward uniform allocation, with gains that can reach up to 140 € per year in some scenarios. Furthermore, it reduces inequality with respect to a hypothetical allocation policy that fully assigns the disputed resource to the PEV with the highest valuation.

5. REFERENCES

[1] K. Clement-Nyns, E. Haesen, J. Driesen. The Impact of Charging Plug-In Hybrid Electric Vehicles on a Residential Distribution Grid . In *IEEE Transactions on Power Systems*, 25(1), pp. 371-380, 2010.

[2] C. Waldspurger and W. H. Weihl. Lottery scheduling: flexible proportional-share resource management. In *Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation*, pages 1–11. USENIX Association, 1994.

The Role of Social Identity, Rationality and Anticipation in Believable Agents

(Extended Abstract)

Rui Prada, Guilherme Raimundo, Joana Dimas, Carlos Martinho
INESC-ID and Instituto Superior Técnico, Technical University of Lisbon
Av. Prof. Cavaco Silva
2744-016 Porto Salvo, Portugal
rui.prada@ist.utl.pt
guilherme.raimundo@gaips.inesc-id.pt
joana.dimas@gaips.inesc-id.pt
carlos.martinho@ist.utl.pt

Márcia Baptista, Pedro A. Santos
Instituto Superior Técnico, Technical University of Lisbon
Av. Prof. Cavaco Silva
2744-016 Porto Salvo, Portugal
marcia.lbaptista@gmail.com
pasantos@math.ist.utl.pt

Jorge F. Peña
The University of Texas at Austin
1 University Station A 1105
Austin, TX 78712, USA
jorge.pena@mail.utexas.edu

Luís L. Ribeiro
PDM&FC
R. Fradesso da Silveira, 4-1B
1300-609 Lisboa, Portugal
luis.ribeiro@pdmfc.com

ABSTRACT

Agents that interact in complex social situations need to take the social context in consideration in order to perform believably. We argue that social identity is an important factor; therefore, agents should incorporate social identity theory in their behaviour, which implies the ability to categorize others (and themselves) into social groups. In addition, social situations often present social dilemmas with expected rational choices. Social identity may influence the agent to deviate from the rational choice. However, in some situations the rational choice may be the expected, and believable, behaviour. In fact, we argue that finding the dynamics between the social bias induced by social identity and the rational motivation is one of the challenges of building believable agents. In all this, anticipation takes an important role, as it is important to understand the others to cope well with a social situation.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*Human Factors*; I.2.0 [**Artificial Intelligence**]: General—*Cognitive Simulation*

General Terms

Human Factors, Design, Theory

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Keywords

Socially intelligent agents, believable agents, social awareness, social identity, individual and social rationality, anticipation

1. INTRODUCTION

Performing in complex social situations is a challenge for intelligent autonomous agents. To perform well agents need to be socially aware and get a good understanding of the social context, as it defines their context of action and interaction. A social situation implies the presence of other agents and is influenced by the characteristics of the society where the agents are situated; this includes its norms and values, the common interests and goals, its members and its social structure.

One particular aspect is that, in fact, any complex society is fragmented in different social groups, each with its own interests, values, structure and identity. Our research stresses the importance of this aspect to the creation of agents, in particular, the relevance of the social identity in the process of decision making. We believe that this is especially important if the aim is to build natural intelligence and believable (e.g. human-like) behaviour.

Social identity is part of an individual's perception of a social situation. It is the identity ascribed with basis on the perception of membership of social groups and the attachment to that membership [7]. It implies the categorization of the agents (including the self) in terms of the belongingness to social groups. Thus, apart from all individual characteristics that build a personal identity each agent should have a social identity comprised of all social groups they belong to.

Social identity often leads to a social bias in decision-making, especially in situations where it is more salient (e.g.

in the presence of other relevant social groups). People are inclined to favour other people with similar identities (e.g. belonging to the same social groups) or blame people of different social identities for the problems in a group.

This social bias influences the collaboration attitudes of the agents and the way they deal with their social commitments. For example, it is easier to break commitments to cooperate with others of distinct social identities [3, 5].

This can be extended to team commitments. Social identity will influence the decision of agents when considering following their own interests or commit (or keep the commitment) to their team goals. For example, imagine a team of 4 elements with 2 distinct identities: 3 elements with a social identity A (e.g. New Yorkers) and 1 of the elements with a social identity B (e.g. Californian). When confronted with the possibility to break the team commitment the element with the social identity B will more likely do it.

Our goal is to build agents that are able to behave believably in teams with human members. Animation studios such as Disney and Studio Ghibli have long created artificial characters that are able to portrait an illusion of life [8]. Nonetheless, the creation of synthetic characters able to, autonomously, generate behaviour perceived as believable is still an open research problem. Mainly due to the inherent multidisciplinary nature of believability, several definitions have been proposed since the seminal definition by Bates [1], and different aspects for achieving believable behaviour have been researched over the years.

One aspect is that agents should be coherent and meet users' social expectations in order to be believable, since they are perceived as social actors [6]. We argue that social identities are part of those expectations, because people will ascribe them to agents in situations where different social groups are present. Hence, agents should exhibit human-like behaviour that aligns with the findings of social identity in social psychology.

In turn, social situations often present social dilemmas (e.g. Prisoners' Dilemma). Social dilemmas are interesting simulation scenarios of agents because they represent paradoxes of individual rationality. Individual rationality is a central postulate of game theory and states that an agent acts rationality if it maximizes its expected payoff when it selects a given strategy. In social dilemmas the collective pursuit of collective rationality can lead to a Pareto sub-optimal outcome, one for which there is another alternative outcome in which no agents would be worse off and at least one agent would be better off [2].

We propose that to be believable agents should take into account individual rationality and social bias in their decision making. The challenge is achieving a good balance between the two, especially, when they lead to different decisions. For example, agents may be influenced by the social bias and behave irrationally from an individual perspective but never if that leads to their demise.

Furthermore, we would like to stress that in order to be believable in social situations and social dilemmas agents need the ability to anticipate and take others in consideration. This is a crucial point for achieving any kind of social intelligence [4]. In our case it is important, as stated before, to identity the social identities of others in order to implement the social bias. But, in addition to that, it is important to establish beliefs about the personality, intentions, plans and strategies of others; these will support predictions of behaviour of others that allow agents to adapt their own behaviour and cope better with the social situation.

As a summary, agents should have the ability to take into account social identity, anticipate others and behave rationally in order to perform as expected in complex social situations, with different social groups, and be perceived as believable.

2. CONCLUDING REMARKS

With this paper we want to raise awareness for the fact that Social Identity is central to social behaviour. It has great impact in a wide range of fields and settings, such as prejudice, stereotyping, cooperation and competition, among several other interesting group phenomena. As such we believe that Social Identity theory not only should be considered but is also of great importance for the creation of agents with believable behaviour. So in order to achieve believable social situated agents that interact in complex social situations with humans and other agents, agents should not only take in consideration themselves and others as a set of individuals but also as group members with shared interests, values and goals. In addition, we believe that social dilemmas present interesting social situations in which agents' believability may be studied. In these situations, achieving a good balance between rational choice and bias of socio-emotional nature can be crucial to achieve believable behaviour.

3. ACKNOWLEDGMENTS

This work was partially supported by the INVITE project[1] (ref. UTA-Est/MAI/0008/2009) funded by FCT under the UT-Austin/Portugal cooperation agreement and by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds.

4. REFERENCES

[1] J. Bates. The nature of characters in interactive worlds and the oz project. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1992.

[2] R. Bates. Social dilemmas and rational individuals. *The new institutional economics and Third World development*, pages 27–48, 1995.

[3] R. Brown and G. Ross. The battle for acceptance: An investigation into the dynamics of intergroup behavior. *Social identity and intergroup relations*, pages 155–178, 1982.

[4] C. Castelfranchi. Modelling social action for ai agents. *Artificial Intelligence*, 103(Issues 1-2):157–182, 1998.

[5] K. McLeish and R. Oxoby. *Identity, cooperation, and punishment.* IZA, 2007.

[6] B. Reeves and C. Nass. *The Media Equation. How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge University Press, 1998.

[7] H. Tajfel. La catégorisation sociale (social categorization). *Introduction à la psychologie sociale*, 1:272–302, 1972.

[8] F. Thomas and O. Johnston. *The Illusion of Life: Disney Animation.* Hyperion Press, 1997.

[1] http://project-invite.eu/

On-the-fly behavior coordination for interactive virtual agents – A model for learning, recognizing and reproducing hand-arm gestures online

(Extended Abstract)

Ulf Großekathöfer, Nils-Christian Wöhler, Thomas Hermann, Stefan Kopp
Center of Excellence Cognitive Interaction Technology, Bielefeld University, Germany
{ugrossek, nwoehler, thermann, skopp}@techfak.uni-bielefeld.de

ABSTRACT

In human conversation, verbal and nonverbal behaviors are coordinated by the interlocutors on the fly. To participate in this, artificial conversational agents must be able to create, adopt, and adjust behaviors flexibly and autonomously. We present a novel approach to learning behavioral patterns online, Ordered Means Models (OMMs), that meets the demands of dynamic behavior coordination in interaction. We describe how OMMs enable the virtual agent VINCE to engage in playing Rock-Papers-Scissors games, in which he learns, adapts to, and recognizes every human opponent's gestures on-the-fly such that he becomes unbeatable after only a few rounds. An evaluation study demonstrating this is presented.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

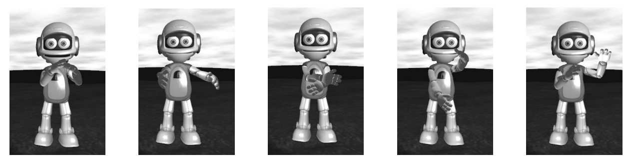
Keywords

Ordered Means Models, Vince, Anytime Classification

1. INTRODUCTION

When interacting with one another, humans rely on a variety of expressive behaviors including words, gestures, facial expressions, or gaze. Being able to interact with others thus requires to perceive, recognize, and interpret such behaviors, as well as to generate and employ them purposefully to fulfill communicative intentions. To act autonomously in communication settings hence implies a number of requirements for virtual conversational agents: (1) fast and on-the-fly recognition, i.e. the ability to create first hypotheses even from partial, ongoing observation; (2) online learning and adaptation, i.e. the ability to learn a behavior from few presentations, and to adjust learned models incrementally to further observations of this behavior; (3) reproduction and generation, i.e. the ability to perform a learned or adapted behavior, e.g., to align to an observed behavior [2, 3, 4, 5]. In this paper we present an approach to endow interactive virtual agents with these abilities for the case of hand-arm gestures. The basic idea is to treat gestures as multivariate time series, e.g. location coordinates of body parts that evolve over time, and to develop a machine learning approach that meets these critical requirements of *on-the-fly recognition capabilities*, *online adaptability*, and *reproduction capabilities*.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.



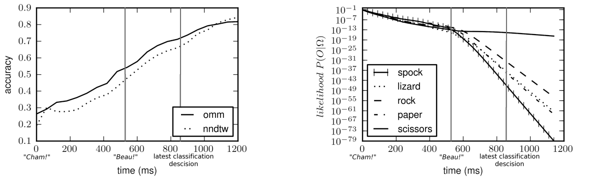
(a) rock (b) paper (c) scissors (d) lizard (e) spock

Figure 1: Screenshots of VINCE performing the learned OMM prototypes for the five classes.

2. ORDERED MEANS MODELS

We use a novel approach to on-the-fly classification of time series, which we refer to as Ordered Means Models (OMMs). OMMs are generative probabilistic state space models that emit a sequence of observation vectors out of K fully connected, hidden states. Thereby, and as opposed to HMMs, OMMs do not include any transition probabilities between states, leading to a simple model architecture. The network of model states follows a left-to-right topology, i.e. OMMs only allow any transition to states with equal or higher indices as compared to the current state. The emissions of each state are modeled as probability distributions and are assumed to be Gaussian. The standard deviation parameter is identical for all states and is used as a *global hyperparameter*. In order to estimate particular model parameters from a set of observations we maximize the complete data log-likelihood by means of an EM-algorithm with respect to the mean vectors. The process of parameter estimation as well as the computation of production likelihood can be achieved efficiently by dynamic programming. This dynamic programming scheme also allows an on-the-fly and incremental evaluation of time series, i.e., sample-by-sample as they are observed. To use OMMs for classification, i.e. to assign a new gesture trajectory to one of J classes, we assume that J class-specific models have been estimated before from the available data. An unknown gesture then is assigned to the class associated with the model that yields the highest production likelihood of all models.

Given the above-mentioned model architecture, an OMM is completely defined by an ordered sequence of reference vec-



(a) Change of recognition accuracies of $\text{OMM}_{\text{on-the-fly}}$ and NN_{DTW} classifiers during observation of gestures.

(b) Likelihoods of class-related $\text{OMM}_{\text{on-the-fly}}$ models during observation of a *spock* gesture.

tors, i.e. the expectation values of the emission distributions. Since these values are elements of the same data space as the observed data examples, the series of reference vectors is fully interpretable and reproducible as a time series prototype.

3. SCENARIO & EVALUATION

In order to evaluate the proposed approach, we realized an extended version of the rock-paper-scissors game for a human player and the virtual agent VINCE [1], adding two extra gestures to the game, a "lizard" and a "spock" gesture: *rock-paper-scissors-lizard-spock (RPSLS)*. In our setup, a Microsoft Kinect$^{\text{TM}}$ sensor captures the scene in 3D, in which we extract a human skeleton for a present user by means of the OpenNI[1] library. The agent does an initial counting phase to sync with the player, but instead of performing a pre-chosen gesture, VINCE tries to recognize as rapidly as possible the gesture of the human player and then to perform a corresponding winning gesture. For recognizing the gestures, VINCE uses an $\text{OMM}_{\text{on-the-fly}}$-based classifier which returns a classification decision if the likelihood ratio between the most-likely and the second-most-likely OMM exceeded a value of 2 or, latest, $310ms$ after "Beau!". In result, the user gets the impression that VINCE presents his gesture without noticeable delay.

After each trial, the user has to name the gesture she just performed. Using this information, the agent learns and adapts to the particular user-specific way of performing the RPSLS-gestures. The game begins with an unlearned classifier and, thus, the human or the agent will win by chance. The classifier is then re-trained after each turn from all data collected so far. Hence, Vince's abilities to predict the gestures of the player improve rapidly during the course of the game. Further, VINCE uses the learned OMM prototypes to generate gestures during the game himself, thus reproducing the observed behaviors and coordinating with the user. Before a model is available, i.e. before the user presented a gesture to VINCE, we use pre-recorded gesture trajectories.

We conducted an evaluation study with 11 participants who played the game with VINCE until either player reaches a score of 20. We collected a data set containing 439 gestures in five classes and recorded the wrist positions of both arms as location coordinates relative to the user's body center for later analysis. Figure 2(a) gives the results of the comparison of the recognition accuracies achieved with $\text{OMM}_{\text{on-the-fly}}$ and Nearest-Neighbor with dynamic time warping distance measure (NN_{DTW}) classifiers in online classification. As can be seen, recognition accuracy of both classifiers increases with each additional sample available. For complete gesture performances, NN_{DTW} classifiers reach a slightly higher accuracy of ≈ 0.84 in contrast to a recognition rate of ≈ 0.82 for $\text{OMM}_{\text{on-the-fly}}$ classifiers. However, for partial gesture performances, $\text{OMM}_{\text{on-the-fly}}$ classifiers yield up to $\approx 10\%$ (on average $\approx 5\%$) higher recognition rates. This indicates that $\text{OMM}_{\text{on-the-fly}}$ classifiers are well suited for *on-the-fly* recognition of behavior patterns. Figure 2(b) shows how the production likelihoods of the $\text{OMM}_{\text{on-the-fly}}$ models for the five different gestures evolve during observation of an example *spock* gesture. After $\approx 600ms$ (approximately on "Beau!") the model related to class *spock* stably stays on a likelihood level of $\approx 10^{-13}$ while the likelihood associated with the other models decreases to a minimum of $\approx 10^{-79}$. In this case, a recognition of this particular gesture performance is possible $\approx 600ms$ after the gesture performance begins, i.e. in synchronization with the gesture presentation. The learning curves show that, at the first turn with unlearned nor adapted OMMs, VINCE is almost completely unable to recognize a gesture the human player performs (recognition accuracy of $\approx 0\%$). Over the first 10-15 rounds, the recognition rate increases up to an average of $\approx 85\%$ demonstrating the rapid learning ability of the used $\text{OMM}_{\text{on-the-fly}}$ classifiers. In total, VINCE managed to win all 11 games with a lead of at least 3 points.

4. CONCLUSIONS

We have proposed Ordered Means Models as a specific kind of probabilistic state-based model that can provide rapid learning, efficient processing, on-the-fly classification, and prototype reproduction. The results from the Rock-Paper-Scissors game scenario demonstrate that OMMs can meet in fact the before-mentioned requirements for fast interpersonal behavior adaptation and coordination. Future work will test how OMMs perform when confronted with natural, more variable communicative gestures and will use OMMs for hierarchical clustering of behavioral patterns.

Acknowledgments

This work was supported by the DFG in the Cluster of Excellence Cognitive Interaction Technology (CITEC).

5. REFERENCES

[1] U. Großekathöfer, A. Sadeghipour, T. Lingner, P. Meinicke, T. Hermann, and S. Kopp. Low Latency Recognition and Reproduction of Natural Gesture Trajectories. In *ICPRAM (Int.Conf. on Pattern Recognition Applications and Methods)*, 2012.

[2] T. Inamura, I. Toshima, and Y. Nakamura. Acquiring motion elements for bidirectional computation of motion recognition and generation. In B. Siciliano and P. Dario, editors, *Experimental Robotics VIII*, volume 5, pages 372–381. Springer-Verlag, 2003.

[3] D. Kulić, W. Takano, and Y. Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761, 2008.

[4] J. Kwon and F. Park. Natural movement generation using hidden markov models and principal components. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(5):1184–1194, 2008.

[5] A. Wilson and A. Bobick. Parametric hidden markov models for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):884–900, 1999.

[1] http://www.openni.org

Live Generation of Interactive Non-Verbal Behaviours (Extended Abstract)

Ken Prepin
LTCI/TSI, Telecom-ParisTech/CNRS,
37-39 rue Dareau,
75014, Paris, France
ken.prepin@telecom-paristech.fr

Catherine Pelachaud
LTCI/TSI, Telecom-ParisTech/CNRS,
37-39 rue Dareau,
75014, Paris, France
catherine.pelachaud@telecom-paristech.fr

ABSTRACT

Psychology, robotic and virtual agents communities commonly claim that to enable natural interaction to take place within dyad of agents, the dyad must be the siege of dynamical coupling: to give to their partners a feeling of natural interaction, interactants be human, robotic or virtual, must be able to make the dynamic of their behaviour emerge both from their own internal states and from their partner's behaviours. However, most virtual agents engines for interactions model communication as a step by step phenomenon, where pre-scripted signals and corresponding feedbacks alternate. We propose here an agent architecture which generates non-verbal behaviours in live, influenced by both the internal state of the agent and the continuously incoming reaction of its partner. This architecture enables an agent facing either another agent or a human, to emphasise shared behaviours (called *Snowball effect*), to decrease un-shared behaviours as well as to align dynamically with its partner's behaviour.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems

General Terms

Theory

Keywords

Human-robot/agent interaction, Peer to peer coordination, Emergent behavior, Modeling the dynamics of MAS, Agent commitments

1. MODEL PRINCIPLES

During a dyadic interaction, partners' behaviours are influenced by both their internal state and the continuously incoming reactions of their partner. When a behaviour is triggered (a smile, a head-nod) how its dynamics will develop through time is not defined ahead of time: decay, emphasis or complete change of the behaviour will depend on the course of the interaction, influenced by the live partner's reactions.

In virtual agent systems, such a dynamical coupling capability is still lacking even though it is necessary for the occurrence of natural interactions [9]. Implementing coupling capabilities require to deal with dynamic adaptation of agent animation in real-time [3, 4, 6].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The model of live generation of interactive non-verbal behaviours that we propose, is implemented in a Neural Network (NN) simulator (Leto/Prometheus (developed in the ETIS lab. by Gaussier et al. [2])) interfaced with a virtual agent engine (the listener agent Greta and its backchannels engine proposed by Bevacqua and Pelachaud [1] and implemented in the SEMAINE platform for Sensitive Artificial Listners (SAL) [8]).

We propose here an agent architecture which generates non-verbal behaviours (head movements and multimodal sequential expressions) *on the fly*, influenced by both the internal state of the agent and the continuously incoming reaction of its partner. The resulting behaviour of a dyad of agents having such an architecture is a *snowball effect* on shared behaviours (when coupling occurs), a decay of not-shared behaviours (when coupling is disrupted), the ability for the two agents involved in the interaction to evaluate their partners engagement by detecting *snowball effects*.

Our model relies on the three properties of every natural communication described bellow:

P1 - Interaction feedbacks modify the course of actions *on the fly*. If the feedbacks from interaction partner are not fast enough regarding the action length, coupling and synchrony cannot occur between partners [6]. In our architecture, the agent which performs the action can have feedbacks concerning this action while s/he is performing the action: the action is commonly built by agent's intentions and partner's feedbacks.

P2 - Perception-Action mapping. There is a natural/structural tendency to imitate the other and to better perceive the other when s/he imitates you.
This mapping has two components: first a default mirror mapping; second a mapping between different actions (cf. *backchannels* [3]).

P3 - Action are built as interpolated sequences of basic elements. Niewiadomski et al. [5] indicates that to be able to convey subtle emotional states, Multimodal Sequential Expressions are needed.

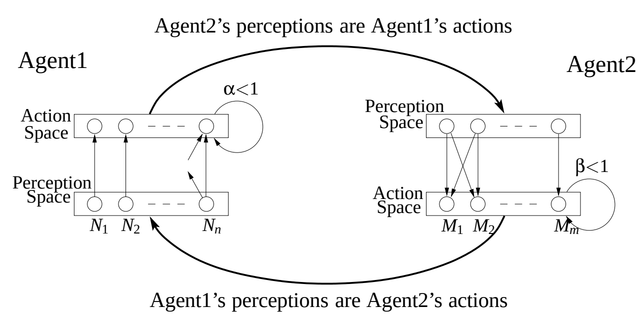


Figure 1: Scheme of the interactive loop within the dyad.

A *Snowball Effect* on shared actions and a decrease of non-shared actions result from P1 and P2 (see Fig.1 for a principle scheme of a dyad of agents implementing P1 and P2).

2. SNOWBALL EFFECT

We assume here and model the fact that during dialogue, sequences of emotional signals (P3) are induced by the coupling and the mutual reinforcement occurring between agents (P1 and P2). Let us consider the example of polite vs friendly smile: if you smile politely to somebody who smiles back at you but more friendly, without interruption, your smile could evolve in a friendly smile as well (see Fig.2).

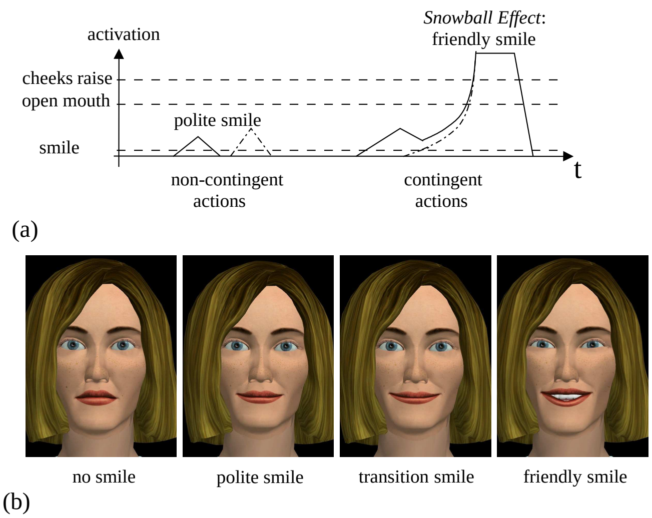


Figure 2: Snowball effect of shared actions. (a) Example of dynamics which would be obtained for smile depending on agents'action contingency. Solid line is for Agent1's activation of smile, dotted line is for Agent2's (b) Politeness smile to amusement smile transition on one of the two agents: frames are generated *on the fly*.

We model P3 enabling agents to recruit a sequence of basic facial signals *on the fly*, depending on the emphasize of their own actions induced by P1 and P2: in our example when facial signals corresponding to a *polite smile* are emphasized enough, new signals, corresponding to *friendly smile* are triggered.
The sequence of emotional signals is not pre-defined but is mainly induced by the mutual reinforcement between agents: the mutual reinforcement occurring when agents perform contingent actions, leads to a specific shared sequence of actions(see Fig.2(a)).

Fig.2(b) shows what the animation from polite to amusement smile looks like with our architecture for live generation of behaviour: the frames are computed *on the fly*, they are structurally building a single continuous behaviour.

3. CONCLUSION

The resulting behaviour of a dyad of agents having such an architecture (i.e. implementing of the three properties P1, interaction feedbacks modify the course of actions *on the fly*; P2, perception-action mapping; P3, action are built as sequences of basic elements) is a *snowball effect* on shared behaviours and a decrease of non-shared behaviours: perception directly influences action; actions last long enough to enable several perceptions to influence them.

This behaviour of the dyad has a direct impact on the interactants' sensitivity to their rapport: *snowball effect* occurs only if interactants share a common state and if they are both aware of each other (cross-perception). That makes the occurrence of the snowball effect a marker of the dyadic state and particularly of partners contingency [7]. To know if whether or not they have an effective interaction, agents can directly observe their own sequential expressions dynamics: novelty and complex sequences is equivalent to high coupling and exchange between partners.

At short term, our aim is to test this architecture between an agent and a human. Snowball effect would be enabled giving to interactants a cue of their level of contingency.

Demos of the *snowball effect* obtained with our architecture an be seen on:
`http://www.tsi.telecom-paristech.fr/mm/?p=778`.

Acknowledgements

This work has been partially financed by the European Project NoE SSPNet, the project SYNC from Institut Télécom and the European Project VERVE. Nothing could have been done without the Leto/Prometheus NN simulator, lend by the Philippe Gaussier's team (ETIS lab, Cergy-Pontoise, France).

4. REFERENCES

[1] E. Bevacqua, S. Hyniewska, and C. Pelachaud. Positive influence of smile backchannels in ecas. In *International Workshop on Interacting with ECAs as Virtual Characters (AAMAS 2010)*, Toronto, Canada, Oct. 2010.

[2] P. Gaussier and S. Zrehen. Avoiding the world model trap: An acting robot does not need to be so smart! *Journal of Robotics and Computer-Integrated Manufacturing*, 11(4):279–286, 1994.

[3] J. Gratch, A. Okhmatovskaia, F. Lamothe, S. Marsella, M. Morales, R. J. Werf, and L.-P. Morency. Virtual rapport. In *Proceedings of 6th International Conference on Intelligent Virtual Agents:IVA'06*, pages 14–27, Sept 2006.

[4] G. Jonsdottir, K. Thorisson, and E. Nivel. Learning smooth, human-like turntaking in realtime dialogue. In H. Prendinger, J. Lester, and M. Ishizuka, editors, *Intelligent Virtual Agents*, volume 5208 of *Lecture Notes in Computer Science*, pages 162–175. Springer Berlin / Heidelberg, 2008.

[5] R. Niewiadomski, S. Hyniewska, and C. Pelachaud. Constraint-based model for synthesis of multimodal sequential expressions of emotions. *IEEE Transactions of Affective Computing*, 2(99):1–13, 2011.

[6] K. Prepin and C. Pelachaud. Effect of time delays on agents' interaction dynamics. In Tumer, Yolum, Sonenberg, and Stone, editors, *The Tenth International Conference on Autonomous Agents and Multi Agents Systems, AAMAS2011*, pages 1–8. IFAAMAS, 2011.

[7] K. Prepin and C. Pelachaud. Shared understanding and synchrony emergence: Synchrony as an indice of the exchange of meaning between dialog partners. In J. Filipe, editor, *Third International Conference on Agents and Artificial Intelligence, ICAART2011*, pages 1–10. Springer, 2011.

[8] M. Schröder. The semaine api: Towards a standards-based framework for building emotion-oriented systems. *Advances in Human-Computer Interaction*, 2010:21, 2010.

[9] F. J. Varela. *Invitation aux sciences cognitives*. Editions du Seuil, Paris, 1995.

Agent Communication for Believable Human-Like Interactions between Virtual Characters

(Extended Abstract)

Joost van Oijen
Utrecht University
PO Box 80.089, 3508 TB
Utrecht, the Netherlands
J.vanOijen@uu.nl

Frank Dignum
Utrecht University
PO Box 80.089, 3508 TB
Utrecht, the Netherlands
F.P.M.Dignum@uu.nl

ABSTRACT

In this paper we present a model for realizing believable human-like interaction between virtual agents situated cognitively in a MAS on one side while embodied in a virtual environment within a game engine on the other side. A middleware approach is taken to facilitate such agents in communication, hereby making a tradeoff between efficiency and believability while taking into account the real-time requirements of games and simulations.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents, Multiagent Systems*
; I.6.3 [**Simulation and Modeling**]: Applications

General Terms

Design, Human Factors

Keywords

Agent Communication, Intelligent Virtual Agents

1. INTRODUCTION

The use of agent technology in the form of multi-agent systems (MASs) seems a good fit to realize the cognitive and decision-making aspects of an Intelligent Virtual Agent (IVA). One of the problems one faces when applying a MAS to control the behavior of virtual characters is how to deal with agent communication in the MAS. Unlike in typical MASs where agents communicate using standard protocols (e.g. FIPA) and mediums (e.g. TCP/IP), agents now become embodied in a real-time virtual environment where they have to resort to the expression and perception of communicative behavior through their embodiment in order to interact in a human-like manner.

In current 3D video games or virtual worlds, human-like interaction between virtual characters has hardly been employed. When it is, it is often realized during so-called cut scenes or in specific situations that are known to occur by design. Since the dialog acts and the context in which the interaction takes place are fully known beforehand, realization can be crafted in detail at design time. Now when we turn to agent technology to design autonomous, goal-directed agents controlling virtual characters, the context in which they might communicate cannot be known beforehand. Hence, communication should be realized dynamically at runtime.

2. CONCEPTUAL GAP

Using MAS technology to control human-like characters with communicative abilities, one has to bridge the inherent conceptual gap between typical agent communication in MASs and human-like communication realized in a virtual environment. This introduces several design issues:

- Agents become embodied and have to resort to the expression and perception of *multimodal* behaviors. The choice and interpretation of these behaviors may depend on a certain *context* (e.g. an agent's identity, its affective state or its beliefs about interlocutors and the social situation).
- Agents become situated in a real-time virtual environment and have to deal with the durative nature of the expression and perception of communicative intents (e.g. monitoring, ability to interrupt, and awareness and interpretation of perceived behavior). Additionally, believable perception should be enforced based on an agent's sensory capabilities and environment physics.

Besides these issues, additional aspects related to natural human-like communication should be considered. E.g. (1) other types of functions besides the common communicative acts typical in agent communication (e.g. meta-conversational, deictic or affective functions), (2) more flexible interaction protocols to simulate natural human-like conversations (e.g. [2]) and (3) the ability to perform listening behaviors and provide backchannel feedback.

3. A MIDDLEWARE APPROACH

We present a model for human-like communication to fill the conceptual gap between agent communication in a MAS on one side and its realization in a virtual environment on the other side, covering the mind-body interface between an agent and its embodiment. The model is illustrated in

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

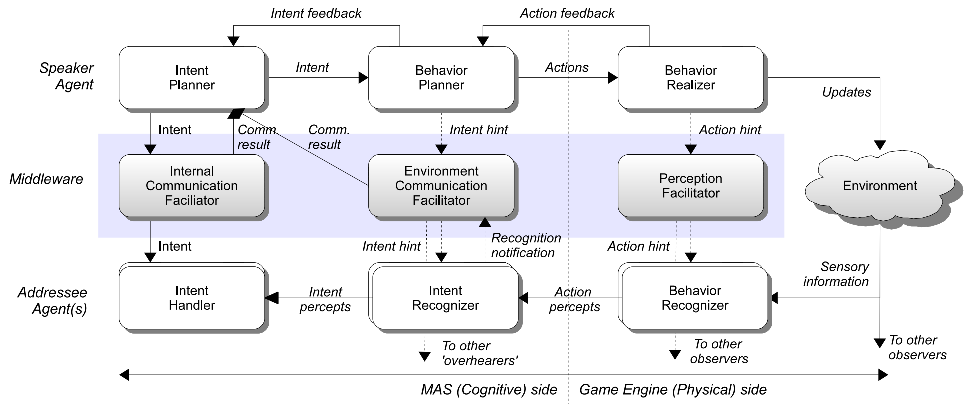


Figure 1: Virtual Agent Communication Model

figure 1 and is designed to tackle the issues described above in an efficient way without loosing believability.

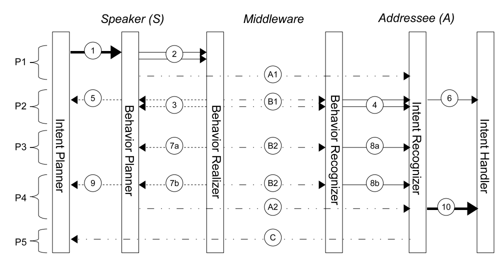
First, the upper part of the model is responsible for realizing a communicative intent generated by a speaker agent using multimodal behavior expressions (similar to the SAIBA framework [1]). Second, the lower part of the model deals with the perception of communicative intents by addressee agents (or overhearers), based on the observation of multimodal behavior expressions. These perception stages can be computationally heavy processes and contribute to design complexity: *behavior recognition* would require observations over time to recognize communicative signals like speech (e.g. stream of sound waves) or gestures (e.g. motion of bones). *Intent recognition* could be seen as a pattern matching problem to match a set of communicative signals to an intent. Although resulting in a fully autonomous process for the perception of communicative intents, we believe this approach is not very practical to implement and is overly complex for use in real-time games.

As an alternative, we propose a design approach employing a middleware layer to facilitate communication between agents. It allows agents to (1) perceive communicative actions and intents without the need to interpret them from sensory information, (2) be notified about the successful recognition of an intent by a receiver agent and (3) perform internal MAS communication (eliminating behavior generation and perception states).

To clarify the communication process within our model, figure 2 illustrates the successful communication of a single communicative intent, realized using multimodal behavior consisting of two actions. To compare, a typical MAS communication using a direct transportation medium (e.g. FIPA) would be represented merely by line 1 and 10 in the example. Our model proposes an extension to cover the more complex medium that would be required for virtual agents, including (1) the cognitive abilities of agents to express and interpret intents, (2) their physical abilities to express and perceive communicative behavior and (3) a transportation medium represented by a virtual environment.

4. CONCLUSION

The proposed design approach for modeling agent communication allows virtual agents to communicate their intents efficiently on the MAS side and realize this in a human-like manner on the game engine side. Believable perception can be enforced through a virtual environment simulating a real-world medium. The primary contribution is the introduction of a middleware layer to simplify the perception stages for communicative actions and intents. We belief with this more practical approach a proper balance between efficiency and believability can be achieved for agent-based human-like communication, suitable for real-time games.



#	Agent	Activity	Message
1	S	schedule intent	*communicate(id=i1,content=inform_child_in_house)*
2	S	schedule action	*speech(id=a1,resource=child_in_house.mp3)*
3	S	receive action feedback	*action_feedback(action=a1,state=started)*
4	A	perceive action	*action_percept(action=a1,state=started)*
5	S	receive intent feedback	*intent_feedback(id=i1,state=started)*
6	A	perceive intent (no content)	*intent_percept(state=started)*
7	S	receive action feedback	*action_feedback(action=a1,state=finsihed)*
8	A	perceive action	*action_percept(action=a1,state=finished)*
9	S	receive intent feedback	*intent_percept(id=i1state=finished)*
10	A	perceive intent	*intent_percept(id=i1,state=ended)*

#	Middleware	Activity	Message
A1	Comm. Facilitator	send intent hint	*intent_hint(intent=i1, actions=a1;a2)(state=started)*
B1	Perception Facilitator	send action hint	*action_percept(action=a1,state=started)*
B2	Perception Facilitator	send action hint	*action_percept(action=a1,state=finished)*
A2	Comm. Facilitator	send intent hint	*intent_hint(intent=i1)(state=finished)*
C	Comm. Facilitator	send communication result	*communication_result(id=i1,observed_by=A)*

Figure 2: Example Communication

5. REFERENCES

[1] S. Kopp and et al. Towards a common framework for multimodal generation: The behavior markup language. In J. Gratch, M. Young, R. Aylett, D. Ballin, and P. Olivier, editors, *Intelligent Virtual Agents*, volume 4133 of *LNCS*, pages 205–217. 2006.

[2] D. Traum, W. Swartout, J. Gratch, and S. Marsella. A virtual human dialogue model for non-team interaction. In L. Dybkjær and et al., editors, *Recent Trends in Discourse and Dialogue*, volume 39 of *Text, Speech and Language Technology*, pages 45–67. Springer, 2008.

A BDI Dialogue Agent for Social Support: Specification of Verbal Support Types

(Extended Abstract)

J.M. van der Zwaan
Delft University of Technology
Jaffalaan 5
2628 BX Delft, The Netherlands
j.m.vanderzwaan@tudelft.nl

V. Dignum
Delft University of Technology
Jaffalaan 5
2628 BX Delft, The Netherlands
m.v.dignum@tudelft.nl

C.M. Jonker
Delft University of Technology
Mekelweg 4
2628 CD Delft, The Netherlands
c.m.jonker@tudelft.nl

ABSTRACT

An important task for empathic agents is to provide social support, that is, to help alleviate emotional distress. In this paper, we select five types of verbal social support (sympathy, compliment, encouragement, advice, and teaching) and present our implementation in a prototype dialogue agent.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms

Design, Experimentation

Keywords

Conversational agents, Verbal and non-verbal expression, Modeling cognition and socio-cultural behavior

1. INTRODUCTION

Social support or comforting refers to communicative attempts to alleviate the emotional distress of another person. Recent developments in affective computing show that empathic agents are increasingly capable of complex social and emotional dialogues, but so far they do not have the ability to comfort users. In our research, we explore how and to what extent Embodied Conversational Agents (ECAs) can provide social support.

Recently, we proposed a design for an ECA that tries to comfort children who are bullied online [5]. Interaction between the agent and the user takes place in two main stages: 1) Gather information about the current situation, 2) Give advice on how to deal with the situation. The agent uses different (verbal and non-verbal) social support strategies. This paper is focused on the dialogue engine of this agent, i.e. verbal strategies for social support. The embodiment and non-verbal behavior of the agent are beyond the scope of this paper.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. SOCIAL SUPPORT

The verbal social support actions presented in this paper are based a typology of social support in online settings [1]. This typology is relevant for the agent, because online communication is mostly textual and does not depend on additional communication channels (such as non-verbal behavior and auditory information). The five main support categories are [1]:

- Information support (messages that convey instructions),
- Tangible assistance (offers to take concrete, physical action in support of the recipient),
- Network support (messages that appear to broaden the recipient's social network),
- Esteem support (messages that validate the recipient's self-concept, importance, competence, and rights as a person), and
- Emotional support (attempts by the sender to express empathy, support the emotional expressions of the recipient, or reciprocate emotion)

Each category breaks down into multiple subtypes. Five subtypes that frequently occurred in counseling conversations by chat [3] were selected to be implemented; that is sympathy, compliment, encouragement, advice and teaching. Table 1 lists descriptions and examples of these support types.

3. THE AGENT

A prototype of the social support dialogue agent was implemented in GOAL, a high level agent programming language [4]. The agent's reasoning engine is modeled according to the Belief-Desire-Intention (BDI) paradigm [2]. This means the agent has beliefs (e.g., about what advice to give in which situations), goals (e.g., to give social support), and plans (e.g., to gather information about the upsetting situation and to give advice after all relevant information is gathered).

The agent and the user communicate through natural language text messages. Given the complexity of interpreting and generating natural language, in the current system, text interpretation and generation have not been implemented.

Support type	Description	Example
Sympathy	Express feelings of compassion or concern	How awful that you are being bullied!
Encouragement	Provide recipient with hope and confidence	I know you can do it!
Compliment	Positive assessments of the recipient and his or her abilities	Good of you to have told your parents!
Advice	Suggestions for coping with a problem	Perhaps you should tell your parents.
Teaching	Factual or technical information	You can block a contact by clicking the 'block' button.

Table 1: The types of social support implemented in the dialogue agent.

Instead, we use logical representations of the contents of utterances (speech acts), for example an utterance such as *I'm being cyberbullied!* is represented by `send(agent, inform, incident(type, cyberbullying))`.

The agent's knowledge is stored in its belief bases. The agent has beliefs regarding the domain (e.g., what questions to ask the user and what advice to give in different situations), social support (e.g., when to give which type of social support), and conversation management (e.g., how to open and close conversations). The contents of the speech acts (and thus of the conversation) are defined by the contents of the belief base.

In the reasoning engine, beliefs are combined to select speech acts the agent will utter.

4. SPECIFICATION OF SUPPORT TYPES

To illustrate the implementation of the social support types, we explain how sympathy was implemented. The information gathering phase of the conversation consists of a recurring pattern of the agent asking a question, the user answering that question, and the agent acknowledging the answer. An acknowledgement is either neutral (e.g., *Okay*) or sympathetic. The agent only expresses sympathy if it follows from his beliefs sympathy is applicable, otherwise it plays safe by staying neutral. The following example shows how sympathetic acknowledgement works:

Agent: *Can you tell me what happened?*
User: *Someone is calling me names on msn*

The user's utterance causes addition of the following `incident` facts to the agent's belief base:

```
incident(type_cb, name_calling).
incident(method_cb, msn).
```

Based on the following rule in the belief base:

```
sympathetic_acknowl(type_cb, name_calling) :-
    incident(type_cb, name_calling).
```

the agent responds sympathetically to the user:

Agent: *That's awful!* (sympathy)

Absence of the `sympathetic_acknowl` rule would have resulted in a neutral acknowledgement of the user's input:

Agent: *I see* (acknowledgment)

The other support types have been implemented in a similar manner. Like sympathy, compliment and encouragement occur in response to the answers the user gives to questions of the agent. Advice and teaching are uttered pro-actively, after the agent gathered sufficient information (this depends on domain knowledge). For advice that requires an explanation, the agent optionally teaches the user how to execute the advice. After giving advice, the agent waits for confirmation from the user. Once the user has confirmed, it moves on to the next piece of advice, or closes the conversation.

5. CONCLUSION

In this paper, we presented five verbal social support types: sympathy, compliment, encouragement, advice, and teaching; and implemented them a BDI dialogue agent. Sympathy, compliment and encouragement are always given in response to user input. Advice and teaching are offered proactively. Whether the agent performs a social support action depends on its beliefs, which, in turn, are determined by domain knowledge.

For future work, we plan to implement more types of support from Braithwaite's typology. In particular empathy is important in supportive communication. To appear empathic, the agent needs the capability to reason about emotions. Therefore, an emotional module will be added to the agent.

6. ACKNOWLEDGEMENTS

This work is funded by NWO under the Responsible Innovation (RI) program via the project 'Empowering and Protecting Children and Adolescents Against Cyberbullying'.

7. REFERENCES

[1] D.O. Braithwaite, V.R. Waldron, and J. Finn. Communication of social support in computer-mediated groups for people with disabilities. *Health Communication*, 11(2):123–151, 1999.

[2] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213 – 261, 1990.

[3] R. Fukkink. Peer counseling in an online chat service: A content analysis of social support. *Cyberpsychology, Behavior, and Social Networking*, 14(4):247–251, 2011.

[4] K.V. Hindriks. Programming Rational Agents in GOAL. In A. El Fallah Seghrouchni, J. Dix, M. Dastani, and R.H. Bordini, editors, *Multi-Agent Programming:Languages, Tools and Applications*, volume 2, pages 119–157. Springer US, 2009.

[5] J.M. van der Zwaan, V. Dignum, and C.M. Jonker. Simulating peer support for victims of cyberbullying. In *Proceedings of the 22st Benelux Conference on Artificial Intelligence (BNAIC 2010)*, 2010.

An Agent-based Annotation Model for Narrative Media

(Extended Abstract)

Mario Cataldi, Rossana Damiano, Vincenzo Lombardo, Antonio Pizzo
CIRMA and Dipartimento di Informatica, Università di Torino
{cataldi,rossana,vincenzo}@di.unito.it

ABSTRACT

In this paper, we present an agent-based annotation model for narrative media. This model borrows from agent theories to describe the behavior of characters in stories, with the long-term goal of building annotated resources for the evaluation, design and editing of virtual agents.

Categories and Subject Descriptors

J.5 [**ARTS AND HUMANITIES**]; I.2.m [**ARTIFICIAL INTELLIGENCE**]: Miscellaneous; I.2.1 [**ARTIFICIAL INTELLIGENCE**]: Knowledge Representation Formalisms and Methods

General Terms

Management, Theory, Human Factors

Keywords

Computational models of story, BDI model, Semantic annotation

1. INTRODUCTION

Stories contain a huge repository of engaging behaviors, designed by authors with originality and creativity. Dramatic stories in particular are a source of well tested connections between goals and actions, due to the need of creating motivated behaviors. In fact, a character's action is believable only if rooted in a deliberative and emotional process [4, 5].

In this paper, we present an agent-based annotation model for narrative media, designed with the goal of building annotated resources for the specification, design and evaluation of virtual agents. In the perspective of interoperability with agent systems, the annotation model borrows from the basics of the BDI agent model, and integrates them with emotions and values [2, 7] The model and the related annotation system are part of the Cadmos (Character-based Annotation of Dramatic Media ObjectS) project.[1]

[1]Funded by Regione Piemonte, Poli di Innovazione, POR-FESR 2007-2013.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ENCODING CHARACTERS IN STORIES

The annotation of stories relies on the basic assumption that a narrative media object, be it a screenplay text or a video fragment, can be segmented into narratively meaningful units, and that these units can be semantically annotated for subsequent retrieval and reuse. The annotated units, then, can be searched and navigated based on a formal description of the characters' behavior: what actions they do and why.

Drama, a "cultural object" developed along two millennia, is based upon the Aristotelian notion of action (drama as "imitation of praxis"), where *emotional characters* engage in *conflicts* that necessarily arise from their deliberative processes [3]. The characters' motivations are mainly rooted in their moral values, put at stake by the dramatic premise of stories [9].

The top level of the ontology consists of five main classes (see Figure 1: `Unit`, `Entity`, `Dynamics`, `Relation` and `DescriptionTemplates`). `Unit` is the core of the annotation, since it models the discretization of the story into fragments, that bring about some relevant change in the story world through unintentional events or characters' actions, and are actualized in some media object (text, video, etc.). Units can have different granularity, so that, along with a horizontal organization of units into sequences, a vertical organization of the units in a hierarchical structure can emerge in stories. When goals are in conflict, the unit becomes a `DramaUnit`. The `Dynamics` of drama encompasses the occurrence of incidents (`Action` and `Event` calsses) and the states (`State` class) that result from them. States occur both in the story world and in the mental states of the characters. Characters' motivations and emotional states are modeled by the `MentalState` class, further subdivided in `Belief`, `Goal`, `Emotion` and `Value`. The `Goal` class is further specialized into goal types to account for the goal taxonomy described by [8]. The `Relation` class encompasses the structural relationships among units (`StructuralRelation` class) and the qualities of agents and objects in a specific Unit (`DramaRelationType` class). Finally, the `DescriptionTemplates` class establishes the connection of the representation of incidents with some external representation of processes in some reference ontology and in some natural language lexicon.

In Figure 2 we illustrate the use of the annotation model by resorting to a well-known example, the Aeschylus' *The Suppliants*. In this tragedy, the Danais' daughters flee to Argo and implore King Pelasgus for shelter, so as to be free from the obligation to marry Aegyptus' sons. The Danais are modeled here a collective subject (through the `Agent`

Figure 1: The ontology of story and character.

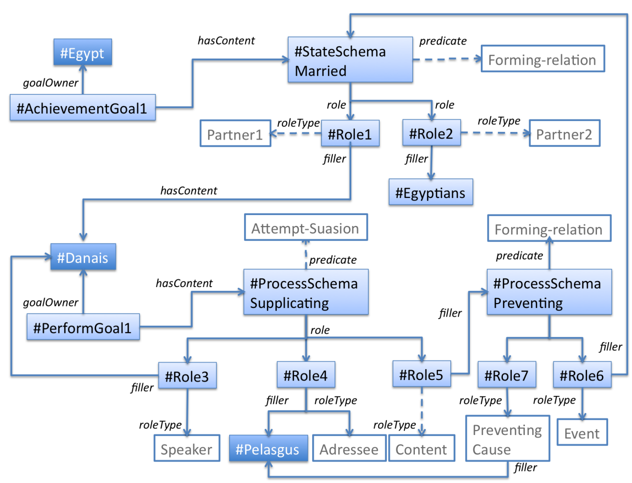


Figure 2: The annotation of the dramatic situation realized in Aeschylus' *The Suppliants*.

class). Their goal is a perform goal (`PerformGoal1`) and its content is described as the act of "supplicating" (through the `ProcessSchema` class). The content of both goals (included Aegypt's achievement goal that the Danais' daughters marry his sons, `AchievementGoal1`) has been described by using FrameNet: for each frame, its FrameElements are employed to assign `Roles` the participants to the action. This example also shows how the proposed annotation model permits to model complex goals and actions by providing a recursive schema: the goal of the Suppliants' action is that another character, Pelasgus, assumes the goal to shelter them.

The conceptualization of the actions, events and entities involved in units relies on the YAGOSUMO project [6]; their description relies on FrameNet [1].

3. CONCLUSIONS

In this paper, we have proposed the annotation of stories with the behavior of characters as a means to gather a large knowledge base for the validation, specification and testing of virtual agents. The annotation relies on computational ontologies, to allow reasoning over processes and to limit the arbitrariness of the annotation terms.

Future work includes the testing of the annotation model on a larger corpus of narrative works, belonging to different genre and media types.

4. REFERENCES

[1] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *Proceedings of the 36th COLING-ACL*, ACL '98, pages 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[2] R. Damiano and A. Pizzo. Emotions in drama characters and virtual agents. In *AAAI Spring Symposium on Emotion, Personality, and Social Behavior*, 2008.

[3] G. W. F. Hegel. *Lectures on fine art*. Clarendon Press, Oxford, England, 1998.

[4] P. Hogan. *Affective Narratology: The Emotional Structure of Stories*. Frontiers of Narrative. UNP - Nebraska, 2011.

[5] A. R. Mele. Effective reasons and intrinsically motivated actions. *Philosophy and Phenomenological Research*, 48(4):pp. 723–731, 1988.

[6] G. d. Melo, F. Suchanek, and A. Pease. Integrating yago into the suggested upper merged ontology. In *Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence - Volume 01*, pages 190–193, Washington, DC, USA, 2008. IEEE Computer Society.

[7] F. Peinado, M. Cavazza, and D. Pizzi. Revisiting Character-based Affective Storytelling under a Narrative BDI Framework. In *Proc. of ICIDIS08*, Erfurt, Germany, 2008.

[8] M. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in Agent Systems: A Unifying Framework. In *Proceedings of AAMAS'08*, 2008.

[9] S. D. Williams. *The Moral Premise*. Michael Wiese Productions, 2006.

Goal-Driven Approach To Open-Ended Dialogue Management using BDI Agents

(Extended Abstract)

Wilson Wong, Lawrence Cavedon, John Thangarajah and Lin Padgham
School of Computer Science and IT, RMIT University, Melbourne, Australia
{wilson.wong,lawrence.cavedon,john.thangarajah,lin.padgham}@rmit.edu.au

ABSTRACT

We describe a BDI (Belief, Desire, Intention) approach and architecture for a conversational virtual companion embodied as a child's Toy. Our aim is to support both structured conversation-based activities (e.g., story-telling, collaborative games) as well as more free-flowing, engaging dialogue with variation and some unpredictability. We argue that a goal-oriented approach to the agent's conversational capabilities provides these competing capabilities.

Keywords

BDI architecture, Dialogue management, Conversational agent

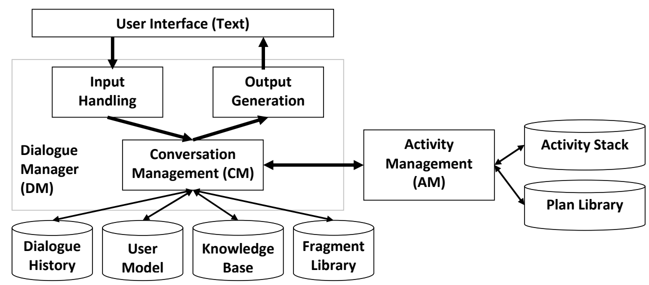


Figure 1: System architecture

1. OVERVIEW

We propose a BDI architecture as shown in Figure 1 for a conversational agent that supports both task-oriented dialogue as well as "chatty" conversations. The BDI agent model has been used successfully in a range of applications requiring a mix of reactive behaviour and goal-directed reasoning, and its design model supports different means for achieving a goal depending on context and other factors [3]. The BDI framework thus allows the conversational agent to select different strategies for satisfying a conversational goal where a conversational goal may involve playing a collaborative game such as a role-play, satisfying a request from the user such as an information request, or simply conversing with the user about a pertinent topic. BDI agent-based approaches to dialogue management have been previously proposed [2]. However, these have typically been for task-oriented conversations where the outcome was to support the user in performing a given task (e.g., accessing email). A significant novelty of our use of the BDI approach is to provide multiple plans to satisfy a given goal (e.g., chat, engage in a shared role-play), and to support variation in the way that goal is then achieved including enforcing variation in the agent's contributions to the conversation. When interacting with the child, the Toy suggests possible *Conversational Activities* such as a cooking game, a story, a quiz, etc. These activities are represented as goal-plan structures, which are a set of plan templates in the Toy's `Plan Library`. These plans are used to guide the different aspects of the activity and the selection of fragments for the Toy to utter in pursuit of that activity. More importantly, the specific utterances are **not** part of the activity structure. The plans can provide contextual information which is then used by the `Conversation Manager` to select the appropriate outputs from the `Fragment Library`. The goal-plan tree which is induced by the `Plan Library` gives a structure that is essentially an AND/OR tree. This provides a large number of possible executions within a relatively compact structure [3]. It is this which we exploit to achieve the desired variability, while retaining a coherent, goal-oriented dialogue. In Figure 2, we show a partial goal-plan tree for a particular activity in the Toy, namely, a cooking role-play activity. The top-level goal has a single plan which guides the struc-

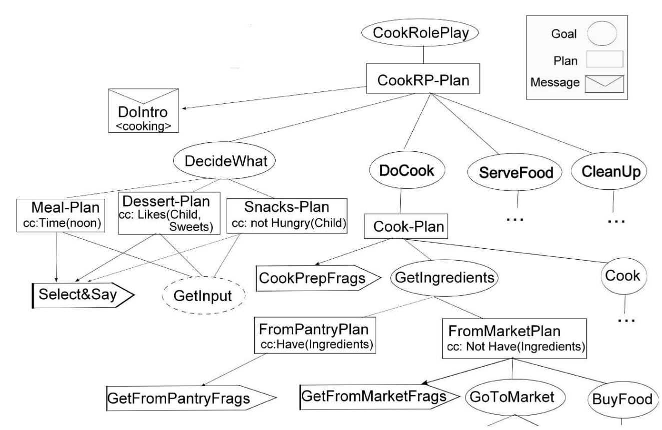


Figure 2: Example activity: Cooking role play

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

ture of the activity. It is possible to have different plans to choose from at the top level to provide more variety. This plan has a sequential set of subgoals, each of which has a set of plans to choose from, and so on. We see that the first goal is `DoIntro` which is a goal which carries information about the current activity (Cooking) and triggers a plan in the `Conversation Manager` to select a suitable introductory fragment for this activity, and prepend it to the next system output (i.e., it is not a fragment with any expected response). Following this is a choice of plans, one of which will be selected in any given conversation. It will then do a number of things such as decide how many interactions to have in this subgoal. Importantly, it will provide some additional keyphrases to be added to the ongoing collection from the dialogue history to assist in fragment selection. It initiates an `Interact` goal which results in the `Conversation Manager` determining an output fragment and analysing the user response, which is then provided back to the plan in the form of keyphrases and a response category. Assuming the response is accepted, when the plan has completed its interactions, it decides (based on the keywords collected) what food it believes is going to be prepared and the activity progresses onto the next subgoal `CookProcess`, which is managed in a similar way. Activity suspension, resumption and abortion, which are the responsibility of the `Activity Manager`, are not discussed here due to space constraints.

2. EXAMPLE AND DISCUSSIONS

Figure 3 shows an example interaction with the Toy that demonstrates a number of the features discussed in the previous sections. We highlight the features of this interaction here. Items 2 to 12 are mainly about the dynamic building of context for fragment selection. Item 13 shows an example of activity selection based on relatedness. Items 13 to 15 illustrates several interactions guided by the goal-plan activity structure. Item 16 is a specific user request which results in the suspension of the current activity. Items 23 to 24 shows the resumption of the suspended activity together with a transition fragment. A brief suspension and resumption is shown in Items 27 to 29. Finally, an activity finishes and a new activity is selected in Item 35.

We performed an initial evaluation of the Toy involving nine human users with a total of 229 system utterances (25 on average) and 162 user inputs (18 on average). The users are asked to rate each of the Toy's responses on a Likert scale of 0 (i.e., *very odd response*) to 4 (i.e., *very sensible response*). The ratings of the majority of the utterances tend towards level 3 (i.e., *sensible*) with a standard deviation between 0.71 to 1.12. At the end of each 10-minute interaction, the users were asked to rate on two 5-point Likert scales, one on overall coherence (*very scattered* versus *very coherent*), and the other regarding variability (*too repetitive* to *good variation*). 7 out of the 9 ratings were on level 3 for *reasonable variation*, with one rating each at levels 2 and 4. However, 4 out of the 9 participants selected level 2 (i.e. *scattered*) for overall coherence. The results were promising even though the coverage of the `Knowledge Base` is currently limited. Our analysis of the evaluation data suggests we could expect increase in overall user satisfaction by improving input processing, extending the amount of content, and improving the coherence measure for new-topic selection.

Our future work includes extending the capabilities listed above together with evaluating the system in the context of

```
01: USER: hi.
02: SYSM: hello. what have you been doing?
03: USER: i just had chocolate.
04: SYSM: do you like chocolate?
05: USER: i love chocolate.
06: SYSM: i love it too. i wish i could eat it
          for breakfast...
07: USER: chocolate for breakfast? yucks!
08: SYSM: what do you eat for breakfast instead
          of chocolate?
09: USER: toast.
10: SYSM: what do you like to eat on your toast.
11: USER: ice cream.
12: SYSM: yum yum! delicious.
13: SYSM: do you want to make some ice cream?
14: USER: yeah.
15: SYSM: do you have the recipe with you?
16: USER: can you please tell me a story?
17: SYSM: sure. susie the ice cream moose was...
...
23: SYSM: that's the end of the story.
24: SYSM: let's continue with our cooking. so, do
          you have the recipe with you?
25: USER: no.
26: SYSM: you'll need milk and chocolate.
27: USER: where does chocolate come from?
28: SYSM: chocolate comes from cacao trees.
29: SYSM: let's continue with our cooking, do you
          need to get out a saucepan or something?
...
34: USER: it's done!
35: SYSM: yum yum! your ice cream is ready.
36: SYSM: now, let's talk about zoo, ok?
37: USER: i like visiting the zoo.
38: SYSM: do you go to the zoo often?
...
```

Figure 3: An example execution trace.

noisy automated speech recognition. We have been mining social forums to automatically extend our `Knowledge Base` [4]. More relevant to the goal-oriented approach, we are developing a model of *proactive engagement*, whereby the conversational agent actively monitors user engagement [1] and applies conversational strategies to increase engagement within the current or a new activity.

Acknowledgment

This work is partially supported by the Australian Research Council and Real Thing Entertainment Pty. Ltd. under Linkage grant number LP0882013.

3. REFERENCES

[1] G. Castellano, A. Pereira, I. Leit, A. Paiva, and P. McOwan. Detecting user engagement with a robot companion using task and social interaction-based features. In *Proceedings of the 11th ICMI*, Cambridge, USA, 2009.

[2] A. Nguyen and W. Wobcke. An agent-based approach to dialogue management in personal assistants. In *Proceedings of the IUI Conference*, San Diego, California, 2005.

[3] L. Padgham and M. Winikoff. *Developing Intelligent Agent Systems: A practical guide*. Wiley Series in Agent Technology. John Wiley and Sons, 2004.

[4] W. Wong, J. Thangarajah, and L. Padgham. Health conversational system based on contextual matching of community-driven question-answer pairs. In *Proceedings of the 20th ACM CIKM (Demo)*, pages 2577–2580, Glasgow, UK, 2011.

Distributed Punishment as a Norm-Signalling Tool

(Extended Abstract)

Daniel Villatoro
IIIA - CSIC
Bellatera, Barcelona, Spain
dvillatoro@iiia.csic.es

Giulia Andrighetto
ISTC - CNR
Rome, Italy
giulia.andrighetto@istc.cnr.it

Jordi Brandts
Business Econ. Dept. (UAB)
IAE - CSIC
jordi.brandts@iae.csic.es

Jordi Sabater-Mir
IIIA - CSIC
Bellatera, Barcelona, Spain
jsabater@iiia.csic.es

Rosaria Conte
ISTC - CNR
Rome, Italy
rosaria.conte@istc.cnr.it

ABSTRACT

Recent studies show that punishment plays a crucial role in favoring and maintaining social order. However, very little attention has been paid so far to the potential of distributed punishment. In this work we put forward the hypothesis that distributed punishment is more effective than individual punishment, because is more persuasive in conveying messages of peer condemnation and of shared norm defense. We test this hypothesis by presenting results from laboratory experiments with humans and virtual agents and from simulations solely with agents.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Incentives for Cooperation, Normative systems, Social simulation, Modeling cognition and socio-cultural behavior

1. INTRODUCTION

Theoretical and laboratory studies indicate that cooperation and the maintenance of social order typically requires a punishment threat, as the temptation to cheat, free-ride and violate norms is always strong for autonomous agents [4, 5].

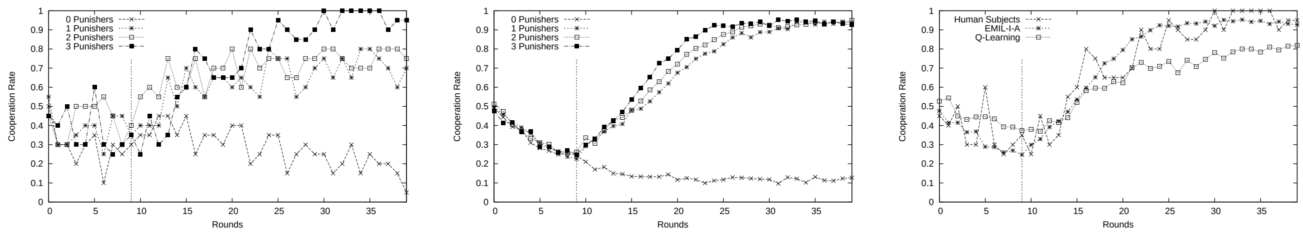
With few exceptions [3], punishment has been usually modeled as (a) a *material* damage, i.e. a cost inflicted to the target, by (b) a *single* agent, that (c) sustains *alone* all the costs of the punishing action (including those consequent to possible retaliations) [2]. On the contrary, ethnographic evidence shows that punishment is often *distributed*, i.e. performed by many, which share the costs of acting, and includes gossip and other forms of explicit or implicit communication. In this work, we focus on the potential of distributed punishment in promoting compliant conduct. With distributed punishment we refer to the practice that occurs when a number n of agents, where $n > 1$, inflicts the target a material damage, such that each punisher sustains a share of the punishment cost. In particular, we suggest that when distributed, punishment works as a *norm-signalling* tool and we put forward the hypothesis that distributed punishment may boost cooperation more than individual one because it is more effective in expressing cooperation norms, as it is more likely to be interpreted as a sanction (for an analysis of the differences between punishment and sanction [6]).

We present cross-methodological evidence supporting our hypothesis: a laboratory experiment with human subjects where we compare the respective effects of individual versus distributed punishment; and an agent-based simulation that allowed us to properly explore the power of "moral suasion" of distributed versus individual punishment

To test the viability of distributed punishment in achieving and maintaining cooperation, we conducted a laboratory experiment reproducing a social dilemma situation. In particular, participants (divided in groups of 4) played a *public goods game* in which they had to decide whether to invest or not their private endowment in a group fund. Payoffs are such that it is individually rational to abstain from investing in the group fund, yet the pro-social group best strategy would be investing in the group fund because this yields a bonus. After having decided whether to contribute or not to the group fund, participants have the possibility to punish. What is special to our set-up is that each group of 4 was composed of one human subject and three confederate virtual players. Human subjects were not informed of the fact that they were playing with confederate virtual players. The reason for putting each human subject in a group with three confederate virtual players is to be able to observe humans in a completely controlled situation.

The experiment consists of four treatments, which differ with respect to the number of the punishing subjects: (1) no punishment, (2) the subject is punished by *one* peer, (3) the subject is punished by *two* peers, (4) the subject is punished by *three* peers. The material damage imposed

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.



(a) Human Cooperation Rates (b) EMIL-I-A Cooperation Rates (c) Behavioural Comparison

Figure 1: Laboratory and Simulation Results.

on the punished agent in treatments 2, 3 and 4 is *identical* (i.e. it reduces the payoffs of the punished subject to zero) and the way the experiment has been implemented prevents the occurrence of reputational effects, as participants cannot identify one another. Thus, the material and symbolic incentives imposed in treatments 2, 3 and 4 are the same.

In Figure 1(a), the average cooperation rates obtained in the four treatments are shown. Only the behavior of human subjects is plotted. After run 10 the four treatments are actived and it is possible to observe their relative effects on the cooperation level. In the no punishment condition, the cooperation level rapidly collapses. On the contrary, in the three punishing treatments the cooperation level increases with respect to the first 10 rounds and is higher than the one obtained in the no punishment treatment. It is interesting to notice that being punished by three group members (i.e. 3 punishers treatment) leads to a higher cooperation level than when they are punished by two or just one subjects.

As the *same* material damage is imposed in all the treatments (except for the *0 Punishment Treatment* one), we hypothesize that the explanation for the difference on the cooperation rates has to be found in additional information that the punished players receive. We suggest that the higher the number of punishers, the less likely the observers will interpret their behaviors as dictated by the self-interest and, conversely, the more likely they will attribute the punishment to impersonal, possibly normative and legitimate reasons. In other words, our hypothesis is that distributed punishment is more likely to be interpreted as a norm-defending act than individual punishment, thus conveying a strong normative message of peer condemnation.

To test this hypothesis, we designed cognitively complex agents able to interpret as normative the social information they are exposed to and to include it into their decision-making. The agent architecture used for such task is EMIL-I-A [1, 6]. We then replicated the experiment conducted in the laboratory through agent-based simulation.

It is interesting to notice that the cooperation dynamics achieved in the simulation experiment with EMIL-I-As (see Figure 1(b)) are very similar to the ones obtained in the experiment with human subjects (see Figure 1(a)). However, the difference in the cooperation levels observed in the three punishment treatments in the laboratory experiment (with a higher level of cooperation when 3 punishers acted simultaneously) is stronger than the one achieved with EMIL-I-A agents). A possible explanation for this difference is that humans in addition to be sensitive to the fact that three punishers acted together, are also influenced by the fact that is the group as a whole that reacts against his conduct. This additional information is not taken into account by EMIL-I-As. Finally, we conducted a simulation experiment in which the game is played by *Reinforcement Learning* agents, not endowed with normative reasoning and driven only by utilitarian motivations. In Figure 1(c) the cooperation levels obtained in the 3 punishers treatment by human subjects, normative agents and Reinforcement Learning agents are confronted. Data show that Reinforcement Learning agents obtain cooperation levels similar to humans, confirming that the utilitarian motivation in humans is very strong, although the cooperation rates are not as high as the ones obtained by humans and EMIL-I-As.

In this study, we have provided some experimental evidence to show the viability of distributed punishment in promoting cooperation. Distributed punishment is shown to be a powerful tool through which messages of peer condemnation and of shared norm defense are conveyed. These data provide support for the hypothesis that punishment is effective in regulating people's behavior not only through the imposition of a material damage, but also thanks to the normative information it conveys and the normative requests it asks people.

Acknowledgments

This work was supported by the Spanish Education and Science Ministry [EVE project, TIN2009-14702-C02-01]; the Generalitat de Catalunya [2009-SGR-1434]; the Complexity-Net Project SEMIRA.

2. REFERENCES

[1] G. Andrighetto and D. Villatoro. Beyond the carrot and stick approach to enforcement: An agent-based model. In *European Perspectives on Cognitive Science*. New Bulgarian University Press, 2011.

[2] R. Boyd, H. Gintis, and S. Bowles. Coordinated punishment of defectors sustains cooperation and can proliferate when rare. *Science*, 328(5978):617–620, 2010.

[3] R. Ellickson. *Order without Law : How Neighbors Settle Disputes*. Harvard University Press, June 2005.

[4] E. Fehr and S. Gachter. Altruistic punishment in humans. *Nature*, 415:137–140, 2002.

[5] E. Ostrom, J. Walker, and R. Gardner. Covenants with and without a sword: Self-governance is possible. *Am. Pol. Sci. Review*, 86(2):404–417, 1992.

[6] D. Villatoro, G. Andrighetto, R. Conte, and J. Sabater-Mir. Dynamic sanctioning for robust and cost-efficient norm compliance. In *Proceedings of the 22nd IJCAI*, 2011.

The "Resource" Approach to Emotion

(Extended Abstract)

Sabrina Campano, Nicolas Sabouret, Etienne de Sevin, Vincent Corruble
Laboratoire d'Informatique de Paris 6
4, place Jussieu
75005 Paris, France
name.surname@lip6.fr

ABSTRACT

In this paper, we present a model for the simulation of affective behaviour without emotion categories, centered around the *theory of conservation of resources* [3]. Each agent can acquire or protect resources, and behaviour choice depends on resources state, as well as agent's needs and preferences. We also present a first evaluation.

Categories and Subject Descriptors

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

Keywords

virtual agent, affective behaviour, believability

1. INTRODUCTION

Emotions have been at the core of many psychological studies for several decades. This topic gave rise to computational models of emotion, either aiming at the simulation of lifelike agents, or at the study of psychological processes. One remaining important issue is the influence of emotions on behaviour. Most computational models rely on emotion variables that must be manually parametrized so as to outline believable affective responses and behaviours. However, in the general case, finding the correct number of parameters, their value, and the influence of each one on the general model is a difficult matter.

In the computational model *Affective Reasoner* [2], several actions, like the somatic responses *flush* or *tremble*, are linked with one emotion label. Actually, the association between emotion and various behaviours can't be done easily. Authors of the OCC model [4] notice that "*the same behavior can result from very different emotions*" and "*very different behaviors can result from the same emotion*".

In 1994, R. Pfeifer published an article entitled "*The 'Fungus Eater Approach' to Emotion*" [5], in which he proposes to view emotion as an emergent phenomenon, that does not need to be engineered in a computational model. Actually, from a psychological point of view, emotions can be considered as interpretations of perceptions [1] instead of being entities acting on behaviours. Following Pfeifer's approach, we claim that it is possible to design an architecture capable of producing *emotional* behaviours (*i.e.* behaviours that can be described with emotion terms by a human observer) without using emotion variables, parameters, dimensions or categories in the model itself. Pfeifer's approach was applied to an environment and agents of "*extreme simplicity*" (*sic*), and was not validated by an evaluation protocol.

In order to apply this approach to virtual agents, we propose to design an architecture capable of handling various behaviours, from primary ones to social ones. Our hypothesis is that the theory of "Conservation of Resources" (COR) by psychologist S.E. Hobfoll [3] offers an interesting lead to this purpose. The main principle of this theory is that individuals strive to protect their resources, and to acquire new ones. The concept of resource refers to many types : social ones such as self-esteem or caring for others, material ones such as a car, or physiological ones such as energy. Hence, we propose an architecture based on this theory, that had not been computationally formalized nor implemented so far.

2. PROPOSED MODEL

2.1 General Overwiew

Our model is centered around the dynamics of acquisition and protection of resources. It is based on the following principles : (a) when an acquired resource is threatened, an agent tries to protect it; (b) when no acquired resource is threatened, an agent tries to acquire resources that it desires.

Resources in the environment are associated with protective and acquisitive behaviours, that agents can realize in order to defend a threatened resource, or to acquire a new one. An agent can only perform one behaviour at the same time. The nature of protective and acquisitive behaviours depends on the resource type. For example "to talk" lets one acquire a resource of the "Social Interaction" type, and "to eat" lets one acquire a resource of the "Energy" type. Each agent has needs for resource types, and these needs define the resources desired by the agent. An overview of resource sets and corresponding behaviours is shown on figure 1.

Each behaviour can have both positive effects (acquisition or protection) and negative effects (threatening or loss) over resources. An agent i passing an agent j in a waiting line to acquire a rank threatens the current rank of j. In turn, agent j can engage in a protective behaviour in order to defend its rank, which may result in a resource loss for i.

Each agent has individual preferences over resources which determine the value of a resource from agent's point of view.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

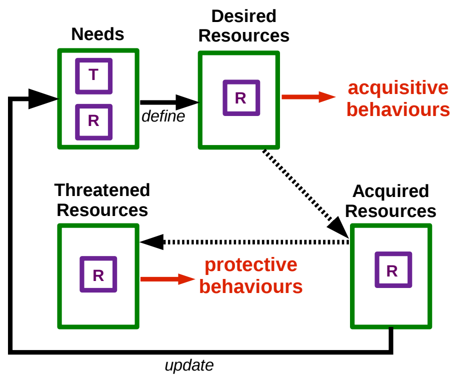


Figure 1: Resource Sets and Behaviours

This reflects in our model the personality of an agent, and to some extent its social role. As an example, for an agent i which is a politician, the "Reputation" type may be preferred to many other resource types, whereas for an agent j which is fond of pop music, the "Pop Music Concert" may be more important. This implies that j can risk to loose its reputation in passing someone in a waiting line for a pop music concert, whereas i will not take this risk. A payoff value is computed automatically for each behaviour according to these preferences and to the behaviour's effects. The behaviour selection is made according to behaviours payoff value, with protective behaviours having precedence over acquisitive ones.

Example : in the context of a waiting line, we define the set of acquisitive behaviours for a resource of "Rank" type $B^+_{Rank} = \{pass(i,j)\}$, which contains the behaviour of an agent i passing an agent j, and the set of protective behaviours $B^o_{Rank} = \{protest(i,j)\}$, which contains the behaviour of i protesting against j. For 2 given agents i and j we define $Reputation \succeq_i Rank$, which means that i prefers the "Reputation" type to the "Rank" type, and $Rank \succeq_j Reputation$. Agent i has an acquired resource that is the second rank in the waiting line and a reputation resource, and agent j has the third position. An effect of passing an agent in a waiting line ($pass(i,j)$) is to loose its reputation. Hence, agent i will not realize this behaviour, since "Reputation" is more important for it than "Rank". On the contrary, agent j can realize this behaviour because "Rank" is less important for it than "Reputation". When the "Rank" of an agent i is threatened by an agent j, i can realize the behaviour $protest(i,j)$.

3. EVALUATION

The evaluation conducted aimed at assessing if human observers can interpret emotions from agents' behaviours exhibited by our implemented model, and if they consider these behaviours as believable. It was our main hypothesis. Our protocol relies on written subjective reports made by observers watching a simulation video clip. Each participant had to respond to a questionnaire about a video clip submitted on Internet. There were 3 video clips from a scenario involving a fire (scenario 1), and 4 video clips from a scenario involving a waiting line (scenario 2). For scenario 1 two characters, an adult and a baby, were in a kitchen. A fire started in the room, and the adult could realize the behaviours : save the baby, save the bird, and save the hamburger. For scenario 2 some characters were in a waiting line, and they had the possibility to wait, to pass other agents, or to protest against intruders.

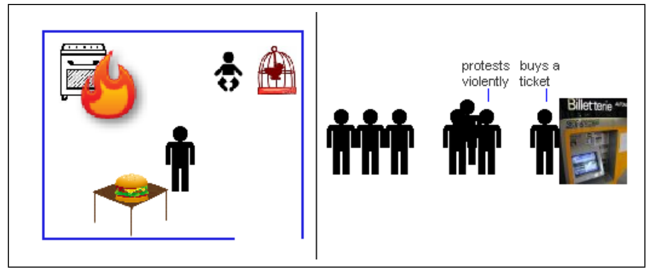


Figure 2: Left : fire scenario - Right : waiting line scenario.

According to our results, our main hypothesis was validated. Participants cited numerous emotion labels when they were explicitely invited to. They also used emotion labels when they were asked to describe and explain agents' behaviours at the beginning of the questionnaire. However, the percentage of participants who used emotion labels in the description remains below 50% per video clip. Participants also rated video clips in accordance with our main hypothesis in terms of believability and emotion interpretation. There were exceptions for two video clips : one was conceived in order to be not realistic, but participants rated it as realistic, and another one was rated as not realistic, which was not expected. An explanation could be that we underestimated the believability of agents' behaviours, and that the threat over some resources was not well represented in our simulation display.

4. CONCLUSION AND PERSPECTIVES

We presented an architecture aimed at providing virtual agents with believable emotional behaviours, which does not manipulate emotion categories. Our main hypothesis was that the simulation of such behaviours does not necessarily require an architecture grounded on emotion categories. Our results, based on the simulation of two different scenarii, validated this hypothesis. Therefore, we can rely on the model presented in this paper for future work on the simulation of affective behaviours. In particular, we plan to work on the hierarchy between resource types in agents' preferences, and to establish a general set of resources which could be used in every scenario.

5. REFERENCES

[1] L. Barrett. Solving the emotion paradox: Categorization and the experience of emotion. *Pers. and social psychol. review*, 10(1):20, 2006.

[2] C. Elliott. The Affective Reasoner: A process model of emotions in a multi-agent system. 1992.

[3] S. Hobfoll. Conservation of resources. *American Psychologist*, 44(3):513–524, 1989.

[4] A. Ortony, G. L. Clore, and A. Collins. *The cognitive structure of emotions*. New York : Cambridge University Press, 1988.

[5] R. Pfeifer. The"Fungus Eater"Approach to Emotion: A View from Artificial Intelligence. *Cognitive Studies*, 1:42–57, 1994.

Emotional Contagion with Virtual Characters (Extended Abstract)

Jason Tsai[1], Emma Bowring[2], Stacy Marsella[3], Milind Tambe[1]

[1]University of Southern California, Los Angeles, CA 90089
{jasontts, tambe} @usc.edu
[2]University of the Pacific, Stockton, CA 95211
ebowring@pacific.edu
[3]USC Institute for Creative Technologies, Playa Vista, CA 90094
marsella@ict.usc.edu

ABSTRACT

In social psychology, emotional contagion describes the widely observed phenomenon of one person's emotions mimicking surrounding people's emotions [8]. While it has been observed in human-human interactions, no known studies have examined its existence in agent-human interactions. As virtual characters make their way into high-risk, high-impact applications such as psychotherapy and military training with increasing frequency, the emotional impact of the agents' expressions must be accurately understood to avoid undesirable repercussions.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence–*Intelligent agents*

General Terms

Human Factors

Keywords

Virtual Agents, Emotional Contagion, Social Influence

1. INTRODUCTION

Emotional contagion is defined as the tendency to catch the emotions of other people [8]. While initial work focused on documenting its existence, recent research has moved to understanding its impacts on everyday life. In the workplace, researchers have examined its influence on promoting employee efficiency and client happiness [12]. Research in administrative sciences has shown emotional contagion to improve cooperation, decrease conflict, and increase perceived task performance in groups and organizations [1]. Small et al. have shown substantial impacts on charitable donation amounts with only a still image [15]. Though its effects are often felt, in-depth understanding of emotional contagion remains an open area of research.

A variety of hypotheses regarding factors that influence emotional contagion have been explored in social psychology. A popular one examines differences in the strength of emotional contagion felt by men and women, with many researchers finding that women are significantly more responsive to emotional contagion than men [4, 16]. Researchers have also found that contagion increases in cases where the subject shares the same ethnicity as the stimulus [4] and when the expression is stronger [18]. Finally, attraction to the stimulus has been shown to have a positive effect on the contagion experienced by subjects [16].

The vast majority of emotional contagion research, however, has come from the social sciences and examines the spread of emotions from humans to other humans. Emotional contagion's impact in virtual agents' interactions with humans, however, is a largely untouched area of research. Specifically, while many researchers have worked to understand immersion, rapport, and influence in other contexts [7, 9], far fewer have looked into the emotional impact that the mere presence of virtual character emotions can have on people. The effects are assumed to either be nonexistent and therefore overlooked entirely or to mimic human-human emotional influences. However, as this work demonstrates, these are both poor assumptions to make and can be harmful to users in sensitive domains. As virtual agents enter high-risk and emotionally delicate applications such as virtual psychotherapy [13, 14], for example, researchers must be cognizant of all potential emotional influences characters can have on users.

Attempting to confirm the aforementioned social psychology findings in agent-human emotional contagion forms the basis of this work. Pursuant of this goal, three sets of studies are conducted. The first study examines the pure contagion case by simply showing subjects a still image of a virtual character with either a happy expression or a neutral expression and then assessing the subject's mood thereafter. The use of a still image as a manipulation follows from previous studies in emotional contagion [15, 18].

The second study adds the presentation of a game-theoretic situation known as a Stag Hunt along with the character image to assess both the contagion the behavioral impact of the virtual character in a strategic setting. While studies have shown that emotional contagion can impact one's propensity to trust and enhance perceived cooperation among other findings [1, 5], there has been far less work showing behavioral impacts in strategic situations. Although people may report themselves to be more trusting, for example, this may not result in any meaningful impact on behavior in a strategic situation. Thus, we also attempt to examine whether behavioral impacts arise in strategic situations from agent-human contagion to better understand its potential impacts in real-world agent applications. Finally, the third study examines the post-hoc hypothesis

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

that the presentation of a decision to the user dampens the emotional contagion effect. Specifically, we present the same strategic situation as in the second study, but with the decision already made for the subject. These studies present the first attempt to assess emotional contagion from virtual characters to human users.

2. BACKGROUND & RELATED WORK

Emotional contagion research in the agents literature falls primarily into three categories: models of emotional contagion, creating rapport between virtual agents and humans, and the impact of agent mood expressions on behavior. Models of emotional contagion have been explored in a computational context that focus on crowd or society simulation. For example, [2, 6, 11] each present alternative models of emotional contagion in agent crowds, while [17] proposes a comparison technique to evaluate such models. This body of work is an attempt to mimic human-human contagion and not an exploration of agent-human contagion which we seek to understand here.

There also exists a large body of work on the interaction between virtual agents and humans [3, 7]. The entire area of virtual rapport, for example, focuses on user opinions of the virtual agents and their interaction. The primary goal is to create agents that users enjoy, appreciate, and relate to. Recent work has looked at the impact of agent expressions in a strategic negotiation setting [3] as well. However, their work focuses on the behavioral impact of varying the intent of agent expressions on user behavior without examining the emotional impact or the mechanism by which the change is induced. Neither of these works explicitly examine the impact of virtual character expressions on the emotions of subjects.

In the social sciences, the literature on emotional contagion is far more expansive. Hatfield et al. [8] popularized the area by compiling a plethora of situations in which the phenomenon had been observed in their work as well as the work of other researchers. Follow-up research by the co-authors as well as researchers in related fields such as managerial and occupational sciences [1, 12, 15] continued to detail the effects of the phenomenon in new domains. Recently, there have been works beginning to quantify emotional contagion and explore cross-cultural variations in attributes that affect emotional contagion [10].

In light of the extensive evidence of emotional contagion's effects in human-human interactions, our work extends the understanding of this phenomenon into the realm of agent-human interactions. While some studies have been conducted with live people as the stimulus, a large body of social psychological studies of emotional contagion features an image or video of only a person's face as the origin of the contagion [15, 18]. With the rapid improvements in virtual agent facial displays, and the accepted assumption that the facial display of emotion plays a key role in emotional contagion, we would expect to see a contagion of emotions from an image of a virtual agent's face to humans. The intricacies of this contagion and its differences with human-human contagion are the subject of this work.

3. ACKNOWLEDGEMENT

This research is supported by the United States Department of Homeland Security through Center for Risk and Economic Analysis of Terrorism Events (CREATE). This research was sponsored in part by the U.S. Army Research, Development, and Engineering Command (RDECOM) Simulation Training and Technology Center (STTC). The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

4. REFERENCES

[1] S. G. Barsade. The Ripple Effect: Emotional Contagion and Its Influence on Group Behavior. *Administrative Science Quarterly*, 47:644–675, 2002.

[2] T. Bosse, R. Duell, Z. A. Memon, J. Treur, and C. N. V. D. Wal. A Multi-Agent Model for Emotion Contagion Spirals Integrated within a Supporting Ambient Agent Model. In *PRIMA-09*, 2009.

[3] C. de Melo, P. Carnevale, and J. Gratch. The Effect of Expression of Anger and Happiness in Computer Agents on Negotiations with Humans. In *AAMAS-11*, 2011.

[4] W. Doherty, L. Orimoto, T. M. Singelis, E. Hatfield, and J. Hebb. Emotional Contagion: Gender and Occupational Differences. *Psychology of Women Quarterly*, 19(3):355–371, 1995.

[5] J. R. Dunn and M. E. Schweitzer. Feeling and Believing: The Influence of Emotion on Trust. *Psychology of Personality and Social Psychology*, 88(5):736–748, 2005.

[6] F. Durupinar. *From Audiences to Mobs: Crowd Simulation with Psychological Factors*. PhD thesis, Bilkent University, 2010.

[7] J. Gratch, A. Okhmatovskaia, F. Lamothe, S. Marsella, M. Morales, R. J. van der Werf, and L.-P. Morency. Virtual Rapport. In *IVA-06*, 2006.

[8] E. Hatfield, J. T. Cacioppo, and R. L. Rapson. *Emotional Contagion*. Cambridge University Press, 1994.

[9] P. Kulms, N. C. Krämer, J. Gratch, and S.-H. Kang. It's in Their Eyes: A Study on Female and Male Virtual Humans' Gaze. In *IVA-11*, 2011.

[10] L.-O. Lundqvist. Factor Structure of the Greek Version of the Emotional Contagion Scale and its Measurement Invariance Across Gender and Cultural Groups. *Journal of Individual Differences*, 29(3):121–129, 2008.

[11] G. Pereira, J. Dimas, R. Prada, P. A. Santos, and A. Paiva. A Generic Emotional Contagion Computational Model. In *ACII-11*, 2011.

[12] S. D. Pugh. Service with a smile: Emotional contagion in the service encounter. *Academy of Management Journal*, 44(5):1018–1027, 2001.

[13] A. Rizzo, J. Pair, P. J. McNerney, E. Eastlund, B. Manson, J. Gratch, R. W. Hill, and W. Swartout. *Development of a VR Therapy Application for Iraq War Military Personnel with PTSD*, volume 111 of *Medicine Meets Virtual Reality*, pages 407–413. IOS Press, 13th Annual Medicine Meets Virtual Reality Conference, Long Beach, CA, 2005.

[14] B. O. Rothbaum, L. F. Hodges, D. Ready, K. Graap, and R. D. Alarcon. Virtual reality exposure therapy for vietnam veterans with posttraumatic stress disorder. *Journal of Clinical Psychiatry*, 63(8):617–622, August 2001.

[15] D. A. Small and N. M. Verrochi. The Face of Need: Facial Emotion Expression on Charity Advertisements. *Journal of Marketing Research*, 46(6):777–787, December 2009.

[16] N. Stockert. *Perceived Similarity and Emotional Contagion*. PhD dissertation, University of Hawaii at Manoa, Dept. of Psychology, 1994.

[17] J. Tsai, E. Bowring, S. Marsella, and M. Tambe. Empirical Evaluation of Computational Emotional Contagion models. In *IVA-11*, 2011.

[18] B. Wild, M. Erb, and M. Bartels. Are emotions contagious? evoked emotions while viewing emotionally expressive faces: quality, quantity, time course and gender differences. *Psychiatry Research*, 102(2):109–124, 2001.

Higher-order social cognition in rock-paper-scissors: A simulation study (Extended Abstract)

Harmen de Weerd, Rineke Verbrugge, Bart Verheij
Department of Artificial Intelligence, University of Groningen
{hdeweerd,rineke,b.verheij}@ai.rug.nl

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms
Algorithms

Keywords
iterated reasoning, repeated games, theory of mind, opponent modeling, agent-based simulation

1. INTRODUCTION

In settings where multiple strategic agents perform actions that influence each other's decision-making process, it is often necessary to accurately predict the behaviour of others in order to respond appropriately [9]. One option to do so is by modeling an opponent explicitly, e.g. through dynamic epistemic logic [13], Interactive POMDPs [6], multi-agent influence diagrams [7], or iterated best-response models such as cognitive hierarchy models [2] and level-n theory [1]. These models allow for recursive modeling of an opponent, by modeling the opponent as an opponent-modeling agent itself, creating increasingly complicated models to predict the actions of increasingly sophisticated opponents.

In humans, the ability to predict the actions of others by explicitly attributing to them unobservable mental content, such as beliefs, desires, and intentions, is known as *theory of mind* [10] or *social cognition*. Experiments in which humans play games show evidence that humans use theory of mind recursively in their decision-making process [8]. For example, when asked to search for a hidden object in one of four boxes, three of which are labeled 'A' and one of them 'B', participants tend to ignore the box labeled 'B', using their nested belief that a hider would believe that a seeker would consider the most obvious place to search for a hidden object to be the box labeled 'B' [4]. Whether any non-human species makes use of theory of mind is a controversial matter [3, 11], and although recursive opponent modeling could continue indefinitely, humans only use higher-order theory of mind (i.e. recursive theory of mind) up to a certain point [14]. In an evolutionary sense, the costs of using higher orders of theory of mind may outweigh the benefits.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

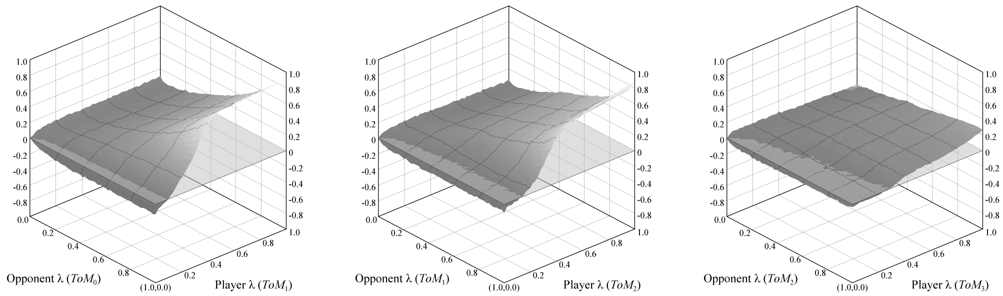
Agent-based modeling has proven to be a useful research tool to investigate how behavioural patterns emerge from interactions between individuals. We have used this approach to investigate the effectiveness of recursive opponent modeling in the setting of a specific game called Limited Bidding [5]. The results in this setting suggest that there may be a limit to the advantage that can be obtained through recursive opponent modeling, but these limitations may also be caused by the specific game structure. Here, we apply the model presented in [5] to rock-paper-scissors (RPS), perhaps the most transparent non-trivial setting in which the role of theory of mind can be investigated. In addition to the standard RPS game, in which 'rock' defeats 'scissors', 'scissors' defeats 'paper', and 'paper' defeats 'rock', we also investigate two variations. Elemental RPS (ERPS) preserves the structure of RPS with a unique best response to each action, but confronts agents with a choice out of five available actions. In rock-paper-scissors-lizard-Spock (RPSLS), agents choose from five actions as well, but this setting differs from ERPS in that each action is defeated by two other actions (e.g. both 'lizard' and 'scissors' defeat 'paper').

2. SIMULATION THEORY OF MIND

In our approach, an agent tries to take advantage of regularities in his opponent's strategy by predicting her behaviour through simulation-theory of mind [9]. An agent takes the perspective of the opponent, and simulates his opponent's decision-making process by making the decision himself. Through the implicit assumption that the opponent's thought process can be accurately modeled by his own, the agent predicts that his opponent will make the same decision he would have made if the roles were reversed.

A zeroth-order theory of mind agent models patterns in his opponent's behaviour, but does not attribute any mental content to her. In contrast, a first-order theory of mind agent considers the possibility that his opponent is trying to win the game for herself, and that she reacts to the choices he makes. For example, suppose that the first-order theory of mind agent remembers that he previously played 'rock' against the opponent he is facing. He realizes that if the roles were reversed, and he would remember her to have played 'rock' before, the agent would play 'paper' more often. The first-order theory of mind agent has the ability to attribute this thought process to his opponent, and predict that she will play 'paper' more often. Given this prediction, the agent reasons that he should play 'scissors' more often.

A second-order theory of mind agent also models his opponent as a first-order theory of mind agent. He believes



(a) First-order (b) Second-order (c) Third-order

Figure 1: Average performance of theory of mind agents playing rock-paper-scissors against an opponent of a lower order.

that his opponent may be putting herself in his position. If the second-order theory of mind agent remembers his opponent to have played 'rock' in previous encounters, he would therefore believe her to predict that he will be playing 'paper' more often. As a result, the second-order theory of mind agent would predict his opponent to play 'scissors' more often, in which case he should play 'rock' more often himself.

3. RESULTS

Using the mathematical model from [5], we determined the performance of the theory of mind agents described in the previous section by placing them in competition. Figure 1 shows the results for the RPS game as a function of the learning speed of both the agent and his opponent. We find that first-order and second-order theory of mind agents clearly outperform opponents that are more limited in their ability to model others. A third-order theory of mind agent mostly outperforms a second-order theory of mind opponent as well, but only marginally.

Results of similar competitions in the ERPS variation suggest that these diminishing returns on higher orders of theory of mind found in RPS are not related to the number of actions available to the agents. When agents choose from five instead of three possible actions, performance of third-order theory of mind agents only improves when playing against opponents that are unable to make use of theory of mind, and play according to a stationary mixed strategy.

Compared to the results of RPS, performance of theory of mind agents in RPSLS is greatly reduced. This suggests that the effectiveness of theory of mind is dependent on the existence of a unique best response. One explanation for this low performance is that when an agent is indifferent between two actions, he chooses either one with equal probability. A slight asymmetry, such that one option is preferable over the other, may therefore benefit agents making use of higher-order theory of mind. Such asymmetries may create a focal point [12] for agents with a lower order of theory of mind, which may result in more predictable behaviour.

4. CONCLUSION

Our results in the RPS game are qualitatively equivalent to those obtained in the setting of Limited Bidding [5]. This shows that the results reported in [5] are not exclusive to the specific game setting studied there, but are generalizable to games of different designs. This provides further support for the hypothesis that first-order and second-order theory of mind provide a clear advantage over opponents of a lower order, while deeper levels of recursion help less [5].

5. ACKNOWLEDGMENTS

This work was supported by the Netherlands Organisation for Scientific Research (NWO) Vici grant NWO 277-80-001.

6. REFERENCES

[1] M. Bacharach and D. Stahl. Variable-frame level-n theory. *Game Econ. Behav.*, 32(2):220–246, 2000.

[2] C. Camerer, T. Ho, and J. Chong. A cognitive hierarchy model of games. *Q. J. Econ.*, 119(3):861–898, 2004.

[3] P. Carruthers. Meta-cognition in animals: A skeptical look. *Mind Lang.*, 23(1):58–89, 2008.

[4] V. Crawford and N. Iriberri. Fatal attraction: Salience, naïveté, and sophistication in experimental "Hide-and-Seek" games. *Am. Econ. Rev.*, 97(5):1731–1750, 2007.

[5] H. de Weerd and B. Verheij. The advantage of higher-order theory of mind in the game of limited bidding. In J. van Eijck and R. Verbrugge, editors, *Proc. Workshop Reason. About Other Minds*, CEUR Workshop Proceedings 751, pages 149–164, 2011.

[6] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *J. Artif. Intell. Res.*, 24(1):49–79, 2005.

[7] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Game Econ. Behav.*, 45(1):181–221, 2003.

[8] B. Meijering, H. van Rijn, N. Taatgen, and R. Verbrugge. I do know what you think I think: Second-order theory of mind in strategic games is not that difficult. In *Proc. 33rd Conf. Cogn. Sci. Soc.*, pages 2486–2491. Cognitive Science Society, 2011.

[9] S. Nichols and S. Stich. *Mindreading.* Oxford University Press, USA, 2003.

[10] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behav. Brain Sci.*, 1(4):515–526, 1978.

[11] M. Schmelz, J. Call, and M. Tomasello. Chimpanzees know that others make inferences. *PNAS*, 108(7):3077–3079, 2011.

[12] R. Sugden. A theory of focal points. *Econ. J.*, 105(430):533–550, 1995.

[13] W. van Ditmarsch, H. van der Hoek and B. Kooi. *Dynamic Epistemic Logic.* Springer, 2007.

[14] R. Verbrugge. Logic and social cognition: The facts matter, and so do computational models. *J. Philos. Logic*, 38:649–680, 2009.

Can I trust you? Sharing information with artificial companions

(Extended Abstract)

Matthias U. Keysermann
MACS, Heriot-Watt University
Edinburgh, UK
muk7@hw.ac.uk

Ruth Aylett
MACS, Heriot-Watt University
Edinburgh, UK
r.s.aylett@hw.ac.uk

Sibylle Enz
Otto-Friedrich Universität
Bamberg, Germany
sibylle.enz@uni-bamberg.de

Henriette Cramer
Mobile Life @ SICS
Kista, Sweden

Carsten Zoll
Otto-Friedrich Universität
Bamberg, Germany

Patricia A. Vargas
MACS, Heriot-Watt University
Edinburgh, UK

ABSTRACT

This paper discusses an experiment to investigate issues of trust and confidentiality when sharing information with a robot companion in an office context.

An online questionnaire was used to collect opinions about information sharing with a robot companion and preferences for collection and treatment of information. In a subsequent live interaction study, subjects role-played new members of an office team exchanging potentially sensitive information with the robot companion. Evaluated results and their implications are summarised and we suggest generic improvements for HRI systems used for information exchange.

Categories and Subject Descriptors

H.5.m [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Experimentation, Human Factors

Keywords

Human-Robot Interaction, Trust, Information Sharing, Privacy

1. INTRODUCTION

Trust has long been a significant topic in software agent research [6, 2], relating to topics such as reliability, transparency and provenance in information exchange. However, in the context of embodied agents such as robots, more social reactions come into play [1, 3, 7].

Autonomous agents need information to successfully deliver their services. Such autonomous behaviour, however, is in direct contrast with principles such as user control, privacy and transparency [5], raising the issue of user trust.

This paper relates to a study of trust in the context of long-term robot companions being developed in the LIREC project[1]. This is work in human-robot interaction [4] in which robots are no longer merely machines for achieving tasks but become social actors in real-world human environments.

If a companion is not purely a personal one and interacts with more than one user, then it may hold information relating to one user that ought not to be relayed to another. For these reasons we carried out an experiment looking at issues of trust and privacy when sharing information with a robot companion. To gather general opinions on this topic we first conducted an online questionnaire study; this was then followed by a practical study in which subjects interacted directly with a robot companion.

2. STUDY 1: QUESTIONNAIRE

The questionnaire was designed to address the following research questions:

1. In which situations is the companion considered helpful?
2. Which kinds of personal/team-related information are people willing to share with the companion?
3. What are people's preferences regarding: the collection of information through the companion; the disclosure of information through the companion; interaction modalities with/control of the companion?

The questionnaire provided the participants with a selection of options regarding various types of information to select from and offered them the opportunity to provide additional input via free text responses.

The most important findings were that the companion is considered particularly helpful when working on a joint task, during absences from team, and when working in separate rooms in the same building. According to these functional preferences, information that people are particularly

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1]http://lirec.eu

willing to share with it concerns meeting dates, important tasks and deadlines, as well as absences from team. Information collected should be reported when collected or only collected when indicated. Before accessing other sources of information (like social networks or other personal internet resources), the companion should ask for permission. As far as disclosure of work-related information is concerned, it should only be disclosed either to classified persons or to team members, but not to others, and it should only be given after authorisation. Personal and private information should never be given away. Information secret to the requesting person should not be given; the companion should indicate that it has no authorisation to give it.

It is very important to control the memory content and treatment of information; however, interactions to exert control should only occur at medium levels of frequency. In order to minimise these interactions over time, sharing preferences should be chosen more and more autonomously by the companion based on previous choices taken by the user. When requesting information, the option to only receive a summarised output of new and currently relevant information can enhance comfort as it provides a quick and effective way of keeping users informed.

3. STUDY 2: LIVE INTERACTION

The second part of the experiment involved live interaction with an actual companion in order to examine how far the issues raised in the questionnaire were translated into interactions within a specific scenario and with a real robot.

The participants were asked to imagine they were a new team member in a team of researchers working on different projects. The goal for the participants was to get to know as much as possible about the other team members and current projects so as to familiarise themselves with their new coworkers and workplace. In the scenario, none of their team members were in the office so that the companion was the only source of information. Participants were able to ask the companion for different kinds of information about the team, and – where considered appropriate – give information about their own role.

A tablet computer was used for requesting information and entering personal data. It offered a simple interface developed in HTML; once logged in users navigated pages using touch buttons and typed on the touch keyboard. The companion responded verbally using a text-to-speech programme that outputs a human-like unit selection female voice.

It could be noticed that not much information was requested about other team members compared to the information the companion held in its memory. We attribute this, along with the limited information participants supplied about their role, to limited engagement of the subjects in the role-play. It was likely that few or none had ever been in the position of entering a new workplace team in real life. We rule out lack of confidence in the system as the reason because participants reported that they felt comfortable when providing information.

Their responses showed that participants gained in trust for the companion because they were informed about who their information could be disclosed to as well as being able to change these authorisation levels. This confirms that data transparency and control over data are very important and should never be neglected.

4. CONCLUSIONS

Control over the information collected is a priority for the great majority of our participants, confirming other research into the importance of transparency, control and user's trust [3]. Expectations, and effectively interactions with the agent, will teach the user which information will or will not be available. In our live interaction study we found the first interactions with the agent set the tone for the further information requests. While participants felt the companion did not give away information they did not expect it to give away, they did have assumptions not necessarily matching the realities of the agent's functionality. Transparency as well as actively finding out the preconceptions potential users hold and ways to counter misconceptions will be crucial.

Allowing for user control, for example implies asking before autonomously trying to collect information from other sources and clearly indicating which information is collected and why. However, participants also indicate they do not want to spend a lot of effort managing control mechanisms. Adaptation to their personal preferences without the need for explicit user input, and taking into account context (such as who else is present possibly 'overhearing' information and reasons why information is requested) would be useful. However, such adaptivity would also be in direct contrast with the control users desire. A clear approach to such issues has not been devised by the research community.

5. ACKNOWLEDGMENTS

This work was partially supported by the European Commission (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (Living with Robots and Interactive Companions). The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

6. REFERENCES

[1] T. W. Bickmore and J. Cassel. Relational agents: A model and implementation of building user trust. In *Proceedings of SIGCHI*, pages 396–403, 2001.

[2] C. Castelfranchi and R. Falcone. Trust and control: a dialectic link. *Applied Artificial Intelligence*, 14(8):799–823, 2000.

[3] H. Cramer, V. Evers, M. van Someren, and B. Wielinga. Awareness, training and trust in interaction with adaptive spam filters. In *CHI '09*, Boston, USA, 2009.

[4] K. Dautenhahn. Socially intelligent robots: Dimensions of human-robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704, 2007.

[5] A. Jameson and E. Schwarzkopf. Pros and cons of controllability. In *Proceedings of AH '02*, pages 193–202, 2002.

[6] J. Lee and K. See. Trust in automation: Designing for appropriate reliance. *Human Factor*, 46(1):50–80, 2004.

[7] B. Reeves and C. I. Nass. *The media equation: How people treat computers, television, and new media like real people and places*. University of Chicago Press, 1996.

MO-LOST: Adaptive ant trail untangling in multi-objective multi-colony robot foraging

(Extended Abstract)

Zhao Song
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
zhaos@sfu.ca

Seyed Abbas Sadat
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
sas21@sfu.ca

Richard T. Vaughan
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
vaughan@sfu.ca

ABSTRACT

In the context of large-population multi-objective robot foraging, we present a novel ant-inspired trail-following algorithm that is able to adaptively untangle multiple trails. The emergent result is often a set of short, non-intersecting trails that produce good system throughput due a good trade off between the dual goals of minimizing travel distance and spatial interference. Empirical simulation experiments with up to 200 robots suggest that the method can usefully improve performance in practice.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Experimentation

Keywords

Velocity obstacle, Multi-robot systems, obstacle avoidance

1. INTRODUCTION

We consider the classical problem of having multiple robots locate a source of resources and transport them to a sink location, repeating indefinitely. Robots use an instance of the general ant algorithm [1] to navigate between source and sink. The first demonstration of ant algorithms for this problem was by Iredi *et al*[2]. Iredi's abstract "ants" have no physical extent and so do not suffer from spatial interference, so the intersection of trails for different objectives was not problematic. Aiming for a practical system, we use a system of virtual pheromones implemented by wireless communication of waypoints in a shared localization space [4]. Sadat *et al* showed that maintaining spatially separated trails tends to keep robots apart and allows them to spend most of their time making progress towards their goals, thus increasing system throughput [3]. However, in multi-objective robot foraging problems we observe that SO-LOST often creates trails that are longer than necessary, and often has trails crossing at right-angles. Below we present a novel adaptive trail-following algorithm that is able to untangle many trails to create non-intersecting, thus low-interference, trails.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. MULTI–OBJECTIVE LOCALIZATION–SPACE TRAILS

We introduce a new ant algorithm called *Multi-Objective Localization-Space Trails* (MO-LOST) which extends and improves upon Sadat's SO-LOST and Vaughan's LOST methods. The details of LOST can be found in [4]. Spaced-Out LOST is a simple extension that modifies the robot's trail-following behaviour, and thus influences subsequent trail formation. In SO-LOST, when a robot is close to a waypoint that is *not* for its current goal, it will shift its velocity vector slightly left compared to the trajectory suggested by unmodified LOST (if local obstacles permit). Thus the fresh trail being laid by the robot will be slightly left of the previous trails. In the single-objective foraging problem examined by Sadat, there were two main trails after convergence; one from home to source and one returning. With the trails in opposite directions, the left shift repeatedly applied has the effect of spreading out the trails in space. SO-LOST is symmetric with respect to the currently assigned task, so there is no way to shift some trails preferentially to others. In multi-objective, multi-colony scenarios this is not always a good idea, as illustrated in Figure 1. Initially-intersecting trails (Fig. 1(top left)) should be de-intersected so that the total trail length is minimized (Fig. 1(bottom right) . MO-LOST extends SO-LOST so that when robots encounter trail intersections, only the longer trail is left-shifted. This asymmetry tends to leave shorter trails intact while wrapping longer trails around them. In large population sizes where trail intersection causes a lot of costly spatial interference, this non-intersecting short-trail configuration is optimal.

Intuitively, the robot moves toward the nearby waypoint labelled with the robot's current task that would take it nearer to its goal. If there is an waypoint labelled with another task nearby (i.e suggesting an interfering trail), that waypoint is tested to see if either (i) it is closer to its goal Place than the on-task waypoint or (ii) it belongs to a task which is cheaper on average than the robot's task. If either of these conditions hold, the robot shifts its driving left slightly to avoid the interfering waypoint in future. This action is shown schematically in Figure 1.

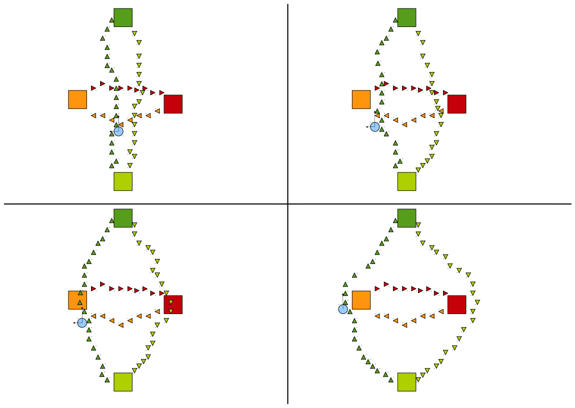


Figure 1: MO-LOST in action. On encountering a waypoint from a shorter trail, robots shift their trajectory slightly to the left. Over time the longer trail wraps around the shorter trail, avoiding intersection and thus reducing spatial interference between robots, while preserving short trails.

3. EXPERIMENTAL RESULTS

We use the well-known Stage simulator to compare the performance of MO-LOST with its predecessors SO-LOST and LOST in a variety of task environments and population sizes. The world size is fixed at 20x20m and contains no fixed obstacles. Robots are Stage's Pioneer 3DX (0.45m long) and SICK LMS200 scanning laser rangefinder models.

Robots are identical except for their permanent task assignment - equal numbers to each task - and starting pose which is the same in each trial, chosen at random and uniformly distribed in the world. Robots start with no information about the location of source and home and must find them by exploration at the start of the trial.

In the screenshots below, Places are large squares, with sources green and homes red. Robots are small red octagons, drawn with yellow diamonds when traveling home with resources. Tiny dots are individuual waypoints stored by one robot. Medium-sized squares in shades of blue are a normalized two-dimensional histogram of robot locations over the previous few minutes. The histogram tends to show the emergent trails quite clearly, with darker blue indicating more defined trails. Recall that "trails" are only perceived by the reader in the pattern of robot behaviour and not represented explicitly by the system.

For lack of space, we present only a selection of results here, in Fig. 2. The performance plot shows that for many population sizes MO-LOST outperforms SO-LOST, while both of these frequently outperform LOST considerably. A T-test shows that MO-LOST outperforms SO-LOST and LOST in most cases, except for the pairs indicated with dotted ellipses. These are most frequent at very low population sizes, where interference is insignificant, and in some easy problems, where MO-LOST was at no advantage over SO-LOST.

4. CONCLUSION

This is the first example of a method to untangle intersecting ant trails from multiple tasks. When trails intersect, the shorter trail is preserved and the longer trail moved, which tends to produce shorter total trail lengths. We determined empirically from a simulation study that Multi-Objective LOST often outperforms Spread-Out LOST and LOST, the original proposal for practical multi-robot foraging using ant-like trails. The performance benefits are most clear in large populations. While MO-LOST is not always applicable, we believe that MO-LOST may be the most practical algorithm yet described for very large population near-decentralized multi-robot, multi-objective foraging from sources to sinks.

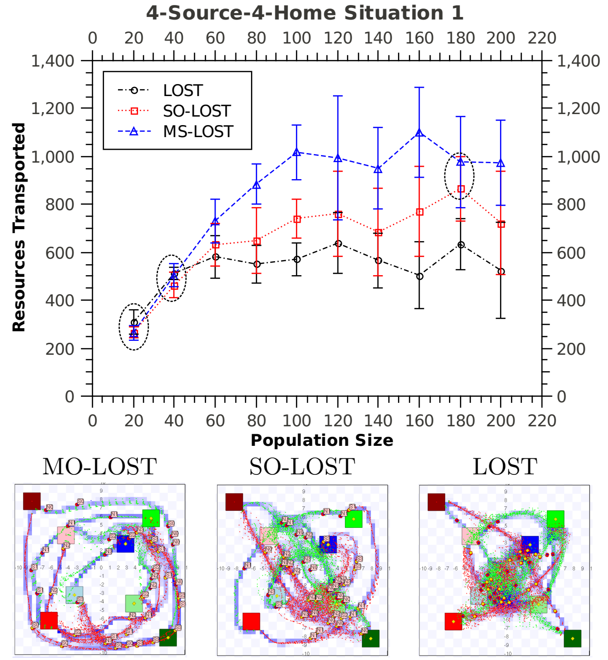


Figure 2: Results: Four-task problems, showing performance vs. population and example trail configurations at the end of a 30 minute trial.
. All results have statistically significant difference except where marked with a dotted ellipse.

5. REFERENCES

[1] M. Dorigo. Optimization, learning and natural algorithms. *PHD thesis, Politecnico di Milano, Italy*, 1992.

[2] S. Iredi, D. Merkle, and M. Middendorf. Bi-criterion optimization with multi colony ant algorithms. In E. Zitzler, L. Thiele, K. Deb, C. Coello Coello, and D. Corne, editors, *Evolutionary Multi-Criterion Optimization*, volume 1993 of *Lecture Notes in Computer Science*, pages 359–372. Springer Berlin / Heidelberg, 2001.

[3] A. Sadat and R. T. Vaughan. So-lost: An ant-trail algorithm for multi-robot navigation with active interference reduction. *In Proceeding of the Twelfth International Conference on Aritifical Life (ALife XII)*, August 2010.

[4] R. T. Vaughan, K. Stoy, A. Howard, G. S. Sukhatme, and M. J. Mataric. Lost: Localization-space trails for robot teams. *IEEE Transactions on Robotics and Autonomous System*, 18(5):796–812, 2002.

Generating Strategies for Multi-Agent Pursuit-Evasion Games in Partially Observable Euclidean Space

(Extended Abstract)

Eric Raboin
University of Maryland
College Park, MD 20742
eraboin@cs.umd.edu

Ugur Kuter
Smart Information Flow Technologies
Minneapolis, MN 55401 USA
ukuter@sift.net

Dana Nau
University of Maryland
College Park, MD 20742
nau@cs.umd.edu

ABSTRACT

We present a heuristic search technique for multi-agent pursuit-evasion games in partially observable Euclidean space where a team of tracker agents attempt to minimize their uncertainty about an evasive target agent. Agents' movement and observation capabilities are restricted by polygonal obstacles, while agents' knowledge of each others' location is limited to direct observation or periodic updates from team members.

Our polynomial-time algorithm is able to generate strategies for games in continuous two-dimensional Euclidean space, an improvement over past algorithms that were only applicable to simple gridworld domains. We show experimentally that our algorithm is tolerant of interruptions in communication between agents, continuing to generate good strategies despite long periods of time where agents are unable to communicate directly. Experimental results also show that our technique generates effective strategies quickly, with decision times of less than a second for reasonably sized domains with six or more agents.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

visibility-based pursuit-evasion, multi-agent planning, game theory

1. INTRODUCTION

Our work introduces a strategy generation technique for multi-agent pursuit-evasion games in continuous, partially observable Euclidean space. We provide a polynomial time algorithm capable of generating online strategies for a team of cooperative tracker agents that wish to pursue an evasive target. The goal of the tracker team is to minimize their uncertainty about the target's location by the end of a fixed time period. The domain may have arbitrarily shaped polygonal obstacles that limit movement as well as observability.

Minimizing uncertainty about a target's location is distinct from the goal assumed by most other pursuit-evasion formalisms. The tracker team must work to maintain visibility on the target, but also to move to strategic locations prior to visibility loss so that recovery will be possible. Previous approaches that sought to maintain visibility on the target for as long as possible [2, 3], or generate patrol strategies to find a hidden target [6, 1], may not be suited for scenarios where the target frequently passes in and out of visibility. Since we want to generate strategies quickly, this also rules out many techniques that are based on deep combinatorial search.

Prior work on this problem included a game-tree search algorithm that could generate strategies for simple gridworld domains, where time was divided into discrete time steps and agents were only permitted to move in one of four cardinal directions [4]. This previous work also assumed that agents would be in constant communication, since it generated trajectories using a heuristic method that required knowing the location of every agent on the team.

In this paper we introduce the *Limited-communication Euclidean-space Lookahead* (*LEL*) heuristic, a method for evaluating tracker strategies in games where agents can move freely in two-dimensional Euclidean space and where there may be long periods of time when communication between agents is interrupted.

Our contributions include–

- An algorithm for computing the *LEL* heuristic in two-dimensional Euclidean space with polygonal obstacles, where communication between agents may be interrupted for long periods of time.
- Complexity analysis showing that our algorithm for computing *LEL* runs in polynomial time with respect to the size of the domain and the number of agents per team.
- Experimental results showing that our algorithm quickly generates strategies for the continuous domain that are twice as effective at retaining visibility on the target when compared to a strategy that follows the shortest path to the target.

2. DEFINITIONS

We define a multi-agent, zero-sum imperfect-information game where a single *target* agent a_0 is pursued by a team of n *tracker* agents $\{a_1, a_2, \ldots a_n\}$. The goal of the tracker team is to minimize its uncertainty about the target's location by the end of the game, while the target agent is free to evade the trackers and move behind obstacles that obstruct visibility.

We assume that each agent a_i is a holonomic point robot with a fixed maximum velocity v_i. The domain can have multiple ob-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

stacles, where each obstacle is a solid polygon in $\mathbb{R}^2$. We define a reachability function $R_i(L, t)$ as the set locations reachable by a_i from any location $l \in L$ by time t, and a visibility function $V_i(L)$ as the set of locations visible to a_i from any $l \in L$.

We define a target's *hidden* region as the set of locations L_{hidden} where the target could be located given the prior observations made by the tracker team. If target a_0 was last observed at location l at time t_j, then it could be located anywhere in $R_0(\{l\}, t_k - t_j)$ by time t_k. This region can be narrowed further by subtracting the locations visible to the tracker team between time t_j and t_k, accounting for the possible trajectories followed by the target during the same time period.

At the end of the game, the utility for the tracker is $-|L_{hidden}|$, while the utility for the target is $|L_{hidden}|$. The best possible outcome for the tracker team is when the exact location of the target is known, meaning L_{hidden} is empty.

3. LEL HEURISTIC

Our work introduces the *Limited-communication Euclidean-space Lookahead* (*LEL*) heuristic, a method of evaluating trajectories for the tracker team by estimating the future size of the target's *hidden* region. This estimate is based off a relaxation of the movement and observation capabilities for each of the agents, which allows the heuristic to be computed in polynomial time but still provide a reasonably good estimate of each trajectory's utility.

To compute LEL, we first determine the set of locations reachable by the target at some time t, given by $R_0(L_{hidden}, t)$. We then determine $R_i(\{l_i\}, t)$ for each tracker agent a_i given their last known location l_i. With this, we can estimate the size of the hidden region at time t by computing,

$$L_{approx}(t) = R_0(L_{hidden}, t) \setminus \bigcup_{i=1}^{n} V_i(R_i(l_i, t)) \qquad (1)$$

which is the set of all locations reachable by the target at time t, minus the set of all locations observable by the tracker. If the target has not been observed by time t, then $L_{approx}(t)$ will be a subset of the actual hidden region, since each tracker a_i will only be able to observe some of the locations in $V_i(R_i(l_i, t))$.

To evaluate a trajectory for agent a_i, we compute the sum of $|L_{approx}(t)|$ over a fixed time interval, where the initial location for a_i is set to a location along the trajectory being evaluated. We have developed a polynomial-time algorithm to do this that utilizes the *Fast Marching Method* [5], a linear-time algorithm for computing the shortest-path distances over two-dimensional Euclidean space. An example of the composite generated by this technique is shown in figure 1, where darker areas represent locations that are contained by L_{approx} for a larger portion of the time interval.

4. EXPERIMENTS

To evaluate *LEL*, we conducted experiments on 500 randomly generated domains with two-dimensional polygonal obstacles. Each trial was run for a fixed amount of time and the size of the hidden region was measured at the end of the game. For comparison purposes we evaluated both *LEL* and the *max-distance* (*MD*) heuristic, a simple hand-coded rule that instructs the tracker team to follow the "shortest-path" to the target [4].

Figure 2 shows the average size of the hidden region at the end of the game for several team sizes, where a smaller hidden region indicates greater certainty about the target's location. Teams using the *LEL* heuristic were more than twice as effective when compared to teams using the *MD* heuristic, even when the teams using the *MD* heuristic had more agents.

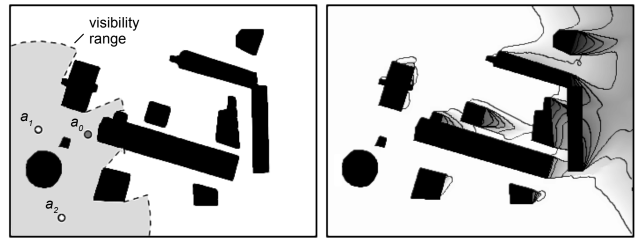


Figure 1: (left) An example game with two trackers (a_1, a_2) and one target (a_0) in their initial positions. (right) *LEL* composite for this state, darker areas indicate potential for visibility loss.

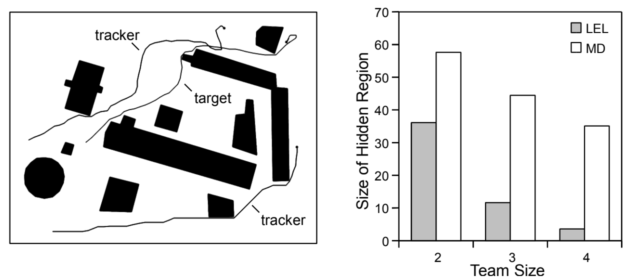


Figure 2: (left) Trajectories generated using the *LEL* heuristic in an example game, (right) Average size of the *hidden* region at the end of the game for different sized teams using the *RLA* and *MD* heuristics. Smaller values equal better performance.

5. ACKNOWLEDGEMENTS

This work was supported, in part, by DARPA and the U.S. Army Research Laboratory under contract W911NF-11-C-0037, and by a UMIACS New Research Frontiers Award. The views expressed are those of the authors and do not reflect the official policy or position of the funders.

6. REFERENCES

[1] S. LaValle, D. Lin, L. Guibas, J. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 737–742, 1997.

[2] T. Muppirala, S. Hutchinson, and R. Murrieta-Cid. Optimal motion strategies based on critical events to maintain visibility of a moving target. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 3826–3831, 2005.

[3] R. Murrieta, A. Sarmiento, S. Bhattacharya, and S. Hutchinson. Maintaining visibility of a moving target at a fixed distance: The case of observer bounded speed. In *Proc. International Conference on Robotics and Automation (ICRA)*, volume 1, pages 479–484, 2004.

[4] E. Raboin, D. S. Nau, U. Kuter, S. K. Gupta, and P. Svec. Strategy generation in multi-agent imperfect-information pursuit games. *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 947–954, 2010.

[5] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci*, pages 1591–1595, 1995.

[6] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21:863–888, 1992.

Induction and Learning of Finite-State controllers from Simulation

(Extended Abstract)

M. Leonetti
Sapienza University of Rome
via Ariosto 25
00185 Rome, Italy
leonetti@dis.uniroma1.it

L. Iocchi
Sapienza University of Rome
via Ariosto 25
00185 Rome, Italy
luca.iocchi@dis.uniroma1.it

S. Ramamoorthy
The University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB, UK
s.ramamoorthy@ed.ac.uk

ABSTRACT

We propose a method to generate agent controllers, represented as state machines, to act in partially observable environments. Such controllers are used to constrain the search space, applying techniques from Hierarchical Reinforcement Learning. We define a multi-step process, in which a simulator is employed to generate possible traces of execution. Those traces are then utilized to induce a non-deterministic state machine, that represents all reasonable behaviors, given the approximate models and planners used in simulation. The state machine will have multiple possible choices in some of its states. Those states are choice points, and we defer the learning of those choices to the deployment of the agent in the actual environment. The controller obtained can therefore adapt to the actual environment, limiting the search space in a sensible way.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Single Agent Learning, Robot planning, Agent development techniques, tools and environments

1. INTRODUCTION

Decision making in unknown environments is characterized by uncertainty at many and different levels. Part of the uncertainty can be captured by models for planning under partial observability, to which a great deal of attention has been payed in recent years. A paramount source of uncertainty lies in the assumptions behind such models themselves, especially if the problem has not been synthesized, but arises from an existing application. This is true regardless of how accurately the model has been designed or learned. Such uncertainty cannot be dealt with at planning time, and requires to monitor the execution in order to identify any discrepancies between what is expected and what is perceived.

Hierarchical Reinforcement Learning (HRL) [1] allows the designer to provide structure to the policies searched, constraining the exploration in fully observable domains. This is a fundamental aspect for real-world applications, as time is a strictly limited resource, and robotic agents are subject to wearing and tearing. The automatic definition of the aforementioned structures is still an open problem, and is usually carried out by hand.

In the following, we propose a method to generate agent controllers automatically, combining several ideas developed in the literature of planning under partial observability and reinforcement learning. We define a multi-step process, in which increasingly accurate models - generally too complex to be used for planning - are employed to generate possible traces of execution by simulation. Those traces are then utilized to induce a machine with non-deterministic states. Those states are *choice points*, and we defer the learning of those choices to the deployment of the agent in the actual environment.

2. GENERATING CONTROLLERS FROM SIMULATION

In this section we define the process to generate a finite state controller for a given problem under partial information.

2.1 Environments, problems, and controllers

We begin with the definition of a *dynamic environment* $\mathcal{E} = \langle \mathcal{A}, \mathcal{O}, \mathcal{S}, \mathcal{I}, \Delta, \Omega \rangle$ in which $\mathcal{A}$ is a finite set of actions, $\mathcal{O}$ is a set of observations, $\mathcal{S}$ is a set of states, $\mathcal{I} \subseteq \mathcal{S}$ is a set of initial states, $\Delta : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is the transition relation, $\Omega : \mathcal{S} \to \mathcal{O}$ is the observation function. We assume to have available only $\mathcal{A}$ and $\mathcal{O}$, while all the other components of the environment are unknown. We also further assume, for this paper, that the environment is deterministic.

A *generalized planning problem* over an environment is a tuple (defined by Bonet et al [2]) $P = \langle F, I, A, G, R, O, D \rangle$ where: F is a set of primitive fluents, I is a set of F-clauses representing the initial situation, A is a set of actions, G is a set of literals representing the goal situation, R is a set of non-primitive fluents, $O \subseteq R$ is the set of axioms defining the fluents in R. In the literature [2, 3], this problem specification is used to derive, by logic, a controller. Due to the uncertainty on the definition of the environment, we only assume to have A, and G. That is, we assume to know the

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

available actions and to be able to recognize the goal states. Such a problem cannot be solved directly by any planner, nor learned as a POMDP, as the model is largely unknown - including the description of the state space. Such a situation is described as *partial information*, which includes partial observability. Being a problem on a actual environment, however, experience can be gathered by acting in it.

A *controller* is a tuple $\mathcal{C} = \langle Q, A^*, O^*, \delta, q_0 \rangle$ where Q is a set of states, $q_0 \in Q$ is the initial state, and A^*, O^*, and δ are the finite set of actions, set of observations, and transition relation respectively. The (partial) transition relation δ maps pairs $\langle q_i, o_i \rangle$ of controller states and observations into actions, and next states q_{i+1}. The controller is *deterministic* if given a pair $\langle q_i, o_i \rangle$, the action and consequently the next states are uniquely determined. A deterministic controller C over an environment $\mathcal{E}$ produces, from each initial state s_0, a single trajectory $t_C(s_0) = \langle o_0, q_0, o_1, q_1, \ldots, o_f, q_f \rangle$. A non-deterministic controller, on the other hand, can produce a set of trajectories that we denote with $T_C(s_0)$.

A trajectory $t_C(s_0)$ is a solution of P from an initial state s_0 iff the terminating observation o_f is such that $o_f \models G$. We are assuming that $o_f \models G \Rightarrow s_f \models G$, that is, if an observation fulfills the goal specification, the underlying, unobservable, state is a goal state. A deterministic controller *solves* a problem P over an environment $\mathcal{E}$ iff each trajectory $t_C(s_0)$, from each initial state s_0, is a solution from s_0. We say that a non-deterministic controller C *can solve* a problem P if $\exists t \in T_C(s_0)$ such that t is a solution from s_0.

Finally, we define a *restriction* of a problem P to $\hat{I} \subseteq I$ as the sub-problem $P(\hat{I}) = \langle F, \hat{I}, A, G, R, O, D \rangle$.

2.2 Simulators

We assume the existence of another environment $\mathcal{E}'$ on which we can define a problem P'. Informally, $\mathcal{E}'$ is a simulator for $\mathcal{E}$, and comprises the designer knowledge of the environment. The characteristic of a simulator is to be a model that provides an approximation of the environment $\mathcal{E}$, that is usually too complex to be used for planning. In such a model, however, experience can be gathered much more cheaply than in $\mathcal{E}$. We acknowledge the inescapable differences between $\mathcal{E}$ and $\mathcal{E}'$, and account for a learning phase to optimize the controller generated in the latter to act in the former. We do not define any direct relationship between $\mathcal{E}$ and $\mathcal{E}'$, we shall rather establish one through controllers.

Although we have a complete specification of both $\mathcal{E}'$ and P', we only use them through simulation, that is, to generate trajectories. P' can be partially observable, but does not necessarily have to. Furthermore, we assume the existence of a *decision maker* that can solve P' in $\mathcal{E}'$.

2.3 Controller induction

The decision maker is deployed in $\mathcal{E}'$ to generate trajectories $t'(s'_0)$ that are solutions to P'.

Considering each action in A' as a symbol, the set of trajectories that are solutions to P', from which observations are removed, form a *language*. We induce a finite deterministic automaton $\mathcal{C}' = \langle Q', A', \emptyset, \delta', q'_0 \rangle$ that accepts such a language. Note how this is equivalent to a controller with an empty observation set.

Finally, we expand the edges of $\mathcal{C}'$ to accommodate the observations of P. We define a controller $\hat{\mathcal{C}} = \langle Q', A', O, \hat{\delta}, q'_0 \rangle$ obtained from $\mathcal{C}'$ such that for each observation $o \in O$, $\hat{\delta}$ connects a state q'_i to a state q'_j when observing o and by executing $a' \in A'$, if and only if δ' connects q'_i to q'_j by executing a'.

If the controller $\hat{\mathcal{C}}$ obtained through the process just described can solve a reduction of P in $\mathcal{E}$ we say that the composition of the decision maker, $\mathcal{E}'$, P', and the method used to induce the automaton is *admissible*. This can be verified by executing $\hat{\mathcal{C}}$ in $\mathcal{E}$.

2.4 Reinforcement learning on controllers

The controller $\hat{\mathcal{C}}$, obtained through simulation and automaton induction, provides at the same time a constraint to the possible behaviors, and a partial specification of a solution for a problem that could not otherwise be solved. If such an automaton is deterministic no further improvement is possible, and it constitutes a completely specified solution to P. More interestingly, if such a controller is non-deterministic, learning in the actual environment can determine (and be limited to) the behavior at choice states. In order to generate a non-deterministic controller $\hat{\mathcal{C}}$ the decision maker must be able to produce more than a solution trajectory, from at least some initial states. The two options are not mutually exclusive.

A stochastic process is derived from $\hat{\mathcal{C}}$ according to the procedure presented by Leonetti and Iocchi [4]. The resulting process is Non-Markovian if the memory embedded in the controller and the observations are not a sufficient statistics for the reward. If that is the case, a policy can still be learned with a specific algorithm [6, 5].

3. CONCLUSION

We proposed a method to generate agent programs, in the form of state machines, by combining different components: an initial decision maker, available to the designer to include any previous knowledge about the task; a simulator, that is, a model too complex to be used for planning, but from which possible trajectories can be extracted; the induction of an automaton that accepts the extracted trajectories; and finally reinforcement learning, on the derived state machine, directly in the actual domain. Simulators are commonly employed, but are rarely integral parts in the development of agent programs. Constraining the final learning, through the interaction of all those components, significantly limit the search space in the actual domain.

4. REFERENCES

[1] A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1-2):41–77, 2003.

[2] B. Bonet, H. Palacios, and H. Geffner. Automatic derivation of memoryless policies and finite-state controllers using classical planners. In *Proc. of ICAPS*, pages 34–41, 2009.

[3] G. De Giacomo, F. Patrizi, and S. Sardina. Agent programming via planning programs. In *Proc. of AAMAS*, pages 491–498, 2010.

[4] M. Leonetti and L. Iocchi. Improving the performance of complex agent plans through reinforcement learning. In *Proceedings of AAMAS*, volume 1, pages 723–730, 2010.

[5] M. Leonetti, L. Iocchi, and S. Ramamoorthy. Reinforcement Learning Through Global Stochastic Search in N-MDPs. In *Proceedings of ECML-PKDD*, volume 2, pages 326–340, 2011.

[6] T. J. Perkins. Reinforcement learning for POMDPs based on action values and stochastic optimization. In *Proceedings of the National Conference on Artificial Intelligence*, pages 199–204, 2002.

Spatial awareness in robotic swarms through local wireless communications

(Extended Abstract)

Frederick Ducatelle
Dalle Molle Institute for Artificial Intelligence (IDSIA)
Lugano, Switzerland
frederick@idsia.ch

Gianni A. Di Caro
Dalle Molle Institute for Artificial Intelligence (IDSIA)
Lugano, Switzerland
gianni@idsia.ch

Luca M. Gambardella
Dalle Molle Institute for Artificial Intelligence (IDSIA)
Lugano, Switzerland
luca@idsia.ch

ABSTRACT

We propose a fully distributed approach to endow robots in a swarm with awareness of their relative position with respect to the rest of the swarm. Such spatial awareness can be used to support spatially differentiated task allocation or for pattern formation. The approach we propose only relies on local communications and is based on a combination of distributed consensus and load balancing. We test the effectiveness of our algorithm in extensive simulation tests and we also validate it in experiments with real robots.

Categories and Subject Descriptors

I.2.9 [**Robotics**]; I.2 [**Distributed Artificial Intelligence**]: Coherence and coordination; C.2 [**Computer Communication Networks**]: Distributed applications

General Terms

Algorithms

Keywords

Swarm robotics, geometric bisectioning, spatial aggregation

1. INTRODUCTION

The aim of this work is to endow robots in a swarm with awareness of their relative position with respect to the rest of the swarm. Such *spatial awareness* can be used to support spatially differentiated task allocation (e.g., split the swarm in different, spatially close, groups, and let each group engage in a different task, such as exploring different regions of an environment), or for pattern formation, among others. The task we focus on is to assign the robots of the swarm to two different classes, C_0 and C_1, in such a way that *the two classes are spatially segregated*: the robots in class C_0 are found on one side of the swarm, and the robots in class C_1 on the other side of the swarm. The problem that we are solving can be formally described as follows. Let $G(V, E)$ be a Euclidean graph where the node set V represents geometric entities, such as robots, positioned in the plan. Nodes are able to communicate with each other over a wireless medium. Two nodes i and j are connected by a link $(i, j) \in E$ if: (i) their Euclidean distance is less than or equal to the maximum communication range R_{max} (*range-constrained connectivity*), and (ii) no major occlusions are present between the two nodes (*line-of-sight communication constraint*). Each node only knows about its neighbors, no other network information is assumed. The objective is to find, adopting a fully decentralized approach, a geometric partitioning of the graph in k classes, where each class contains (approximately or precisely) the same number n_k of nodes, and the nodes in each partition are geometrically close to each other. We focus on the case $k = 2$. Figure 1 illustrates different partitionings in two classes. We aim to obtain partitionings like in (a) and (b).

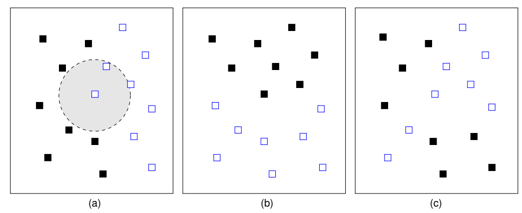


Figure 1: Examples of different ways to realize a geometric partitioning in two classes (indicated by the black and white squares) given a communication range (indicated by the grey disk in (a)).

To solve this problem, we look for an algorithm that is robust, scalable, efficient, works in a decentralized way, and has limited requirements in terms of available sensor or actuators. We rule out the use of global positioning information (not always available, especially in indoor environments) as well as the use of physical mobility (not always possible, slow, and energy-greedy). Instead, we propose an algorithm which uses only *local communication*. The robots/nodes only need to be able to identify their neighbors and communicate with them. Only a relatively *low channel bandwidth* is required to let the algorithm working effectively. We consider the general case of robots/nodes equipped with a wireless communication interface. The algorithm combines elements from different approaches to similar problems: algorithms for solving *minimum bisection problems* [1]; algorithms for *swarm robotics aggregation* [3], and distributed algorithms for *consensus load balancing* [2].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ALGORITHM OVERVIEW

The degree of membership of a robot i to one of the two classes C_0 and C_1 is represented by using *load variables* $u_i \in [0, 1]$. $u_i = 1$ means full membership of i to class C_1, $u_i = 0$ means membership to class C_0, while intermediate values indicate different degrees of membership to C_0 and C_1. At the start, each robot i decides with a probability of 0.5 whether it is loaded or not, and sets accordingly a variable u_i. Each robot i also keeps a value $v_i \in [0, 1]$, which is an estimate of *how loaded on average* the robots in its neighborhood are. After the initialization, the robots start to communicate locally, with two goals: to update the estimate v_i, and to let loads travel through the swarm, until they stop at different robots. Local values of v_j variables decide when to leave, where to go, and where to stop. Load traveling across the robot network goes in a number of phases. Each phase aims, in different ways, to eventually create a single connected cluster of robots of class C_1 which is spatially well separated from the cluster of unloaded robots (i.e., of class C_0), as illustrated in Figure 1 (a) and (b).

In **phase 0**, following the first creation or the reception of a load, the load leaves robot i if i's neighborhood stays unloaded for a certain amount of time (i.e, v_i is less than a threshold v_{min}). **Phase 1** is a *steepest ascent* with respect to current load distribution: the load is iteratively sent to the neighbor j with the highest v_j, until the local maximum is reached. **Phase 2** is a *steepest descent*: the load moves to a local minimum of v_j, meaning that it looks for an area which is unloaded. **Phase 3** is again a *steepest ascent*: the load greedily looks for a new loaded area, possibly with a higher value of v_j than that of phase 1. **Phase 4** is a *slowest descent*: the load moves from robot to robot to decreasing values of u_i, until it reaches an unloaded robot ($u_j = 0$), where it moves back to phase 0. The idea is that the slow descent will make the loads rather go towards areas where there are only a few unloaded nodes, so that the load goes to fill small empty pockets. If no unloaded robot is found before reaching a local minimum, the load takes a random step, and returns to phase 1: start all over again. Once the load has reached phase 0 at a robot i, it sets the local value u_i to 1. If more loads cluster around robot i, the local value v_i will grow, keeping the load stationary at robot i. In this way, $v_i \geq v_{min}$, preventing the load to leave i and letting the cluster grow further. Eventually, loads stop moving, converging in two spatially separated clusters.

3. EXPERIMENTAL RESULTS

We ran simulation tests considering as reference robot the *foot-bot*, developed during the *Swarmanoid* project (`http://www.swarmanoid.org`). For the work presented here, the relevant on-board device is the infrared-based *range-and-bearing* that provides *line-of-sight* communication. It sends messages of 10 bytes at a rate of 10 messages per sec.

Each simulation test (50 trials per test) runs for 1200 time steps = 2 minutes. We measure two things: *linear separability* and *imbalance*. Linear separability is evaluated by fitting a line to the space in which the robots are placed, in such a way that the loaded robots are found on one side of the line and the unloaded ones on the other side. Results for linear separability range from 0 (optimal) to 0.5 (worst). The imbalance evaluates whether the two classes are of the same size. We report the number of robots in the smallest of the two classes, divided by the total number of robots in the swarm. The optimal value is 0.5, the worst possible is 0.

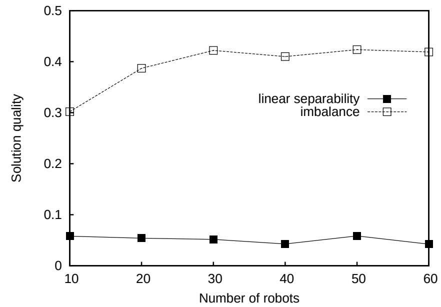


Figure 2: Experiments with varying number of robots maintaining constant the area.

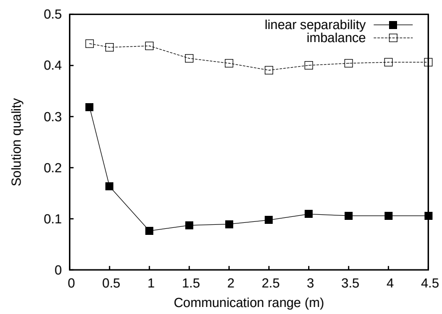


Figure 3: Experiments with varying communication range maintaining fixed to 50 the number of robots.

In a first series of tests, we vary the *number of robots* in the swarm, from 10 up to 60. The communication range of the robots is limited to 1 m. The results in Figure 2 show that the algorithm works quite well in separability, and is robust with respect to the number of robots, although for the smallest swarms, results become a bit less good because of less connectivity and too few loads around.

In a second series of tests, we vary the *communication range*, from 0.25 up to 4.5 m, fixing the the number of robots to 50. The results in Figure 3 show that the algorithm works badly at short communication ranges, due to the fact that the communication network gets disconnected. For medium and high communication ranges, the results are very good.

We also ran a limited set of experiments using a small swarm of 15 foot-bots deployed in different initial configurations. Sample videos are available here: `http://www.idsia.ch/~gianni/SwarmRobotics/GeometricSplitting.html`.

4. REFERENCES

[1] R. Battiti and A. Bertossi. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Transactions on Computers*, 48(4), April 1999.

[2] M. Franceschelli, A. Giua, and C. Seatzu. A gossip-based algorithm for discrete consensus over heterogeneous networks. *IEEE Transactions on Automatic Control*, 55(5):1244–1249, 2010.

[3] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, and G. Theraulaz. Collective decision-making by a group of cockroach-like robots. In *Proceedings of IEEE Swarm Intelligence Symposium (SIS)*, pages 233–240, 2005.

Multi Robot Learning by Demonstration

(Extended Abstract)

Michiel Blokzijl-Zanker
Dept. of Electrical and Electronic Engineering
Imperial College London
London SW7 2AZ
michiel.blokzijl07@imperial.ac.uk

Yiannis Demiris
Dept. of Electrical and Electronic Engineering
Imperial College London
London SW7 2AZ
y.demiris@imperial.ac.uk

ABSTRACT

In this paper, we investigate the feasibility of a Multi Robot Learning by Demonstration system, which allows multiple teachers to give a demonstration to multiple robots simultaneously. A novel, complete end-to-end system was developed, which extracts data from a live human group demonstration, and allows the robots to imitate the demonstration by adapting the demonstration dataset to the current, possibly different environment. The complete system was evaluated using a series of increasingly difficult benchmark experiments, including a collaborative door opening experiment using a group of heterogeneous robots. The results showed, that the system is resistant to changes in the environment, as it was possible to give a demonstration in one environment, move the robots to a physically different but similar location, where the robots could still imitate the demonstration in this new context. The door opening experiment also shows that this system can be used to demonstrate and learn collaborative behaviour. Our results demonstrate a novel and promising method for teaching a group of robots to perform a joint task by human team demonstration.

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Artificial Intelligence—*Robotics*

General Terms

Algorithms, Experimentation

Keywords

multi-robot systems, robot, learning, learning by demonstration, adaptation, template matching, imitation

1. INTRODUCTION

The Learning by Demonstration (LbD) paradigm has been suggested as a method to tackle the complexity of robot programming. To date however, research into LbD systems has mostly focused on "a single robot being taught by a single teacher" [1]. The scenario of multiple teachers teaching multiple robots has so far received little research attention, especially regarding LbD systems in which multiple teachers teach multiple robots *simultaneously* [4]. This paper will investigate the feasibility of such a multi robot Learning by Demonstration system by developing and testing such an end-to-end system.

One challenge common to most LbD systems is the question of how to deal with an "undemonstrated state" [1]. This challenge arises because it is unlikely for a teacher to be able to demonstrate the correct behaviour for every possible state the robots may find themselves in [1], and hence it is necessary to develop a strategy for coping with new situations. Here, we shall take the approach of *adapting* the demonstration dataset to a new situation, using the (plan) adaptation algorithm presented in [5]. The plan adaptation algorithm was presented in the context of multi robot path planning, and essentially represents the environment using point features. These are recorded for the demonstration (the *template*) and the new imitation environment (the *target*). Then, the correspondences between features in the template and the target are found, and used to find a mapping that can *warp* the target to fit the new context, yielding an *adapted plan*.

In order to develop a full, end-to-end multi robot learning by demonstration system, based on the described plan adaptation approach, three components are required: a "template extraction system", which extracts the demonstration data from the environment, a "plan adaptation system", which adapts one or more templates to the new context, and a "plan execution system", which executes the adapted plan on the robots.

2. METHODOLOGY

2.1 Template extraction

The template extraction process starts with creating a laser-scan based map of the local demonstration context. Next, this map is converted to "corner" and "wall" point features, using Harris corner detectors [3], and a sliding window algorithm, respectively. The latter classifies a static obstacle as a wall feature if no other wall feature has been found within the current window.

During the demonstration, the robot location on the map is tracked using AMCL [2], and the features close to the robots are identified (marked) using a K-Nearest Neighbour (KNN) search. More specifically, the feature closest to the robot that has *not yet been marked* will be marked. Next, about half of the selected "wall" features will be deselected to avoid over-constraining the template, which is done us-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

ing another KNN search based algorithm: this one essentially tries to "hop" from one marked wall feature to the next, deselecting every other one. The remaining marked features will be "required to match", (F_r, as defined in [5]), and the marked corner features will additionally be required to match to exactly one feature in the target (F_{11} in [5]). This data gives us the template.

2.2 Plan adaptation & execution

The plan adaptation system extracts a representation of the environment and the robots' relative location using the same techniques as described in section 2.1. These are sent to the plan adaptation algorithm of [5], which has been extended to take the waypoint order into account when mapping the robots in the template to the ones in the target. The resulting adapted waypoints are then sent to the robots.

The last part consists of sending the adapted waypoints to the robots, taking into account the (adapted) waypoint order. We used the ROS navigation stack to drive to robots to waypoints.

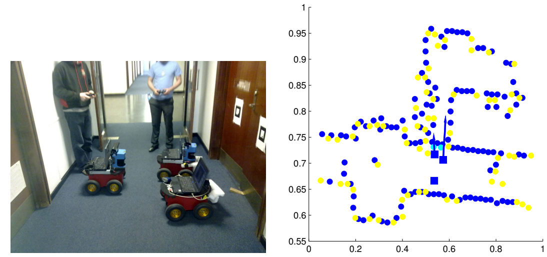
3. RESULTS & DISCUSSION

In order to evaluate the end-to-end system, a series of increasingly difficult benchmark experiments were designed. These started with two humans demonstrating to two robots how to drive straight ahead along a corridor, and then requesting an imitation in the same context. Next, the various parameters of the experiments were varied: a) the number of robots was increased to three, b) the demonstrated movements were varied (e.g. curved trajectories around a corner, intersecting trajectories), c) the demonstration and imitation environments were changed, such that the robots were given a demonstration in one context and asked to imitate in another, and d) the robots' relative displacement (their formation) was varied. It was also verified, that the template could be mirrored, scaled, translated and rotated to match a target. Furthermore, a collaborative door opening experiment was performed to see whether this type of system could be used to demonstrate collaborative behaviour to a group of robots (see Figure 1).

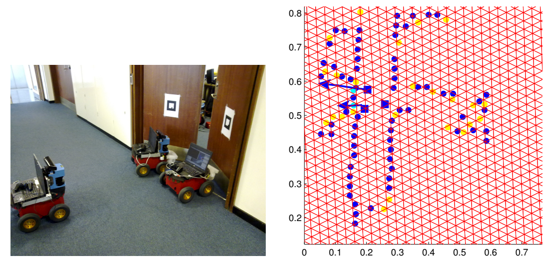
From these experiments, it was observed that the system was resilient to changes in the context (environment): the robots could be demonstrated how to open a door in one location, and then perform the imitation at another, similar door in a different environment. The algorithms were also tolerant to about 30-60cm of displacement of each robot from their original position in the robots' formation (in a 2m wide corridor). We verified this by taking a template and a matching target from our experiments, and applying a simulated, spiral-shaped displacement to one robot at a time, and running the matching algorithm on the changed target. This tolerance will vary depending on the template and target though, and further work is required to quantify this for a larger sample of scenarios.

4. CONCLUSION

This work presented one approach to developing an end-to-end Multi Robot Learning by Demonstration system, that allows multiple teachers to give a demonstration simultaneously to multiple robots, thus allowing the demonstration of collaborative behaviour. The experiments consisted of both a series of tests used to gain insights into the system's performance in various scenarios, as well as an experiment in which a group of robots was demonstrated how to collaboratively open a marked door. The experiments showed, that the system coped well with changes in the environment, and that it allowed for small displacements of the robots relative to each other. In conclusion, this paper showed the feasibility of a Multi Robot Learning by Demonstration system and will hopefully lead to further work in that area, ideally leading to users being able to customise the behaviour of groups of robots in the field.



(a) Template obtained from demonstration



(b) Autonomous execution of adapted plan

Figure 1: The results of the collaborative door opening experiment. Two robots are equipped with bumpers, while the one at the back uses the Microsoft XBox Kinect camera to detect the markers (cyan dots). Blue dots represent "wall features", while yellow dots represent corners.

5. REFERENCES

[1] B. D. 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] D. Fox. Adapting the sample size in particle filters through kld-sampling. *The International Journal of Robotics Research*, 22(12):985–1003, 2003.

[3] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[4] M. F. Martins and Y. Demiris. Learning multirobot joint action plans from simultaneous task execution demonstrations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 931–938, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[5] B. Takacs and Y. Demiris. Multi-robot plan adaptation by constrained minimal distortion feature mapping. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 742 –749, May 2009.

Distributed Value Functions for the Coordination of Decentralized Decision Makers

(Extended Abstract)

Laëtitia Matignon, Laurent Jeanpierre and Abdel-Illah Mouaddib
Université de Caen Basse-Normandie, CNRS - UMR6072 GREYC - F-14032 Caen, France
{laetitia.matignon,laurent.jeanpierre,abdel-illah.mouaddib}@unicaen.fr

ABSTRACT

In this paper, we propose an approach based on an interaction-oriented resolution of decentralized Markov decision processes (Dec-MDPs) primary motivated by a real-world application of decentralized decision makers to explore and map an unknown environment. This interaction-oriented resolution is based on distributed value functions (DVF) techniques that decouple the multi-agent problem into a set of individual agent problems and consider possible interactions among agents as a separate layer. This leads to a significant reduction of the computational complexity by solving Dec-MDPs as a collection of MDPs. Using this model in multi-robot exploration scenarios, we show that each robot computes locally a strategy that minimizes the interactions between the robots and maximizes the space coverage of the team. Our technique has been implemented and evaluated in simulation and in real-world scenarios during a robotic challenge for the exploration and mapping of an unknown environment by mobile robots. Experimental results from real-world scenarios and from the challenge are given where our system was vice-champion.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics—*Autonomous vehicles*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Theory, Experimentation

Keywords

Cooperative multi-robot systems, robot coordination, robot planning, multi-robot exploration, distributed problem solving

1. INTRODUCTION

The approach developed in this paper is primary motivated by a real-world application of decentralized decision makers for an exploration and mapping multi-robot system. Our system has been developed and applied successfully in real-world scenarios during a DGA[1]/ANR[2] robotic challenge[3]. We focus only on the decision model allowing robots to cooperatively explore an unknown area and efficiently cover the space by reducing the overlap between explored areas of each robot. Such multi-robot exploration strategies have been proposed. They adopt either central agents or negotiation protocols with complicated processes [9, 1]. Besides existing works do not address the local coordination problem. So we took an interest in decentralized partially observable Markov decision processes (Dec-POMDPs) and their recent interaction-oriented (IO) resolution [8, 5, 2]. It takes advantage of local interactions and coordination by relaxing the most restrictive and complex assumption consisting in considering that agents are permanently in interaction. It is based on a set of interactive individual decision making problems and reduces the complexity of solving Dec-POMDPs thereby becoming a promising direction concerning real-world applications of decentralized decision makers. Consequently we propose in this paper an IO approach using distributed value functions so as to compute multi-robot exploration strategies in a decentralized way.

2. DISTRIBUTED VALUE FUNCTIONS

We propose an IO resolution of decentralized decision models using distributed value functions (DVFs) introduced in [6]. Our robots are independent and can share information by communication leading to some kind of observability completion. We assume full local observability for each robot and limited share of information. So our approach takes place in the Dec-MDP framework. DVF describes the Dec-MDP with a set of individual agent problems (MDPs) and considers possible interactions among robots as a separate interaction class where some information between robots are shared. This leads to a significant reduction of the computational complexity by solving Dec-MDP as a collection of MDPs. This could be represented as an IO resolution with two classes (no interaction class and interaction class) where each robot computes locally a strategy that minimizes conflicts, *i.e.* that avoids being in the interaction class. The interaction class is a separate layer solved independently by computing joint policies for these specific joint states.

DVF technique allows each robot to choose a goal which should not be considered by the others. The value of a goal depends on the expected rewards at this goal and on the fact that it is unlikely selected by other robot. Our DVF

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1]French Defense Procurement Agency.
[2]French National Research Agency.
[3]http://www.defi-carotte.fr/

is defined solely by each robot in an MDP $< S, A, T, R >$. In case of permanent communication, robot i knows at each step t the state $s_j \in S$ of each other robot j and computes its DVF V_i according to :

$$\forall s \in S \quad V_i(s) = \max_{a \in A} \Bigg(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \\ [V_i(s') - \sum_{j \neq i} f_{ij} P_r(s'|s_j) V_j(s')] \Bigg) \qquad (1)$$

where $P_r(s'|s_j)$ is the probability for robot j of transitioning from its current state s_j to state s' and f_{ij} is a weighting factor that determines how strongly the value function of robot j reduces the one of robot i. Considering our communication restrictions, the robots cannot exchange information about their value functions. So we relax the assumptions concerning unlimited communication in DVF technique: each robot i can compute all V_j by empathy. Thus each robot computes strategies with DVF so as to minimize interactions. However when situations of interaction occur, DVF does not handle those situations and the local coordination must be resolved with another technique. For instance joint policies could be computed off-line for the specific joint states of close interactions.

3. EXPERIMENTS AND RESULTS

In these section is given an overview of our experiments. More details concerning our MDP model, our experimental platforms and our results can be found in [4]. First we made simulations with Stage[4] using different number of robots and various simulated environments. In fig. 1, we plot the time it takes to cover various percentages of the environment while varying the number of robots. During the beginning stage, robots spread out to different areas and covered the space efficiently. However, there is a number of robots beyond which there is not much gain in the coverage and that depends on the structure of the environment [7]. In the hospital environment, four robots can explore separate zones but the gain of having more robots is low compared to the overlap in trajectories and the risk of local interactions. Second we performed real experiments with our two robots besides the ones made during the challenge. Videos, available at `http://lmatigno.perso.info.unicaen.fr/research`, show different explorations of the robots and some interesting situations are underlined as global task repartition or local coordination.

4. CONCLUSIONS AND PERSPECTIVES

Our approach addresses the problem of multi-robot exploration with an IO resolution of Dec-MDPs based on DVFs. Experimental results from real-world scenarios and our vice-champion rank at the robotic challenge show that this method is able to effectively coordinate a team of robots during exploration. Though our DVF technique still assumes permanent communication similarly to most multi-robot exploration approaches where robots maintain constant communication while exploring to share the information they gathered and their locations. For instance classical negotiation based techniques assume permanent communication. However, permanent communication is seldom the case in

[4]http://playerstage.sourceforge.net/

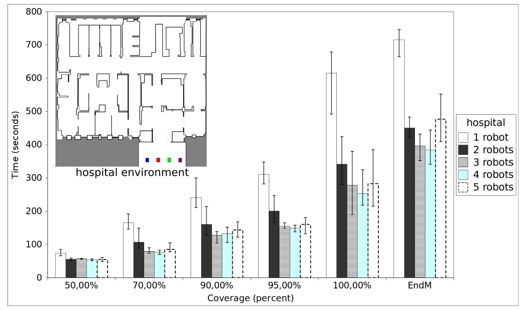


Figure 1: Hospital environment from Stage with starting positions and results averaged over 5 simulations.

practice and a significant difficulty is to account for potential communication drop-out and failures that can happen during the exploration leading to a loss of information that are shared between robots. Some recent multi-robot exploration approaches that consider communication constraints only cope with limited communication range issue and do not address the problem of failures as stochastic breaks in communication. So our short-term perspective is to extend our DVF to address stochastic communication breaks as it has been introduced in [3].

5. ACKNOWLEDGMENTS

This work has been supported by the ANR and the DGA (ANR-09-CORD-103) and jointly developed with the members of the Robots_Malins consortium.

6. REFERENCES

[1] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21:376–386, 2005.

[2] A. Canu and A.-I. Mouaddib. Collective decision-theoretic planning for planet exploration. In *Proc. of ICTAI*, 2011.

[3] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib. Distributed value functions for multi-robot exploration: a position paper. In *AAMAS Workshop: Multiagent Sequential Decision Making in Uncertain Domains (MDSM)*, 2011.

[4] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib. Distributed value functions for multi-robot exploration. In *Proc. of ICRA*, 2012.

[5] F. Melo and M. Veloso. Decentralized mdps with sparse interactions. *Artif. Intell.*, 175(11):1757–1789, 2011.

[6] J. Schneider, W.-K. Wong, A. Moore, and M. Riedmiller. Distributed value functions. In *Proc. of ICML*, pages 371–378, 1999.

[7] R. Simmons, D. Apfelbaum, W. Burgard, D. an Moors M. Fox, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proc. of the AAAI National Conf. on Artificial Intelligence*, 2000.

[8] M. T. J. Spaan and F. S. Melo. Interaction-driven markov games for decentralized multiagent planning under uncertainty. In *AAMAS*, pages 525–532, 2008.

[9] R. Zlot, A. Stentz, M. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. of ICRA*, volume 3, pages 3016–3023, May 2002.

Auctioning Robotic Tasks With Overlapping Time Windows (Extended Abstract)

Ernesto Nunes*
Dept of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
enunes@cs.umn.edu

Maitreyi Nanjanath
Dept of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
nanjan@cs.umn.edu

Maria Gini*
Dept of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
gini@cs.umn.edu

ABSTRACT

This work investigates allocation of tasks to multi-robots when tasks are spatially distributed and constrained to be executed within assigned time windows. Our work explores the interaction between scheduling and optimal routing. We propose the Time-Sensitive Sequential Single-Item Auction algorithm as a method to allocate tasks with time windows in multi-robot systems. We show, experimentally, that the proposed algorithm outperforms other auction algorithms that we modified to handle time windows.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Performance, Experimentation

Keywords

Auctions, time windows, task allocation, multi-robot systems

1. INTRODUCTION

Many real world problems require tasks to be executed within a specified time window. For example, a region may need surveillance at regular intervals or at specific hours, and in search and rescue, much of the exploration has to be done in well defined stages. Time windows make task allocation harder as it is no longer possible to arbitrarily arrange the order of execution of tasks to decrease travel costs.

Auctions are becoming popular for allocating tasks to robots [4]. However, limited attention has been devoted to allocation of tasks that have to be completed within specified time windows (see [3] for an example), and even less for tasks that have overlapping time windows. When time windows are pairwise disjoint, tasks can be strictly ordered and robots can choose any permutation of tasks. With overlapping time windows this is no longer possible.

Time windows are often treated as soft constraints on time of arrival to a task location (see [5] for an example). In this case late arrival to a task is subject to a penalty, which increases the cost but does not affect feasibility. We treat time windows as hard constraints, as a robot can no longer perform a task to which the robot arrives late.

We are interested in approximate algorithms that are computationally efficient and that minimize the sum of the path costs over all the robots, while avoiding time conflicts. The main contributions of this paper are the Time-Sensitive Sequential Single-Item Auction (TS-SSIA) algorithm and experimental results comparing its performance to other single item auction algorithms.

2. AUCTIONS WITH TIME CONSTRAINTS

Formally, $r = \{r_1, r_2, \ldots, r_n\}$ is a set of robots; each robot has a Cartesian location (x_{r_i}, y_{r_i}). Let $t = \{1, 2, \ldots, m\}$ be a set of tasks; each task j has a time window defined by its earliest start time $es(j)$, latest finish time $lf(j)$, location (x_j, y_j), and time duration $dur(j)$. The objective is to assign to each robot r_i a subset of tasks $\{j, j+1, ...k\} \subseteq t$ such that that the sum of the path costs is minimized while completing the largest number of tasks.

For each task we compute its latest start time $ls(j) = lf(j) - dur(j)$. Let $RT(r_i, k, j)$ be the time it takes robot r_i to travel between tasks k and j. Then, lateness is defined as $l(r_i, k, j) = lf(k) + RT(r_i, k, j) - ls(j)$. If $l(r_i, k, j) > 0$, the robot r_i is not able to reach task j in time to do the task.

In an auction allocation method, the robots bid on tasks based on the amount of effort they need to complete them, which includes the cost of traveling to the task and any additional cost for doing the task itself.

Combinatorial auctions produce optimal solutions, but finding a set of non-conflicting bids that maximizes revenue is NP-complete and impractical for large numbers of tasks, hence it is common to auction each task separately. When all tasks are put up for bids at the same time in a *parallel single-item auction* the solution can be far from optimal, because robots cannot account in their bids for complementarities among tasks. This shortcoming can be reduced by repeating the auctions periodically at fixed time interval [1]. Alternatively, the tasks can be put up for bids one at a time

*Work supported in part by NSF IIP-0934327 and the Safety, Security, and Rescue Research Center at the University of Minnesota

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Number of Tasks	Parallel single-item auction		Sequential single-item auction		TS-SSIA	
(10 robots)	μ	(σ)	μ	(σ)	μ	(σ)
30	2062	(178.04)	2061	(184.37)	*1965*	(170.23)
50	3420	(232.17)	3392	(228.03)	*3257*	(223.92)
70	4824	(281.54)	4785	(278.90)	*4590*	(265.85)
90	6396	(369.55)	6117	(367.01)	*5869*	(343.03)
(20 robots)	μ	(σ)	μ	(σ)	μ	(σ)
30	1894	(180.96)	1893	(183.58)	*1845*	(174.01)
50	3217	(225.67)	3216	(223.27)	*3106*	(218.33)
70	4496	(317.13)	4489	(317.15)	*4317*	(278.15)
90	5766	(330.75)	5762	(320.35)	*5528*	(286.40)
(50 robots)	μ	(σ)	μ	(σ)	μ	(σ)
1000	58520	(909.12)	57710	(948.68)	*56350*	(889.10)

Table 1: Solution cost (mean and standard deviation) of auction methods in a 100×100 grid, averaged over 30 runs. TS-SSIA orders tasks in ascending order of earliest start times. Numbers in italics are the best results.

in a *sequential single-item auction (SSIA)* [2]. In this case, robots account for previous task commitments while bidding on the next task, so they bid the insertion cost in their current path. When the sum of the path costs is minimized, the solution is a constant factor away from the optimum.

We extend these auction methods by enabling them to deal with time windows. In *repeated parallel single-item auctions with time windows* the auctioneer chooses each winning bid depending on two criteria: the cost of the winning bid is the minimum amongst all bids, and the winning robot has no time conflict with the task. In *sequential single-item auctions with time windows* the winner of each task is the robot with the minimum insertion cost into its path and no time conflict. The addition of time windows makes the solution to depend on the order in which tasks are put up for auction. This suggests a change in the algorithm to take advantage of both spatial and temporal synergies among the tasks.

In *time-sensitive sequential single-item auctions (TS-SSIA)* the auctioneer orders tasks for auction according to one or more sorting criteria. We experimented with ordering tasks by their earliest or latest start times, either in ascending or descending order. The key difference between TS-SSIA and the sequential single-item auction algorithm is the fact that that the auction process is informed by the time windows of the tasks. By ordering the tasks up for bids according to their time windows, robots can easily take into account the time constraints in their bids instead of considering only their distance to the tasks.

3. CASE-STUDY: TASKS ON A 2D GRID

For this set of experiments, we created a 100×100 Cartesian grid and distributed tasks uniformly on the grid. Any robot can reach any task with a cost proportional to the Cartesian distance between the robot and the task. We used 10, 20 and 50 robots and $30, 50, 70, 90$ and 1000 tasks respectively. The cost for each task was uniformly distributed between $0 - 100$. The goal of the experiment is to compare the sum of the path costs produced by TS-SSIA with those produced by the other single-item methods.

In this experimental setup, there was no difference in solution cost between sorting tasks by start times or by deadline, consequently, we present only results for sorting by start times. When tasks are sorted in descending order, on average, the algorithms produced higher costs than when tasks are sorted in ascending order.

In Table 1 we see that TS-SSIA outperforms the other single item auction methods. TS-SSIA reports gains as large as 8.2% against parallel and 4.1% against sequential (90 tasks 10 robots case). The difference in performance between TS-SSIA and the sequential algorithm and between TS-SSIA and the parallel algorithm is statistically significant with p-values 0.0017 and 0.0002 respectively.

4. CONCLUSIONS AND FUTURE WORK

We have presented a variant of SSIA, which we call TS-SSIA, that works better for allocation of tasks that have time constraints. We have compared its performance to other single item auction algorithms. In the 2D grid case-study, TS-SSIA produced better solutions than the other single-item auctions in every case. Going forward, we plan to improve the TS-SSIA algorithm to add the ability to adjust the schedule of tasks already allocated when this can result in the allocation of a new task. We also plan to provide a formal theoretical analysis of the algorithm, and extend the experimental work to additional case-studies.

5. REFERENCES

[1] M. B. Dias. *TraderBots: A Market-Based Approach for Resource, Role, and Task Allocation in Multirobot Coordination*. PhD thesis, Carnegie-Mellon Univ., 2004.

[2] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain. Auction-based multi-robot routing. In *Robotics: Science and Systems*, pages 343–350, Cambridge, USA, June 2005.

[3] J. Melvin, P. Keskinocak, S. Koenig, C. A. Tovey, and B. Y. Ozkaya. Multi-robot routing with rewards and disjoint time windows. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 2332–2337, 2007.

[4] M. Nanjanath and M. Gini. Repeated auctions for robust task execution by a robot team. *Robotics and Autonomous Systems*, 58(7):900–909, July 2010.

[5] S. Ponda, J. Redding, H.-L. Choi, J. How, M. Vavrina, and J. Vian. Decentralized planning for complex missions with dynamic communication constraints. In *American Control Conference*, 2010.

Real-World Testing of a Multi-Robot Team (Extended Abstract)

Paul Scerri, Prasanna Velagapudi, Balajee Kannan, Abhinav Valada, Christopher Tomaszewski, John Dolan, Adrian Scerri, Kumar Shaurya Shankar, Luis Bill and George Kantor
The Robotics Institute
Carnegie Mellon University
pscerri@cs.cmu.edu

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Experimentation

Keywords

Multi-robot systems, Autonomous surface vehicle, Human-robot/agent interaction, Flood disaster mitigation, Autonomous sampling

Introduction

Multi-robot systems (MRS) have received a great deal of attention recently due to their potential to address complex distributed tasks such as environmental monitoring, search and rescue, agriculture, and security[3, 4, 5, 1, 2]. One specific type of multi-robot system that has significant near term promise is fleets of autonomous watercraft for applications such as flood response, water monitoring and bathymetry. Small watercraft are an attractive option for real world multi-robot systems because some of the most critical robotic problems are minimized on water - movement is relatively simple and dangers are relatively low.

In this work, we have addressed many engineering issues behind developing teams of Cooperative Robotic Watercraft (CRW). Deploying fleets of boats at remote locations helped clarify assumptions, change priorities and expose new issues for the community as well as help close the gap between the identified challenges and real-world deployment of such systems.

The success of our approach has been validated through field trials, including a four-day test at an irrigation pond in Maryland, a six-week expedition to various locations in the Philippines, a two day trip to a highly polluted canal in New York and hundreds of boat hours around Pittsburgh.

Design

Hardware.

We chose an airboat design (Figure 1), where the propulsion provided by a fan placed above water, for our watercraft

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

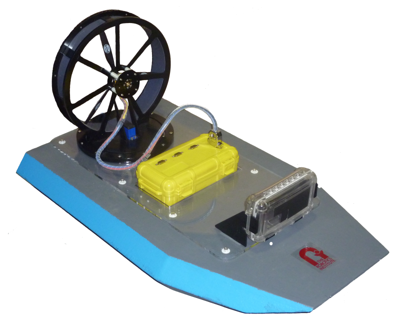


Figure 1: A complete airboat.

platform for two important reasons. First, keeping the propeller above water is advantageous where the water might be shallow, e.g., in flooded environments or in ecologically interesting areas like reefs or estuaries. Second, the above water fan can be simply encased in a wire mesh for safety, making the boats safe for autonomous operation even around curious children.

Figure 2 shows the basic components of the airboat. A boat is approximately 70cm long and weighs about 4.4kg without batteries. A NiMh battery that weighs 1.5kg allows the boat to drive continuously at approximately 10km/h for a period of two hours. The size and weight of the boat were chosen to suit urban flood conditions, where safety and maneuverability are key requirements.

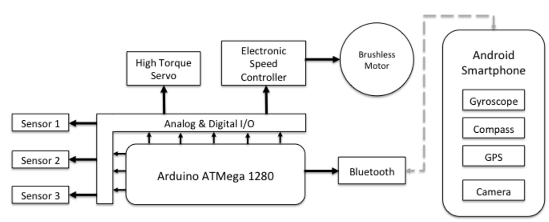


Figure 2: Hardware functional diagram.

Electronics.

Rather than individually assembling a computing platform, a core design decision was to use a commercial smartphone to provide the computing, camera and communications for the boat. It is impractical to put together a similarly powerful, robust and tightly packaged custom computing platform at anywhere near the cost of a smartphone. Moreover, using a smartphone gives us access to multiple modes of communication, since most phones have WiFi, 3G and Bluetooth. We chose Android-based phones because of their relatively open and powerful development environment.

For communicating with sensors, motors and servos, we used an Arduino Mega, a relatively low-cost microcontroller board that provides a fast, flexible array of digital and analog I/O for controlling the fan shroud, gyros, and external sensor modules. The Arduino and smart phone communicate via Bluetooth, which works extremely well over the short distance between the phone and Arduino. The servo for actuating the fan, the fan itself and sensors are all connected directly to the Arduino which has a simple, high-level protocol to the phone.

External sensors are plugged directly into the Arduino, using either digital or analog channels, depending on the sensor. The entire electronics assembly is encased in two waterproof boxes.

Software.

The control software builds on the Robot Operating System (ROS), which provides a flexible *publish-subscribe* architecture with extensive built in debugging capabilities and a manageable development path. Layers of functionality separate general modules from application specific modules. An end user interface provides a single operator with an overview of the state of the boats and provides high and low level commands for interacting with them.

Field Trials

In September 2011, three undergraduate students took five boats to the Philippines. They were joined by observers from the University of the Philippines and from local aid organizations. Primary testing lasted for one week, after which two of the students returned home leaving one (non-CS) undergraduate student to continue testing. Testing was performed in several locations including Laguna de Bay, Taal volcano, a village during flooding in the aftermath of twin typhoons and a fish farm. A key aim was to have all five boats in the water at the same time, under the control of the same operator. This was achieved a number of times. In total there were more than 15 tests in seven different locations. The boats were predominantly used for water sampling, but were also briefly evaluated in the aftermath of a typhoon. The testing resulted in more than 100 boat hours in the water, tens of kilometers covered and hundreds of thousands of data points. While initial testing was slow, frustrating and involved a lot more time with the boats out of the water than in, by the end the process and boats were sufficiently usable and robust that one non-computer science undergraduate student and local Filipinos with no formal education were able to deploy and use the boats. In fact, one of the biggest surprises was the comfort of local Filipino people with the technology and the speed at which they were able to familiarize themselves with it. By far the biggest problem encountered was with wireless communication, with the real-world details of various wireless technologies, particularly 3G, causing difficulties.

Figure 3 shows the path taken by a boat at the fish farm, an interesting environment because of the complexity of the water and the need to keep the water healthy. Figure 4 shows a plot of the water temperature in the lake inside Taal volcano immediately before (left) and after (right) rain. This lake is important because a recent unexpected, rapid and significant rise in temperature caused $1.3M in losses to fish farming in the lake. The plot shows considerable variation in the temperature and significant differences due to the rain.

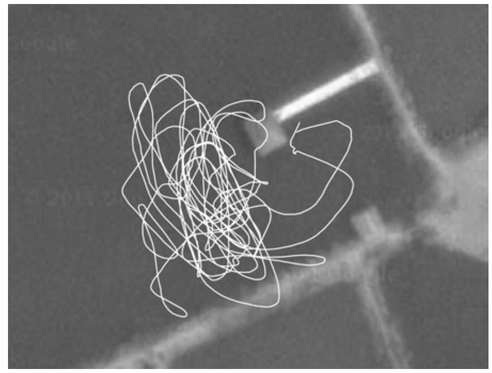


Figure 3: Airboat trajectory of a single airboat operating in a fish farming pong in Dagupan.

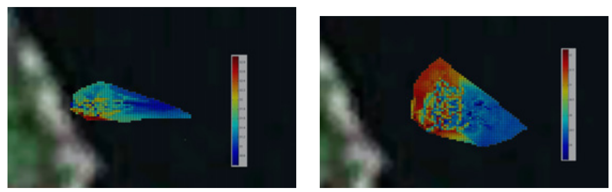


Figure 4: Plots of temperature in Taal Lake before (left) and after (right) a tropical rain storm.

Figure 5: Our six-week deployment in the Philippines demonstrated an ability to deploy five airboats simultaneously in remote locations with a control interface simple enough to be used by a child.

REFERENCES

[1] H. Endres, W. Feiten, and G. Lawitzky. Field test of a navigation system: Autonomous cleaning in supermarkets. In *ICRA*, 1998.

[2] S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, and M. Kurita. Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, 105(2):160–171, 2010.

[3] K.H. Low, G. Podnar, S. Stancliff, J.M. Dolan, and A. Elfes. Robot boats as a mobile aquatic sensor network. In *Workshop on Sensor Networks for Earth and Space Science Applications at IPSN*, 2009.

[4] G.A. Oosthuizen, A. Al Shaalane, et al. Evaluation of existing robot technologies for deep level mining applications. In *ISEM 2011*, 2011.

[5] Charles L. Ortiz, Regis Vincent, and Benoit Morisset. Task inference and distributed task management in centibots robotic systems. In *AAMAS*, 2005.

Online Planning for Large MDPs with MAXQ Decomposition

(Extended Abstract)

Aijun Bai
School of Computer Science
Univ. of Sci. & Tech. of China
Hefei, Anhui 230027 China
baj@mail.ustc.edu.cn

Feng Wu
School of Computer Science
Univ. of Sci. & Tech. of China
Hefei, Anhui 230027 China
wufeng@mail.ustc.edu.cn

Xiaoping Chen
School of Computer Science
Univ. of Sci. & Tech. of China
Hefei, Anhui 230027 China
xpchen@ustc.edu.cn

ABSTRACT

Markov decision processes (MDPs) provide an expressive framework for planning in stochastic domains. However, exactly solving a large MDP is often intractable due to the curse of dimensionality. Online algorithms help overcome the high computational complexity by avoiding computing a policy for each possible state. Hierarchical decomposition is another promising way to help scale MDP algorithms up to large domains by exploiting their underlying structure. In this paper, we present an effort on combining the benefits of a general hierarchical structure based on MAXQ value function decomposition with the power of heuristic and approximate techniques for developing an online planning framework, called MAXQ-OP. The proposed framework provides a principled approach for programming autonomous agents in a large stochastic domain. We have been conducting a long-term case-study with the RoboCup soccer simulation 2D domain, which is extremely larger than domains usually studied in literature, as the major benchmark to this research. The case-study showed that the agents developed with this framework and the related techniques reached outstanding performances, showing its high scalability to very large domains.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

Keywords

MDP, Online Planning, MAXQ, RoboCup 2D

1. MAIN RESULTS

Markov decision processes (MDPs) have been proved to be a useful model for planning under uncertainty. In general, online planning interleaves planning with execution and chooses the best action for the current step. Given the MAXQ [2] hierarchy of an MDP, the main procedure of MAXQ-OP evaluates each subtask by forward search to compute the recursive value functions $V^*(i,s)$ and $Q^*(i,s,a)$ online. This involves a complete search of all paths through the MAXQ hierarchy starting from the root task M_0 and ending with some primitive subtasks at the leaf nodes. After the search process, the best action $a \in A_0$ is selected for the root task M_0 based on the recursive Q function. Meanwhile, the true primitive action $a_p \in A$ that should be performed first can also be determined. This action a_p will be executed to the environment, leading to a transition of the system state. Then, the planning procedure starts over to select the best action for the next step.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1.1 Task Evaluation over Hierarchy

The search starts with the root task M_i and the current state s. Then, the node of the current state s is expanded by trying each possible subtask of M_i. This involves a recursive evaluation of the subtasks and the subtask with the highest value is selected. The evaluation of a subtask requires the computation of the value function for its children and the completion function. The value function can be computed recursively. Therefore, the key challenge is to calculate the completion function.

Intuitively, the completion function represents the optimal value of fulfilling the task M_i after executing a subtask M_a first. Obviously, computing the optimal policy is equivalent to solving the entire problem. In principle, we can exhaustively expand the search tree and enumerate all possible state-action sequences starting with s, a and ending with s' to identify the optimal path. However, this may be inapplicable for large domains. In Section 1.2, we will present a more efficient way to approximate the completion function.

1.2 Completion Function Approximation

To compute the optimal completion function, $C^{\pi^*}(i,s,a)$, the agent must know the optimal policy π^*, which is unavailable in the online settings. Due to the time constraint, it is intractable to find the optimal policy online since the search process is equivalent to solve the entire problem. When applying MAXQ-OP to large problems, approximation should be made to compute the completion function for each subtask. We assume that each subtask M_i will terminate at its terminal states in G_i with a prior distribution of D_i. In principle, D_i can be any probability distribution associated with each subtask. It can also take into consideration of the task parameters. For simplicity, we take uniform distribution as an example, then $C^{\pi}(i,s,a)$ can be approximated

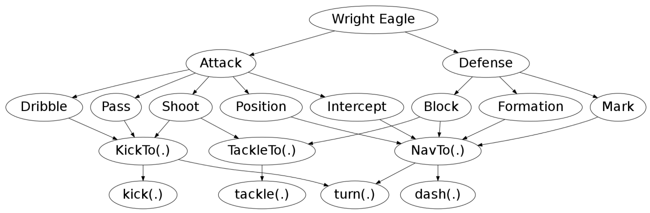


Figure 1: MAXQ task graph for Wright Eagle

Table 1: History results of Wright Eagle

Competitions	Games	Goals	Win	Draw	Lost
RoboCup 2005	19	84 : 16	15	2	2
RoboCup 2006	14	57 : 6	12	2	0
RoboCup 2007	14	125 : 9	11	1	2
RoboCup 2008	16	74 : 18	13	1	2
RoboCup 2009	14	81 : 17	12	0	2
RoboCup 2010	13	123 : 7	11	0	2
RoboCup 2011	12	151 : 3	12	0	0

as:

$$C^{\pi}(i,s,a) \approx \frac{1}{|\tilde{G}_a|}\sum_{s' \in \tilde{G}_a} V^{\pi}(i,s'), \qquad (1)$$

where $\tilde{G}_a \subset G_a$ is a set of sampled states drawn from uniform distribution D_a. A recursive procedure is proposed to estimate the completion function. In practice, the prior distribution $P(s', N|s, a)$–a key distribution when computing the completion function, can be improved by considering the domain knowledge.

1.3 Heuristic Search in Action Space

For some domains with large action space, it may be very time-consuming to enumerate all possible actions (subtasks) exhaustively. Hence it is necessary to introduce some heuristic techniques (including prune strategies) to speed up the search. Intuitively, there is no need to evaluate those actions that are not likely to be better. Different heuristic techniques can be chosen for different subtasks, such as hill-climbing, gradient ascent, branch and bound, etc.

2. CASE STUDY: ROBOCUP 2D

It is our long-term effort to apply the MAXQ-OP framework to the RoboCup soccer simulation 2D domain–a very large testbed for the research of decision-theoretic planning [3]. In this section, we present a case-study of this domain and evaluate the performance of MAXQ-OP based on the general competition results with several high-quality teams in the RoboCup simulation 2D community. The goal is to test the scalability of MAXQ-OP and shows that it can solve large real-world problems that are previously intractable.

2.1 Solution with MAXQ-OP

The graphical representation of the MAXQ hierarchical structure of our team Wright Eagle[1] is shown in Figure 1, where a parenthesis after a subtask's name indicates this subtask will take parameters. It is worth noting that state abstractions are implicitly introduced by this hierarchy. To deal with the large action space, heuristic methods are critical when applying MAXQ-OP. Table 1 summarizes the general performance of our team with MAXQ-OP in the RoboCup completion of past 7 years.[2]

There are multiple factors contributing to the general performance of a RoboCup 2D team. It is our observation that our team benefits greatly from the abstraction we made for the actions and states. The key advantage of MAXQ-OP in our team is to provide a formal framework for conducting the search process over a task hierarchy. Therefore, the team can search for a strategy-level solution automatically online by given the pre-defined task hierarchy. To the best of our knowledge, most of the current RoboCup teams develop their team based on hand-coded rules and behaviors. Overall, the goal of this case-study is twofold: 1) it demonstrates the scalability and efficiency of MAXQ-OP for solving a large real-world application such as RoboCup soccer simulation 2D; 2) it presents a decision-theoretic solution for developing a RoboCup soccer team, which is more general and easy for programming high-level strategies.

[1]Team website: `http://www.wrighteagle.org/2d`

[2]Logfiles: `http://ssl.robocup-federation.org/ftp/2d/log/`

3. CONCLUSIONS

This paper presents MAXQ-OP–a novel online planning algorithm that benefits from both the advantage of hierarchical decomposition and the power of heuristics. A key contribution of this work is to approximate the prior distribution when computing the completion function. By given such prior distributions, MAXQ-OP can evaluate the root task online without actually computing the sub-policy for each subtask. Similar to our work, Barry et al. proposed an *offline* algorithm called DetH* to solve large MDPs hierarchically by assuming that the transitions between macro-states are totally deterministic [1]. In contrast, we assume a prior distribution over the terminal states of each subtask, which is more realistic. The case study shows that MAXQ-OP is able to solve a very large problem such as the RoboCup 2D that are previously intractable in the literature of the decision-theoretic planning. This demonstrates the soundness and stability of MAXQ-OP for solving large MDPs with the pre-defined task hierarchy. In the future, we plan to theoretically analyze MAXQ-OP with different task priors and try to generate these priors automatically.

4. ACKNOWLEDGMENTS

This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150 and the Natural Science Foundations of China under grant 60745002 and 61175057.

5. REFERENCES

[1] J. Barry, L. Kaelbling, and T. Lozano-Perez. Deth*: Approximate hierarchical solution of large markov decision processes. In *International Joint Conference on Artificial Intelligence*, pages 1928–1935, 2011.

[2] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Machine Learning Research*, 13(1):63, May 1999.

[3] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12(2-3):233–250, 1998.

Enabling Robots to Find and Fetch Objects by Querying the Web (Extended Abstract)

Thomas Kollar, Mehdi Samadi, Manuela Veloso
School of Computer Science, Carnegie Mellon University
tkollar@cmu.edu, {msamadi, mmv}@cs.cmu.edu

ABSTRACT

This paper describes an algorithm that enables a mobile robot to find an arbitrary object and take it to a destination location. Previous approaches have been able to search for a fixed set of objects. In contrast, our approach is able to dynamically construct a cost function to find any object by querying the web. The performance of our approach has been evaluated in a realistic simulator, and has been demonstrated on a companion robot, which can successfully execute plans such as finding a "coffee" and taking it to a destination location like, "Gates-Hillman Center, Room 7002."

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Robotics

General Terms

Algorithms

Keywords

Robotics, Object Search, Web, Cyber-physical systems

1. INTRODUCTION

Our aim is to make robots that can interact with people in a natural and intuitive way. Toward this end, we look at a problem domain where people ask a robot to find an object and then have the robot deliver the object to a specified destination. This is a challenging problem since people will specify object names using open-ended natural language, leading to many distinct queries. Humans address the variability in the query by using common-sense knowledge. For example, if a person is asked to find and deliver the object "coffee," a person easily knows that "coffee" is likely to be found in the "kitchen." Robots, however, generally have limited access to such knowledge.

To address this limitation, we enable robots to access the web when they are missing task-related knowledge. In the object-finding domain this knowledge relates objects to locations; for example, assuming the robot has no prior experience with an object, such as "coffee," the robot will search the web. Using the results of the web search, the robot is able to predict that a "coffee" is generally found in a "kitchen," instead of in a "printer room." Our approach dynamically incorporates these predictions into a cost function, which minimizes the distance the robot travels and the number of interactions it has with people, while at the same time maximizing the the probability that an object will be found in each of the visited locations. The inferred multi-step plan consists of a sequence of locations to visit and questions to ask people.

We evaluate the robot's performance by executing plans to find 80 objects in a realistic simulator and have demonstrated our approach on a companion robot. This work builds off of OpenEval [4], which is able to evaluate the probability of arbitrary facts (predicates) by querying the web.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. APPROACH

To command a robot to find and fetch objects, people specify a query object (e.g., "coffee") and a place to which the object should be taken (e.g., "Gates-Hillman, room 7002"); the robot infers a plan to find the object and take it to the destination. Our robot overcomes its limitations in object detection and manipulation by asking humans for help [3].

We formulate the problem of finding and fetching objects as maximizing a utility function. If O is an object name (e.g., coffee), then the problem can be formulated as finding the plan that maximizes the utility function U:

$$\arg\max_{\text{plan}} U(\text{plan}|O) \tag{1}$$

The utility of a plan is broken down into the utility of each element: $U(\text{plan}_i)$. Each element of the plan ($plan_i$) visits a location and asks for an object from a person.

$$U(\text{plan}|O) = \sum_{i=1}^{N} U_i(\text{plan}_i|O) \tag{2}$$

The utility U_i consists of three components:

- The probability of the plan. We approximate this term as the probability of a location in the environment (e.g., "kitchen") given a query object (e.g., "coffee"). This probability is high when the location is likely to contain the object.
- A reward for traveling as little as possible. This term is computed by subtracting the distance traveled from the maximum distance the robot could travel for the current component of the plan.
- A reward for the number of interactions that the robot has with a person. This term is computed by subtracting the number of interactions required to search a location from the maximum number of interactions the

robot can have for the current component of the plan.

The first component (the probability of the plan) requires the system to compute the probability of finding an object in a location. This term connects a query object (e.g., "coffee") to a location type in the environment (e.g., "kitchen"). Connecting a query word for an object to a place is challenging because there are thousands of different object names people can use.

2.1 Querying the Web

To evaluate the probability of a plan, we have developed a general predicate evaluator called OpenEval, which returns a probability distribution over instances of predicates [4]. Unlike other approaches which read the web [1], OpenEval evaluates the validity of existing predicates and returns results immediately. For this paper, OpenEval has been trained on a single predicate, *locationHasObject(L, O)*, which is true only when location L contains object O:

$$p(L = \text{kitchen} | O = \text{coffee}) \\ \triangleq p(\text{locationHasObject(kitchen, coffee)}) \quad (3)$$

At training time a small number of predicate instances, such as *locationHasObject(kitchen, refrigerator)*, are provided. A web search for {"kitchen", "refrigerator"} returns a set of documents (web-pages), from which text snippets are extracted. These text snippets are treated as ground truth examples of the predicate instance, and a SVM classifier is trained to discriminate between different location types. Because Equation 3 is multinomial over locations, OpenEval only needs positive training examples of location / object pairs. At test time, OpenEval will evaluate the probability of a new predicate instance, such as *locationHasObject(kitchen, coffee)* by converting the input relation instance to a search query, such as {"coffee" "kitchen"}, and downloading the highest ranked web pages. This set of documents (web-pages) is classified into one of the location types; a probability is computed according to the proportion of web-pages that is classified as being in a "kitchen" given the object "coffee."

2.2 Optimization

The system takes as input the name of an object (e.g., "coffee"), a destination (e.g., "Gates-Hillman, room 7002"), the current location of robot, a set of example predicates that are used to train OpenEval, and a map that includes the type of each room (e.g., "office," "kitchen," "bathroom," or "other"). A cost function is then instantiated given the query object (Equation 1), and our approach uses beam search to find a sequence of candidate locations that maximizes the utility. The robot then executes the corresponding plan, recomputing the plan after visiting each location.

3. EVALUATION

We have evaluated our approach by showing that OpenEval is able to correctly categorize the location of novel objects and locations. Table 1 shows the probability of different test objects for each of the four location types. The only erroneous prediction is that bathrooms likely contain cups.

To show that our approach is able to search for objects, we have simulated a semantic map of 305 spaces over three floors of an office building. We have evaluated the effectiveness of our approach against two baseline approaches on 80 commands for 40 different object types that were not a part of the training set for OpenEval. The first baseline (frontier) searches nearby locations, using the rewards from Section 2, but not the probability returned by searching the web. The second baseline (ESP) provides a baseline similar to previous work [2]. It uses all the terms in Section 2, but instead of searching the web for relevant documents, it searches data collected from ESP [5].

Object	Location Types			
	Bathroom	Printer Room	Kitchen	Office
laptop	0.07	0.23	0.13	**0.56**
papers	0.1	**0.57**	0.12	0.22
cup	**0.36**	0.16	0.29	0.18
coffee	0.22	0.21	**0.3**	0.28

Table 1: The probability that OpenEval assigns to different locations given each novel object type. The location type which has the maximum probability is shown as bold.

The frontier approach finds objects in an average of 45.4 visited locations (standard error of 6.8), ESP finds objects in an average of 31.56 visited locations (standard error of 5.64) and our approach finds objects in an average of 19.21 visited locations (standard error of 4.62). This indicates a clear downward trend in the number of steps required to find query objects when using the web to retrieve knowledge about the physical environment. The number of visited locations is relatively high because a few objects require searching many (or all) locations in the environment.

Sometimes our approach will retrieve a reasonable location for the object, but the test environment does not contain the object in that location. For example, our approach searches bathrooms for "cleaning liquid" first because it can often be found in a home bathrooms, even though office bathrooms in our environment do not contain this object. In addition, sometimes our approach will infer the correct location, but it may still take several steps before the object is found. For example, since not every "office" has a "laptop," "jacket," or "hat" (e.g., often people have desktops, or don't wear hats), our approach will visit a few locations before finding the object. Finally, we have demonstrated our approach on our companion robot (CoBot), showing that it is successfully able to find a "coffee" in the "kitchen" and take it to "Gates-Hillman, room 7002."

4. REFERENCES

[1] A. Carlson, J. Betteridge, R. C. Wang, E. R. H. Jr., and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of WSDM*, 2010.

[2] T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *Proceedings of ICRA*, pages 4116–4121, 2009.

[3] S. Rosenthal, J. Biswas, and M. Veloso. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proc. of AAMAS*, pages 915–922, 2010.

[4] M. Samadi, M. Veloso, and M. Blum. Evaluating correctness of propositions using the web. In *Workshop on Learning by Reading and its Applications in Intelligent Question-Answering*, IJCAI, 2011.

[5] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of SIGCHI*, pages 319–326, 2004.

Configurable Human-Robot Interaction for Multi-Robot Manipulation Tasks

(Extended Abstract)

Bennie Lewis
Department of EECS
University of Central Florida
Orlando, FL 32816-2362
bennielewis@knights.ucf.edu

Gita Sukthankar
Department of EECS
University of Central Florida
Orlando, FL 32816-2362
gitars@eecs.ucf.edu

ABSTRACT

Multi-robot manipulation tasks can be complicated, due to the need for tight temporal coupling between the robots. However, this is an ideal scenario for human-agent-robot teams, since performing all of the manipulation aspects of the task autonomously is not feasible without additional sensors. To ameliorate this problem, we present a paradigm for allowing subjects to configure a user interface for multi-robot manipulation tasks; using a macro acquisition system for learning combined manipulation/driving tasks. Learning takes place within this social setting; the human demonstrates the task to the single robot, but the robot uses an internal teamwork model to modify the macro to account for the actions of the second robot during execution. This allows the same macro to be useful in a variety of cooperative situations. In this paper, we show that our system is highly effective at empowering human-agent-robot teams within a household multi-robot manipulation setting and is rated favorably over a non-configurable user interface by a significant portion of the users.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Operator interfaces

General Terms

Algorithms

Keywords

human-robot interaction, multi-robot manipulation, programming by example

1. INTRODUCTION

Human-agent-robot teams [1] fill an important niche in robotics since they can accomplish tasks that robots cannot complete autonomously, forming a team unit that is greater than the sum of the parts. Ideally the human users focus on the difficult cognitive and perceptual tasks, the robots manage the planning and execution of repetitive physical tasks, while the agents handle the most cumbersome information processing tasks. At the core of designing an effective social system that includes human, agent, and robot teammates is the question of communication between the biological and synthetic entities—how to create a user interface that empowers rather than hinders teamwork and social learning?

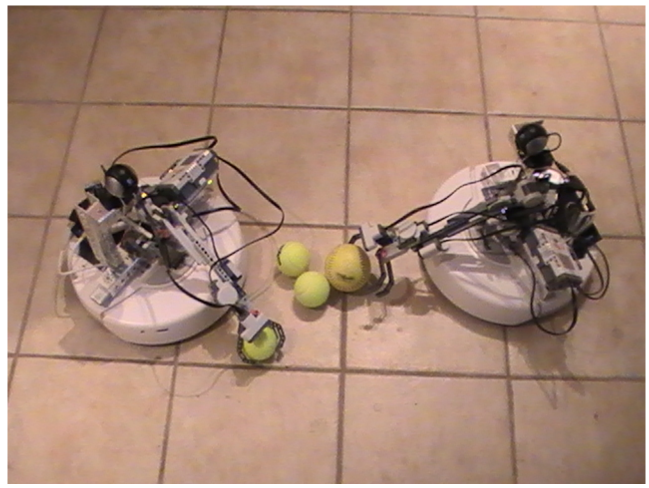


Figure 1: Two HU-IE robots cooperating together to clear the environment of objects and deposit them in the goal location.

Here we focus on the problem of multi-robot manipulation; the human user guides a team of robots to lift and clear clutter in a household environment. Since some of the objects are too large to be raised by a single robot, the robots must work together in tight temporal coordination to lift and transport the clutter to the goal area. Coordination failure leads to dropped objects and slow task completion times. The users must also effectively control the multiple degrees of freedom that the robot offers (wheelbase, arm, and claw).

2. USER INTERFACE

The user views the environment and interacts with the HU-IE robot team through our configurable user interface (IAI: Intelligent Agent Interface). A rudimentary agent is embedded within the user interface to support teamwork by managing information propagation between the team mem-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Figure 2: State representation of a recorded macro

bers; it governs the information that gets sent to the robots and displayed on the user interface. Additionally it contains a macro acquisition system that allows the user to identify four key subtasks which are abstracted and used to create robot behaviors which the user can deploy during task execution. All commands to the robots are issued through an Xbox 360 gamepad, using a button to switch between robots.

3. MACRO ACQUISITION

During the macro acquisition phase, the robot's state space trajectory is recorded, paying special attention to the initial and final states of the trajectory. The state includes the following features in absolute coordinates: drive start/end position, arm start/end, claw open/closed. Additionally, the status of all of the key sensor systems (cliff, wall, and bumper sensors) is logged. The agent also notes the current location of known movable objects in the environment and whether the user is teleoperating the second robot. The state space trajectory is then used to create an abstract workflow of the task which can be combined with the teamwork model and the path planner to generalize to new situations. To build the workflow, the state space trajectory is separated into drive, arm, and claw segments. Adjacent drive and arm segments are merged to form one long segment. The terminal position of the robot is retained in both absolute coordinates and also the relative position to the nearest object or robot.

After the macro acquisition phase, there is an acceptance phase during which the operator is given a chance to verify the macros' performance. When the human operator is satisfied that the macro was performed correctly then the macro is accepted and mapped to one of the Xbox 360 buttons. During the acceptance phase, the macro is evaluated in multiple locations on the map and with the HU-IE robot arm at different angles.

If the macro representation was not accepted by the human operator, the system attempts to modify the macro using a set of taskwork rules. For instance, during the initial phase, it is assumed that the terminal positions are of key importance and that the robot should use the path planner to return to the same absolute position. In the second demonstration, the system used the recorded sensor date to identify the most salient object located near the terminal position and return the robot to that area. If an object is dropped during the acceptance phase, it is assumed that the drop is the principal reason for the macro non-acceptance and the macro is repeated using the same abstraction but with minor modifications to its positioning relative to the object using the ultrasonic sensor. For simplicity of user interaction, macro acquisition is done by teleoperating a single robot but during actual task execution many of the macros are actually executed in mirror mode, using the preprogrammed teamwork model. One of the most common macros developed by both expert and novice users was a macro for driving the robot to the goal.

4. RESULTS

The users were asked to clear objects from a cluttered household environment and transport them to a goal area using two robots guided by the configurable user interface. We evaluated the performance and quality of the IAI system Macro Mode on a variety of measures, including usability of the macros, speed of task completion, number of object drops, and user satisfaction. Two indoor scenarios, a training scenario, and One macro recording phase were employed in our user study. The user was asked to execute each of the scenarios using our Intelligent Agent Interface Macro Mode.

The macros created by users varied in length and complexity, with a general trend that game skill correlated with shorter macros and longer periods of user teleoperation. This can be contrasted with the pattern of novice macro usage that shows a heavier reliance on macros. Overall, we found it encouraging that the configurable aspects of the user interface were more heavily used by novice users.

From observation, we noted that the users created macros to help them with parts of the task that they struggled on during training; for instance, users who experienced more failed pickups would often focus on creating a good object pick up macro. In a post hoc comparison to users from a previous study who used a non-configurable version of the same user interface, macros appeared to confer a slight time advantage. The most significant results were in the user rankings of the interface which enthusiastically (70%) preferred the configurable user interface; overall, the interface scored high ratings in the post-questionnaire user ratings.

5. ACKNOWLEDGMENTS

This research was supported in part by NSF award IIS-0845159.

6. CONCLUSION AND FUTURE WORK

In this paper we demonstrate a macro acquisition system for learning autonomous robot behaviors by example; by separating taskwork and teamwork, we can generalize single robot macros to multi-robot macros. We plan to extend the teamwork model in the future by having the system learn user-specific teamwork preferences separately through demonstrations on a non-manipulation task. Users expressed a significant preference for the configurable autonomy of macros over the built-in autonomous functions, and gave the user interface high overall ratings. Adding a configurable user interface to a human-agent-robot team empowers the human operator to structure his/her user experience by expressing task-specific preferences for the amount of interdependence vs. autonomy between human and robot. This is consistent with the coactive design model for human-agent-robot systems.

7. REFERENCES

[1] P. Scerri, L. Johnson, D. Pynadath, P. Rosenbloom, N. Schurr, M. Si, and M. Tambe. Getting robots, agents, and people to cooperate: An initial report. In *AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*, 2003.

Evaluating POMDP Rewards for Active Perception (Extended Abstract)

Adam Eck and Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska-Lincoln
256 Avery Hall, Lincoln, NE, 68588, USA
+1-402-472-4257
{aeck, lksoh}@cse.unl.edu

ABSTRACT
One popular approach to active perception is using POMDPs to maximize rewards received for sensing actions towards task accomplishment and/or continually refining the agent's knowledge. Multiple types of reward functions have been proposed to achieve these goals: (1) state-based rewards which minimize sensing costs and maximize task rewards, (2) belief-based rewards which maximize belief state improvement, and (3) hybrid rewards combining the other two types. However, little attention has been paid to understanding the differences between these function types and their impact on agent sensing and task performance. In this paper, we begin to address this deficiency by providing (1) an intuitive comparison of the strengths and weaknesses of the various function types, and (2) an empirical evaluation of our comparison in a simulated active perception environment.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multiagent systems*

General Terms
Performance, Design, Experimentation

Keywords
Active Perception, POMDP, Reward Functions, Sensing

1. INTRODUCTION
Recently, one application of intelligent agents growing in popularity is intelligent information gathering. Here, developers commonly model the agent's reasoning about sensing as an **active perception** problem (e.g., [9]), where the agent makes explicit decisions about sensing to maximize the quality and/or quantity of its gathered information. One popular approach to active perception is to make sequential decisions using the partially observable Markov decision process (POMDP) [5], e.g., in user preference elicitation [2, 3] and agent-based classification [4].

To illustrate, we consider a robotic mining simulation called *MineralMiner*, a testbed for sensing research similar to RockSample [8]. Here, an intelligent agent completes frequent mineral collection tasks with firm deadlines. To discover minerals (gold, silver, uranium), the agent performs sensing on various mines in the environment. The agent models its sensing at each mine with an active perception POMDP, where the states and observations represent the possible mineral types in the mine and the actions represent various sensing activities with different cost and accuracy, as well as drilling actions (which stop sensing) for each type of mineral. Of note, drilling for an incorrect mineral type destroys a mine. Thus, quality sensing is necessary for completing tasks.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4–8 June 2012, Valencia, Spain.

2. REWARD FUNCTION COMPARISON
Several types of reward functions for active perception POMDPs have been proposed in the literature. First, **state-based rewards** $R(s,a)$ (e.g., [3, 4]) follow the traditional design of reward functions in the POMDP literature [5], where rewards are the benefit or cost of actions in different states with respect to the accomplishment of tasks and environment impact. An agent handles its uncertainty about the hidden state of the environment by marginalizing expected rewards over beliefs about each state:

$$\sum_{s\in S} b(s)R(s,a) \quad (1)$$

We present $R(s,a)$ values for two state-based reward functions for MineralMiner in Table 1 (similar to [3]), where (1) Cost Sensing encodes the actual costs incurred by the agent for each action, and (2) Zero Cost Sensing focuses only on task rewards.

Alternatively, recently proposed **belief-based rewards** [1] break from tradition and consider only measures on the entire belief state of the agent, independent of individual states. For example, if the primary goal of sensing is to reduce the uncertainty in the agent's beliefs, the agent can use the entropy in its belief state as a measure of uncertainty, then maximize the negative of its expected entropy to minimize uncertainty:

$$-E[H(b^{a,o})] = E\left[\sum_{s\in S} b^{a,o}(s)\log_{|S|} b^{a,o}(s)\right] \quad (2)$$

Other belief-based reward functions accomplish similar goals including maximizing information gain or the expected top belief:

$$E[\max_{s\in S} b^{a,o}(s)] \quad (3)$$

The intuitive strengths and weaknesses of these functions include:

- State-based rewards *directly encode the costs of sensing activities*, allowing the agent to minimize sensing costs, whereas belief-based rewards ignore such information.
- Belief-based rewards *directly encode the benefits of sensing* (i.e., belief state improvement), whereas state-based rewards only implicitly consider this information through finding policies of actions that reach task accomplishment the fastest.
- State-based rewards *provide a natural stopping condition for sensing*: when the expected reward of using information exceeds further sensing costs, and thus are appropriate for task-based environments. Belief-based rewards, instead, *require*

Table 1: State-based Rewards for MineralMiner

Action	Cost Sensing		Zero Cost Sensing	
	Correct State	Incorrect State	Correct State	Incorrect State
Advanced (80% Accuracy)	-5	-5	0	0
Basic (50% Accuracy)	-2	-2	0	0
Wait (do nothing)	0	0	0	0
Drilling	100	-500	100	-500

an external stopping condition (e.g., stop when a confidence threshold is reached for the top belief: $b(s) \geq 0.85$).

- Belief-based rewards *optimize beliefs for continual sensing when it is unknown when information will be used*, whereas state-based rewards might be inappropriate for such environments [1] due to lacking task rewards to guide sensing.

Finally, **hybrid rewards** consider both of the other types simultaneously in the form of a weighted function [1, 6], e.g.,

$$w \sum_{s \in S} b(s) R(s,a) - (1-w)\, E[H(b^{a,o})] \qquad (4)$$

where w weights the impact of the two reward types. Below, we use a hybrid of Cost Sensing (c.f., Table 1) with negative expected Entropy (Eq. 3) using three weights: $w = 0.25, 0.50, 0.75$.

Hybrid functions have the potential to merge the strengths of state- and belief-based rewards while mitigating their weaknesses:

- Hybrid rewards *add cost information for sensing activities to belief-based reward functions* to improve sensing.
- Hybrid rewards *incorporate belief state revision into state-based rewards* to speed up belief state convergence and promote faster task accomplishment.
- However, the weight between the two types of rewards must be properly tuned, which can be difficult to set *a priori.*

3. INVESTIGATION

We now provide results from an empirical investigation using MineralMiner to evaluate our intuitive comparison of the various active perception POMDP reward function types. To maximize rewards, we choose actions from limited depth policy trees created online [7] from the current belief state for the current mine. This approach (1) finds exact solutions with low computational cost due to the small POMDP size, and (2) allows us to compare how performance depends on policy depth $(1,2,3,4,5)$ due to the different properties of the functions. We used 600 mines/tasks to provide many opportunities for sensing and ran our experiments 30 times with different random seeds to minimize variance.

Figures 1-2 present: (1) the number of mines correctly identified/drilled, measuring sensing *effectiveness*, and (2) the task-based rewards earned by the agent (Cost Sensing, c.f., Table 1), measuring sensing *efficiency*. Due to space constraints, we highlight the key analyses from our results, confirming our intuitions:

- State-based functions almost always improved as policy depth increased. This is because myopically, state-based functions only minimize *immediate* costs, whereas non-myopically they minimize *total* costs through less sensing.
- Belief-based functions performed consistently for all policy depths due to always looking ahead to expected belief improvement, but achieved lower task rewards (higher costs) than state-based functions for longer policy depths $(3, 5)$.
- Hybrid rewards $(w = 0.75)$ performed the best due to combining cost-awareness (state-based rewards) and rapid belief improvements (belief-based rewards).

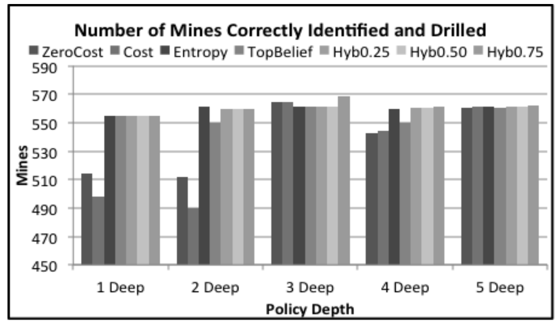


Figure 1: Number of Mines Correctly Identified and Drilled

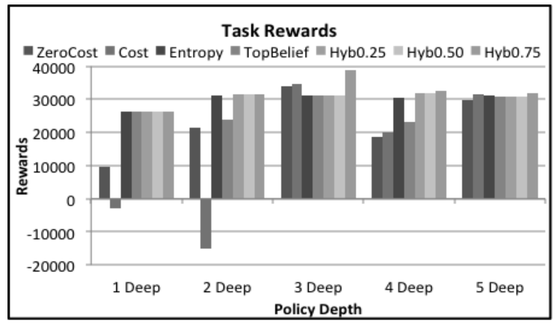


Figure 2: Cumulative Task-based Rewards
Note: Cost has negative task rewards due to very poor sensing

- Increasing policy depth from 3 to 4 worsened performance of all functions but at different rates. Thus, looking farther ahead is not always beneficial, which we intend to further study.

4. ACKNOWLEDGMENTS

This research was supported by a NSF Graduate Research Fellowship (grant DGE-054850) and was completed utilizing the Holland Computing Center of the University of Nebraska.

5. REFERENCES

[1] Araya-Lopez, M., Buffet, O., Thomas, V., and Charpillet, F. 2010. A POMDP extension with belief-dependent rewards. *Proc. of NIPS'10*.

[2] Boutilier, C. 2002. A POMDP formulation of preference elicitation problems. *Proc. of AAAI'02*. 239-246.

[3] Doshi, F. and Roy, N. 2008. The permutable POMDP: fast solutions to POMDPs for preference elicitation. *Proc. of AAMAS'08*. 493-500.

[4] Guo, A. 2003. Decision-theoretic active sensing for autonomous agents. *Proc. of AAMAS'03*. 1002-1003.

[5] Kaelbling, L.P., Littman, M.L., and Cassandra, A.R. 1998. Planning and acting in partially observable stochastic domains. *AI*. 101. 99-134.

[6] Mihaylova, L. et. al. 2002. Active sensing for robotics – a survey. *Proc. of NM&A'02*.

[7] Ross, S., Pineau, J., Paquet, S., and Chaib-draa, B. 2008. Online planning algorithms for POMDPs. *JAIR*. 32. 663-704.

[8] Smith, T. and Simmons, R. 2004. Heuristic search value iteration for POMDPs. *Proc.UAI'04*. 520–527.

[9] Weyns, D., Steegmans, E., and Holvoet, T. 2004. Towards active perception in situated multi-agent systems. *Applied Artificial Intelligence*. 18. 867-883.

Finding new consequences of an observation in a system of agents

(Extended Abstract)

Gauvain Bourgne
NII
Tokyo, Japan
bourgne@nii.ac.jp

Katsumi Inoue
NII
Tokyo, Japan
ki@nii.ac.jp

Nicolas Maudet
LIP6, UPMC
Paris, France
nicolas.maudet@lip6.fr

ABSTRACT

When a new observation is added to an existing logical theory, it is often necessary to compute new consequences of this observation together with the theory. This paper investigates whether this reasoning task can be performed incrementally in a distributed setting involving first-order theories. We propose a complete asynchronous algorithm for this non-trivial task, and illustrate it with a small example. As some produced consequences may not be new, we also propose a post-processing technique to remove them.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory

Keywords

Distributed Consequence Finding, Incremental Consequence Finding, Abduction

1. INTRODUCTION

This paper deals with the problem of finding all interesting new consequences which can be derived from some observations, given a full clausal theory. A consequence is deemed *interesting* if it respects a given language bias, and *new* if it is a consequence of the observations taken together with the theory but was not a consequence of the theory alone. Consequence finding is a general reasoning problem which lies at the heart of many AI applications. By focusing on computation of *new* consequences, one can perform efficient online computation of interesting consequences, an essential feature in dynamic contexts. On top of it, some problems specifically require to compute only new consequences, such as abduction by the principle of *inverse entailment*. Indeed, the set of abductive hypotheses is exactly the set of the negation of new consequences of the negated observation wrt the background theory. The computation of new interesting consequences is thus a very important challenge.

Of course, one can always compute new consequences by computing all consequences of the theory with and without the observations, and making the difference. But focusing only on new consequences is much more efficient.This can be especially interesting in contexts where information is accessed progressively. The research question we address here is the following: does it still hold in a distributed setting? There exist methods for computing new consequences in a distributed setting [1], but restricted to the propositional case. On the other hand, some recent work [2] allows computation of all interesting consequences of a distributed first-order theory, but it cannot focus on the new ones. We propose here a method that can deal with first order clausal theories while focusing on interesting new consequences.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. FINDING NEW CONSEQUENCES

A *clause* is a disjunction of literals. A clause C *subsumes* another clause D if there is a substitution θ such that $C\theta \subseteq D$. A *clausal theory* is a set of clauses, interpreted conjunctively. A *consequence* of Σ is a clause entailed by Σ. A clause C *belongs to* a *production field* $\mathcal{P} = \langle \mathcal{L} \rangle$, where $\mathcal{L}$ is a set of literals closed under instantiation, iff every literal in C belongs to $\mathcal{L}$. The set of all subsumption-minimal consequences of a theory Σ belonging to a production field $\mathcal{P}$ is called the *characteristic clauses* of Σ wrt $\mathcal{P}$[3], and denoted by $Carc(\Sigma, \mathcal{P})$. When some observations O are added to a clausal theory Σ, further consequences are derived due to this new information. Such new and interesting consequences are called *new characteristic clauses*. It is formally defined as the set of all subsumption-minimal consequences of $\Sigma \cup O$ belonging to $\mathcal{P}$ that are not consequences of Σ, and is denoted by $Newcarc(\Sigma, C, \mathcal{P})$.

We now consider a system of n_A agents $I = \{0, \ldots, n_A - 1\}$, each having a clausal theory Σ_i. Some of these agents make new observations (or acquire new information), represented as a set of clauses O_i. The objective is to determine all the new consequences of those new observations $O = \bigcup_{i \in I} O_i$ wrt the whole theory $\Sigma = \bigcup_{i \in I} \Sigma_i$ belonging to the shared *target production field* $\mathcal{P} = \langle \mathcal{L}_P \rangle$, that is, to compute $Newcarc(\bigcup_{i \in I} \Sigma_i, \bigcup_{i \in I} O_i, \langle \mathcal{L}_P \rangle)$. This specifies a distributed new consequence finding problem. We emphasize that agents do not share their theories, though for better efficiency, they share their respective languages.

Example 1. Consider a system of 4 agents, whose knowledge (theory and new observations) is defined as follows:

0:	$\Sigma_0 = \{f \vee g, a \vee g\}$,	$O_0 = \{e\}$.
1:	$\Sigma_1 = \{\neg a \vee b, \neg g \vee h\}$,	$O_1 = \emptyset$.
2:	$\Sigma_2 = \{\neg b \vee c \vee d, \neg d \vee \neg e\}$,	$O_2 = \emptyset$.
3:	$\Sigma_3 = \{\neg c \vee \neg f\}$,	$O_3 = \emptyset$.

The target production field is $\mathcal{P} = \langle\{h\}\rangle$ (*i.e.* $\mathcal{L}_P = \{h\}$).

3. DISTRIBUTED ALGORITHM

As in [4, 2], the main principle of our algorithm is to compute locally all relevant new consequences (and only those ones) and forward them to agents that can resolve them. Relevant consequences are either (i)new characteristics clauses of the problem, or (ii) *bridge* consequences, that is, consequences that can be used by one or more other agents to build such a new characteristic clause. In that latter case, they necessarily contains literals that can be resolved by other agents. We thus define, for each agent i the *output language* $\mathcal{L}_{i\rightarrow}$ as the set of all literals that (i) i might produce and (ii) can be resolved with a clause from another agent. Likewise, the *input language* $\mathcal{L}_{\rightarrow i}$ of an agent i is the set of all literals that (i) might be produced by another agent and (ii) can be resolved by some clause in its knowledge. Agents do not know each other theories, but they know each other input languages. Agents can focus their computations by using $\mathcal{L}_{\rightarrow i}$ and $\mathcal{L}_P$. Though a bridge consequence C could have literals that are not in these production fields, such literals can only appear if they were in a received clause. We thus define the reduction of C wrt some language $\mathcal{L}$ (reduc$(C, \mathcal{L})$) as the set of all literals that appear in C, but do not appear in positive nor negative form in $\mathcal{L}$. To achieve better efficiency, we apply a prune function to the received clauses, which checks them against $\Sigma_i \cup listCsq_i$, removing any subsumed clause.

Algorithm 1 Asynchronous algorithm

```
Global variables of agent i:
   Σ_i, O_i: initialized by problem, constant
   firstRun ← true
   listCsq_i ← ∅
// Whenever agent i receive sentCl from an agent
Receive(sentCl)
   if firstRun then sentCl ← sentCl ∪ O_i end if
   firstRun ← false
   // Computing new consequences
   prune(sentCl)
   pField ← ⟨L_P ∪ L_{i→} ∪ reduc(sentCl, L_{i→})⟩
   newCsq_i ← newcarc(Σ_i ∪ listCsq_i, sentCl, pField)
   listCsq_i ← listCsq_i ∪ newCsq_i
   // Sending relevant new consequences to neighbours
   for all agents j do
      toSend[j] ← ∅
      for all c ∈ newCsq_i do
         if c contains literals from L_{→j} then
            toSend[j] ← toSend[j] ∪ {c}
         end if
      end for
      if toSend[j] ≠ ∅ then
         send(j, toSend[j])
      end if
   end for
   //Check new consequences as output
   for all c ∈ newCsq_i do
      if belongs(c, L_P) then
         Output c
      end if
   end for
End
```

Example 2. (ex. 1 ctd.) Figure 1 illustrates the unfolding of the asynchronous algorithm. Each box represents an agent applying the *receive* procedure. Arrows between two boxes correspond to the communication of some clauses (given as label) by the first agent to the second one. The process is initiated by 0, who send e to 2 (as e is only in $\mathcal{L}_{\rightarrow 2}$). Then 2 computes the new consequences of e wrt to Σ_2 with production field $\langle\{h, \neg b, \neg e, c\}\rangle$ getting $\neg b \vee c$, which partially belongs to $\mathcal{L}_{\rightarrow 1}$ (through $\neg b$) and $\mathcal{L}_{\rightarrow 3}$ (c). It is thus sent to these two agents. Then 1 computes $Newcarc(\Sigma_1, \neg b \vee c, \langle\{h, \neg a, b, \neg g, c\}\rangle)$, and gets $\neg a \vee \neg c$, which is sent to 0 and 3, and so on, until h is sent as output and other branches are closed.

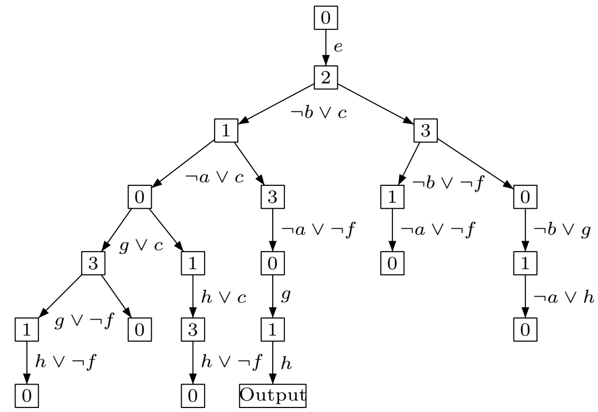


Figure 1: Asynchronous resolution of pb 1.

Termination is guaranteed for non-recursive theories. Otherwise, we need to enforce termination by fixing a limit to the number of resolve operations that can be applied to get a consequence. This algorithm is complete for multi-agent new consequence finding, meaning that it outputs all new consequences of $\bigcup_{i\in I} O_i$ wrt $\bigcup_{i\in I} \Sigma_i$ and $\langle\mathcal{L}_P\rangle$. It also ensures that each output is indeed a consequence of $\bigcup_{i\in I} O_i \cup \bigcup_{i\in I} \Sigma_i$. However, it might also be a consequence of the theory alone (and thus not strictly a new consequence). If our purpose is to incrementally compute all characteristic clauses, this is not a problem at all, but in some other cases, such as the computation of abductive hypothesis, we should only output *new* characteristic clauses. This can be ensured by computing, for each candidate consequence C, $N_C = Newcarc(\bigcup_{i\in I} \Sigma_i, \neg C, \langle\emptyset\rangle)$. If $N_C = \{\emptyset\}$, C is not new, otherwise, it can be kept as a solution.

4. CONCLUSION

We proposed in this paper a complete asynchronous algorithm to compute the new interesting consequences of some observations with respect to a full clausal theory distributed among a set of agents. Termination is guaranteed in cases where the centralized case also terminates, and soundness is ensured for incremental computations of consequences. Moreover some post processing was proposed to ensure soundness for computation of new consequences.

5. REFERENCES

[1] P. Adjiman, P. Chatalic, F. Goasdoué, M.-C. Rousset, and L. Simon. Distributed reasoning in a peer-to-peer setting: Application to the semantic web. *JAIR*, 25:269–314, 2006.

[2] G. Bourgne and K. Inoue. Partition-based consequence finding. In *Proc. of ICTAI'2011*, pages 641–648, 2011.

[3] K. Inoue. Linear resolution for consequence finding. *Art. Intel.*, 56(301–353), 1992.

[4] S. McIlraith and E. Amir. Theorem proving with structured theories. In *Proc. of IJCAI'01*, pages 624–634, 2001.

User-Centric Preference-Based Decision Making

(Extended Abstract)

Ingrid Nunes
PUC-Rio, LES
King's College London
ionunes@inf.puc-rio.br

Simon Miles, Michael Luck
King's College London
London, United Kingdom
{simon.miles,michael.luck}
@kcl.ac.uk

Carlos J.P. de Lucena
PUC-Rio, LES
Rio de Janeiro, Brazil
lucena@inf.puc-rio.br

ABSTRACT

The automation of user tasks by agents may involve decision making that must take into account user preferences. This paper introduces a decision making technique that reasons about preferences and priorities expressed in a high-level language in order to choose an option from the set of those available. Our technique includes principles from psychology, concerning the way in which humans make decisions. Our preference language is informed by a user study on preference expression, which is also used to evaluate our approach by comparing our results with those provided by a human expert. The evaluation indicates that our technique makes choices on behalf of the user with as good quality as made by the expert.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence —*Intelligent agents*

General Terms

Algorithms

Keywords

Agent Reasoning, Preferences, Decision Making, Autonomous Agents

1. INTRODUCTION

The automation of user tasks by agents may involve decision making that must take into account user preferences. Our vision is for agents to make decisions for users so that their choices match those of users themselves, given adequate time and knowledge. People do not act in isolation, and agents acting on their behalf should not do so either. Where the option chosen for one user may affect that of another (e.g., in deciding which hotel to stay at, we both prefer to stay at the same hotel), agents need to coordinate their actions. Such coordination between users reflects just one among the many interacting preferences that agents may need to consider. We argue that, by reflecting how users themselves decide, there is a *rationale* for choices that is *convincing* to users.

This paper describes the first step towards this vision. Before we can have decisions appropriate to multiple users, we must first have agent reasoning appropriate to a single user. However, many different approaches have been proposed for reasoning about preferences, but they address a restricted set of preference types, and therefore are not able to process preferences provided by users in many realistic scenarios. We propose a novel approach for reasoning about preferences. Specifically, the contributions of this work are (i) a high-level preference language, informed by user preferences in natural language; and (ii) an automated decision-making technique based on preferences and available options, and exploiting psychological research into the way in which humans make choices.

Our decision-maker takes as input a set of options over which a choice is made, and a set of preferences expressed in the language introduced below. It processes the preferences to select one option, in such a way that the choice, and the decision not to choose alternatives, can be justified by the preferences. The output is a partially ordered set, organised in four different levels: (i) the *chosen option*; (ii) *acceptable options* that are *close* to the chosen option, but not chosen; (iii) *eliminated options*, discarded because of a hard constraint; and (iv) *dominated options*. We apply heuristics used by humans, specifically the principles of *trade-off contrast* and *extremeness aversion* [3], so that decisions more closely mirror user choices if users are provided with sufficient time and knowledge. In outline, the steps of our technique are as follows.

Pre-processing. Options are analysed to extract the essential data. This includes how well option attributes meet the preferences, and how options compare with regard to individual attributes.

Explication. Some preferences include important *implicit* information, in addition to their literal meaning, and we extract it.

Elimination. Next, we eliminate options that do not meet strict constraints, or that are dominated in every regard by other options.

Selection. Finally, we make the choice itself. As the remaining options have both costs and benefits, we need to take account of all preferences that lead to a decision, such as the relative importance of attributes, plus heuristics, including the principles of *trade-off contrast* and *extremeness aversion* adopted by humans [3].

2. PREFERENCE LANGUAGE

Humans express preferences in many ways, and we wish to provide them with this natural expressivity when delegating decisions to a software agent. We propose a preference language derived from an existing user study on choosing a laptop, based on around 200 preference specifications [1]. Our language defines seven types of preference: *constraints* specify values that attributes must (not) have; *goals* specify which attributes should be minimised or maximised; *orders* specify preferences over attributes; *qualifying* preferences state how much an attribute value is wanted or needed; *rating* preferences specify which values are best or worst; *indifferences* specify the absence of preference between two attribute

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

values; and *don't care* preferences specify irrelevant attributes. In addition, preferences may apply only conditionally, where the *condition* is expressed in terms of attribute values, and *priorities* can be expressed either between attributes or preferences, so that an attribute or preference is given more weight in decision-making.

3. PRE-PROCESSING

Preferences can be *monadic* or *dyadic*, where the former evaluate a single referent, e.g. an apartment less than 2.5km away from the university is preferred, and the latter indicate a relation between two referents, e.g. lower price is better. First, we pre-process the options with regard to monadic and dyadic statements, thus building two models for use in later steps.

Performatives such as *need*, *require*, and *love* are widely adopted by users to express preferences over attributes, and so are included in our language. Similarly, users may rate preferences from best to worst. The rates and performatives used in monadic preferences are captured by a *Preference Satisfaction Model (PSM)*, which consists of a table indicating how options satisfy monadic preferences in terms of each attribute. The Options-Attribute Preference Model (OAPM) is a table that captures comparisons between two options, for individual attributes, showing which is better, or that no conclusion can be drawn from the provided preferences. The OAPM is based on preferences not used in the PSM, together with the PSM itself, processed separately in a specific order, and establishing a precedence: (i) order and indifference; (ii) goals; and (iii) PSM.

4. EXPLICATION AND ELIMINATION

Preferences always provide a literal meaning, but can also bring additional information to derive new preferences, referred to as *implicit preferences*. These never override information of explicitly provided preferences, but enable determination of whether an option is preferred to another with respect to a certain attribute, when this could not otherwise be concluded. We update the OAPM by considering these implicit preferences, such as considering that a higher value of an attribute (maximisation goal) is better then a lower one, if there is a preference that establishes a lower bound for this attribute, and both options satisfy this preference.

A typical approach adopted by users in making a choice is the stepwise elimination of options until there remains a set of acceptable options, ideally containing only one element, as in *elimination by aspects* [4]. In the elimination step, we discard two types of options: (i) dominated options; and (ii) options that do not satisfy hard constraints. The OAPM and the PSM are used to identify these options, respectively.

5. SELECTION

After *elimination*, we must choose an option from the acceptable set, i.e. available options without those eliminated. Humans commonly make use of heuristics [2], that demand different amounts of effort, typically choosing them by matching the effort required to the importance of the decision. Our approach does not aim to reproduce this behaviour, which relies on human decisions on investment of effort, but instead seeks to understand *how* users resolve trade-offs, regardless of the effort made.

We begin the process of choosing an option by evaluating each pair of options and assessing their costs and benefits. First, we analyse the benefits of option o_1 compared to option o_2 for each attribute, and do the same for o_2 compared to o_1. Benefits are captured by a real value from 0 to 1, indicating how much better one option is than another, with respect to to one attribute. If the OAPM indicates that o_1 is not better than o_2 for an attribute *att*, then the benefit is 0, otherwise, to compute this benefit, we use the preference used to set the OAPM value. Having considered attributes in isolation, we now examine overall option benefits, via the priorities provided. Based on priorities, we build an attribute partial order, associating one level with each attribute. A function is adopted to generate attribute weights, and we calculate the overall benefits from o_1 with respect to o_2 using a weighted sum. It is important to highlight that benefits are obtained solely from high-level preferences, without requiring further interactions with the user.

If there are no dominated options in the set of acceptable options then, for any two options, one is better for some attributes and the same applies for the other. As a consequence, a trade-off must be resolved to choose one of the two options. According to Simonson and Tversky [3], people not only consider the two options being compared and their costs and benefits, but also the cost and benefit relationship (ratio), which is positioned in relation to this ratio between other options. This is referred to as *trade-off contrast*. In addition, humans also consider how *extreme* options are. Extreme options have a large improvement for some attributes, e.g. quality, and a high penalty for others, e.g. price. In general, humans avoid extreme options [3], referred to as *extremeness aversion*. We therefore incorporate two new factors in the process of choosing an option, based on a function that shows the trade-off between two options and how extreme they are.

We have analysed three aspects of options: benefits, trade-off relative to available options, and extremeness. The last two aspects are also seen as benefits (or costs): if the trade-off between to options is better according to the average of the trade-off between every other pair of options, it is also a benefit, and the less extreme option has a benefit in comparison to the more extreme. The final value of an option with respect to another is thus a weighted sum of these benefits. Based on the v function, we identify the chosen option as better than or equal to every other option. If different options have the same value with respect to another, and they are better than every other option, we randomly choose one of them.

6. CONCLUSIONS

Intelligent agents provided with mechanisms that enable them to reason about preferences and make choices on behalf of users are a promising solution for reducing user effort in the automation of tasks. In this paper, we propose an automated decision making technique, which chooses an option from the set of those available based on preferences and priorities expressed in a high-level preference language. We improve decision-making by incorporating user-centric principles (trade-off contrast and extremeness aversion) that are not explicitly expressed as preferences. Based on an empirical evaluation, we can conclude that our technique makes choices as good as those of a (human) domain expert.

7. REFERENCES

[1] I. Nunes, S. Barbosa, and C. Lucena. Understanding how users express preferences: a user study. Technical Report CS-2010-19, University of Waterloo, Canada, 2010.

[2] J. W. Payne, J. R. Bettman, and E. J. Johnson. Adaptive strategy selection in decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(3):534–552, 1988.

[3] I. Simonson and A. Tversky. Choice in context: Tradeoff contrast and extremeness aversion. *Journal of Marketing Research*, 29(3):281–295, 1992.

[4] A. Tversky. Elimination by aspects: a theory of choice. *Psychological Review*, 79(4):281–299, 1972.

Lagrangian Relaxation for Large-Scale Multi-Agent Planning*

(Extended Abstract)

Geoffrey J. Gordon†, Pradeep Varakantham‡, William Yeoh‡, Hoong Chuin Lau‡, Ajay S. Aravamudhan‡ and Shih-Fen Cheng‡
†Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
‡School of Information Systems, Singapore Management University, Singapore 178902
†ggordon@cs.cmu.edu ‡{pradeepv,williamyeoh,hclau,ajaysa,sfcheng}@smu.edu.sg

ABSTRACT

Multi-agent planning is a well-studied problem with applications in various areas. Due to computational constraints, existing research typically focuses either on unstructured domains with many agents, where we are content with heuristic solutions, or domains with small numbers of agents or special structure, where we can find provably near-optimal solutions. In contrast, here we focus on provably near-optimal solutions in domains with many agents, by exploiting *influence limits*. To that end, we make two key contributions: (a) an algorithm, based on Lagrangian relaxation and randomized rounding, for solving multi-agent planning problems represented as large mixed-integer programs; (b) a proof of convergence of our algorithm to a near-optimal solution.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed AI

General Terms

Algorithms, Experimentation

Keywords

Multi-agent Planning, Lagrangian Relaxation

1. INTRODUCTION

Rapid progress in ubiquitous computing has enabled real-time delivery of contextualized information via devices (such as mobile phones and car navigation devices) over wide areas. As a result, a new kind of information service for mass user support is beginning to emerge. Examples include services that coordinate movements of first responders during a disaster rescue [1], movements of taxis in a fleet [3] and movements of visitors in leisure destinations (such as theme parks or world expositions). In these services, users are typically represented by computational agents that perform real-time planning and adaptation. Designing coordination mechanisms that can govern these services in ways that meet global criteria such as fairness, revenue maximization, stability/convergence, and efficient resource utilization is a research challenge. Motivated by this challenge, we present an algorithm, based on Lagrangian relaxation and randomized rounding, for large-scale multi-agent planning problems. We prove convergence to an optimal solution as the number of agents increases; in fact, the quality of the solution actually *improves* as the problem size increases.

2. ILLUSTRATIVE DOMAIN

We motivate our work with a theme park crowd management problem, represented with a tuple $\langle A, P, \{A(p_i)\}_1^n, \{d_{a_i}\}_1^k, \{U_i\}_1^n, H\rangle$, where $A = \{a_i\}_1^k$ is the set of attractions in the theme park; $P = \{p_i\}_1^n$ is the set of patrons in the theme park; $A(p_i) \subseteq A$ is the subset of attractions that patron p_i prefers to visit; d_{a_i} is the service rate of attraction a_i, that is, the number of patrons it can serve per time step; U_i is the utility function of patron p_i; and H is the time horizon. The goal is to find the route π_i for each patron p_i such that the sum of utilities $U_i(\pi_i)$ over all patrons is maximized.

3. MULTI-AGENT PLANNING PROBLEM

We represent the multi-agent planning problems as a large-scale mixed-integer program with special structure. This representation is very general, subsuming for example Markov decision processes, network flows, and graphical models such as influence diagrams, via reductions based on sampling scenarios [2]. Our chief assumptions are *factored structure*, the existence of *local planning subroutines*, and an *influence limit* for each agent. The efficiency of our algorithm will depend on the factored structure and the number and difficulty of local planning problems; our solution quality bounds will improve with more agents and tighter influence limits.

In more detail, we suppose that agent i's plan is represented by a set of decision variables $x_i \in \mathbb{R}^{n_i}$, subject to *local* constraints $A_i x_i = b_i$, $x_i \in X_i$ and *local* costs $c_i^\top x_i$. The agents interact through *coupling* constraints $\min f_j(y_j)$, where $y_j = \sum_{i=1}^n \ell_{ij}^\top x_i$ is resource consumption and $f_j : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ is a closed proper convex function representing resource cost. The *global planning* problem is

*This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Inputs: $c_i, \mathcal{C}_i, \ell_{ij}, f_j, \eta, T, \epsilon_j, \alpha_j^{\max}, \alpha_j^{\min}$ Outputs: $\bar{x}_i, \bar{\lambda}_j$

$\lambda_{j0} \leftarrow 0$ $\quad j = 1 \dots m$

for $t \leftarrow 1, 2, \dots, T$

$\quad x_{it} \leftarrow \arg\max_x [c_i^\top x - \sum_{j=1}^m \lambda_j \ell_{ij}^\top x]$ $\quad i = 1 \dots n$

$\qquad$ s.t. $x \in X_i,\ A_i x = b_i$

$\quad y_j \leftarrow \epsilon_j + \sum_{i=1}^n \ell_{ij}^\top x_{it}$ $\quad j = 1 \dots m$

$\quad z_j \leftarrow \arg\max_z [\lambda_j z - f_j(z)]$ $\quad j = 1 \dots m$

$\quad \lambda_{jt} \leftarrow \lambda_{j,t-1} + \frac{\eta}{\sqrt{t}}(y_j - z_j)$ $\quad j = 1 \dots m$

$\quad \lambda_{jt} \leftarrow \max(\alpha_j^{\min}, \min(\alpha_j^{\max}, \lambda_{jt}))$ $\quad j = 1 \dots m$

$\quad \bar{x}_i \leftarrow \frac{1}{t} \sum_{k=1}^t x_{ik}$ $\quad i = 1 \dots n$

$\quad \bar{\lambda}_j \leftarrow \frac{1}{t} \sum_{k=1}^t \lambda_{jk}$ $\quad j = 1 \dots m$

round $\bar{x}_i$ to x_i as described in text $\quad i = 1 \dots m$

Figure 1: SLR Pseudocode

therefore:

$$\max_x \ V_p(x) \ \text{ s.t. } \ A_i x_i = b_i,\ x_i \in X_i \quad \forall i \tag{1}$$

$$V_p(x) = \sum_{i=1}^n c_i^\top x_i - \sum_{j=1}^m f_j \left(\sum_{i=1}^n \ell_{ij}^\top x_i \right)$$

This problem is NP-hard and inapproximable; but, we can take advantage of a limit on the largest *influence* of any agent to solve it efficiently. More formally, we assume, first, that no agent controls a disproportionate share of the utility or resources: there is a constant $U > 0$ such that

$$-\frac{U}{n}|V_p^*| \le c_i^\top x_i \le \frac{U}{n}|V_p^*| \tag{2}$$

$$-\frac{U}{n}|y_j^*| \le \ell_{ij}^\top x_i \le \frac{U}{n}|y_j^*| \tag{3}$$

for all i, j, and $x_i \in X_i$. Here V_p^* is the optimal value in Eq. 1 and y_j^* is the usage of resource j in some optimal solution. Second, we suppose that the optimization problem as a whole is well conditioned: suppose we redefine the consumption cost in Eq. 1 to be

$$f_j\left(\epsilon_j + \textstyle\sum_{i=1}^n \ell_{ij}^\top x_i\right) \tag{4}$$

for some small $\epsilon_j \ge 0$. Let V_ϵ^* be the optimal value of Eq. 1 in this case. Then, we assume that there exists an $\epsilon_{\max} > 0$, a $\kappa > 0$, and a $\Delta > 0$ such that, whenever $0 \le \epsilon_j \le \epsilon_{\max}$,

$$V_\epsilon^* \ge V_p^* - \kappa \textstyle\sum_j \epsilon_j - \Delta/n \tag{5}$$

4. SLR ALGORITHM

Fig. 1 shows the *Subgradient Lagrangian Relaxation* (SLR) algorithm. Inputs are the problem parameters c_i, A_i, b_i, ℓ_{ij}, f_j (as in Eq. 1); learning rate $\eta > 0$ and number of iterations T; bounds $\alpha_j^{\max}$ and $\alpha_j^{\min}$ on the slope of f_j; and target margins ϵ_j. Outputs are expected plans $\bar{x}_i$ for each agent, as well as prices $\bar{\lambda}_j$ for each resource; the latter can be used to check convergence.

We cannot directly execute the final aggregated policy $\bar{x}$: since the domains X_i are typically non-convex, averaging feasible solutions does not typically yield a feasible solution. To remedy this problem, we use randomized rounding: each agent independently picks a random locally-feasible policy x_i according to a distribution which makes $\mathbb{E}(x_i) = \bar{x}_i$. (One such distribution is the uniform distribution over x_{it} for $t = 1 \dots T$.) To ensure feasibility, we set the margin ϵ_j to trade off total predicted utility against the possibility of violating resource constraints.

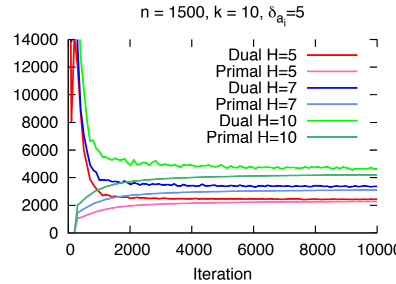


Figure 2: Experimental Results of SLR

The following theorem shows that we can set the parameters of SLR to guarantee that rounding yields a high-quality plan with high probability, and that, with these parameters, the expected runtime of SLR will be a low-order polynomial in the problem size. In particular, we can pick any desired failure probability, say $\delta = 0.01$, and a decreasing convergence tolerance, say $\gamma = 1/\sqrt{n}$. Then, we can set $\epsilon = \Theta(\frac{\ln 1/\delta}{\sqrt{n}})$, $\eta = \Theta(1)$, and $T = \Theta(\gamma^{-2}) = \Theta(n)$ to achieve low error, polynomial runtime, and high success probability. (And, we can make the success probability arbitrarily close to 1 by repeating the rounding step.)

THEOREM 1. *Suppose influence limits are guaranteed by Eqs. 2–5. Fix $\epsilon \le \epsilon_{\max}$, set $\epsilon_j = \epsilon$ for all j, and run SLR (Fig. 1) to some tolerance γ. Let each agent randomize independently with $\mathbb{E}(x_i) = \bar{x}_i$. Set*

$$\delta = e^{-n\epsilon^2/2U^2|V_p^*|^2} + m e^{-n\epsilon^2/2U^2|y_j^*|^2}$$

Then, with probability at least $1 - \delta$,

$$V_p(x) \ge V_p^* - \Delta/n - (\kappa m + 1)\epsilon - \gamma$$

5. EXPERIMENTAL RESULTS

For our experiments, using the notation from above, we set $|A| = 10$, $\delta_{a_i} = 5$ for all attractions, $n = 1500$, $k = 10$, and vary H from 5 to 10. Fig. 2 shows a set of representative results, where we plot the primal and dual values (from Eq. 1 and its dual) across iterations. The primal and dual values increase with H, since higher H lets some patrons visit attractions that they would otherwise have skipped. Convergence is fast for all problems, and the duality gap is small, indicating that we have reached a near-optimal solution in this large-scale problem instance.

6. REFERENCES

[1] S. Koenig, P. Keskinocak, and C. Tovey. Progress on agent coordination with cooperative auctions [Senior Member Paper]. In *Proceedings of AAAI*, pages 1713–1717, 2010.

[2] A. Ng and M. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Proceedings of UAI*, pages 406–415, 2000.

[3] P. Varakantham, S.-F. Cheng, and T. D. Nguyen. Decentralized decision support for an agent population in dynamic and uncertain domains. In *Proceedings of AAMAS*, pages 1147–1148, 2011.

Tree-based Pruning for Multiagent POMDPs with Delayed Communication

(Extended Abstract)

Frans A. Oliehoek
MIT CSAIL / Maastricht University
Maastricht, The Netherlands
frans.oliehoek@maastrichtuniversity.nl

Matthijs T.J. Spaan
Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

ABSTRACT

Multiagent POMDPs provide a powerful framework for optimal decision making under the assumption of instantaneous communication. We focus on a delayed communication setting (MPOMDP-DC), in which broadcast information is delayed by at most one time step. Such an assumption is in fact more appropriate for applications in which response time is critical. However, naive application of incremental pruning, the core of many state-of-the-art POMDP techniques, is intractable for MPOMDP-DCs. We overcome this problem by introducing a *tree-based pruning* technique. Experiments show that the method outperforms naive incremental pruning by orders of magnitude, allowing for the solution of larger problems.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance

Keywords

Multiagent planning under uncertainty, Multiagent POMDP, Delayed communication

1. INTRODUCTION

Planning under uncertainty in multiagent systems can be neatly formalized as a *decentralized partially observable Markov decision process (Dec-POMDP)*, but solving a Dec-POMDP is a complex (NEXP-complete) task. Communication can mitigate some of these complexities; by allowing agents to share their individual observations the problem reduces to a so-called *multiagent POMDP (MPOMDP)*, a special instance of the standard POMDP [3] which is 'merely' in PSPACE. However, this model requires the agents to perform full synchronization of their knowledge before selecting a next action, which is inappropriate in domains in which agents may need to act fast in response to their individual observations.

In this paper we focus on a class of problems where agents share their individual observations with a one step delay. That is, agents act using a *one step delayed sharing pattern*, resulting in an *MPOMDP with delayed communication (MPOMDP-DC)*. Solutions for such settings are also useful under longer delays [5]. Moreover, this class is particularly interesting, because it avoids the delay in action selection due to synchronization, while it is very similar to the standard POMDP. However, even though dynamic programming algorithms date back to the seventies [2], computational difficulties have limited the model's applicability.

The MPOMDP-DC value function is piecewise-linear and convex over the joint belief space [2], which is a property exploited by many regular POMDP solvers. However, *incremental pruning (IP)* [1], that performs a key operation, the so-called *cross-sum*, more efficiently, is not directly able to achieve the same improvements under delayed communication. A problem is the need to loop over a number of decision rules that is exponential both in the number of agents and in the number of observations.

In this paper, we target this additional complexity by proposing tree-based pruning with memoization, TBP-M, a method that operates over a tree structure in order to perform the cross-sum operation. Our experimental results indicate that it successfully avoids duplicate work by caching the result of computations at internal nodes and thus accelerates computation (at the cost of memory).

2. MODEL

An MPOMDP consists of the following componenents: a finite set of n agents; a finite set of states $\mathcal{S}$; a set $\mathcal{A} = \{a^1, \ldots, a^{|\mathcal{A}|}\}$ of joint actions $a = \langle a_1,...,a_n \rangle$; a set $\mathcal{O} = \{o^1, \ldots, o^{|\mathcal{O}|}\}$ of joint observations $o = \langle o_1,...,o_n \rangle$; a transition and observation function that specify the probabilities $P^a(s'|s)$ and $O^a(o|s)$; a reward function that specifies the reward $R^a(s)$; and h is the (finite) horizon. An MPOMDP-DC is an MPOMDP where communication is received with a one-step delay. The joint policy $\pi = (\delta^0, \delta^1, \ldots, \delta^{h-1})$ in such settings is a sequence of joint decision rules that specify an individual decision rule $\delta^t = \langle \delta^t_1, \ldots, \delta^t_n \rangle$ for each agent. Each δ^t_i maps $\langle b^{t-1}, a^{t-1}, o^t_i \rangle$-tuples to individual actions a^t_i. The value of an MPOMDP-DC is a function of joint beliefs:

$$Q^t(b,a) = R^a_{\mathcal{B}}(b) + \max_{\beta} \sum_{o} P^a(o|b) Q^{t+1}(b', \beta(o)), \qquad (1)$$

where $\beta = \langle \beta_1, \ldots, \beta_n \rangle$ is a decentralized control law which the agents use to map individual observations to actions: $\beta(o) = \langle \beta_1(o_1), \ldots, \beta_n(o_n) \rangle$. That way we decompose δ^t into a collection of β, one for each $\langle b,a \rangle$-pair.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

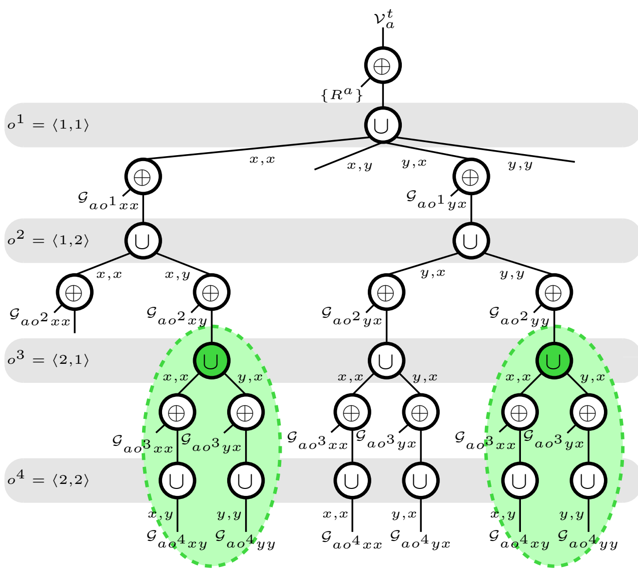


Figure 1: The computation tree of $\mathcal{V}_a^t$.

3. TREE-BASED PRUNING

As for an (M)POMDP, we can represent (1) using vectors [4]. However, in the MPOMDP-DC case not all combinations of next-stage vectors are possible; the actions they specify should be consistent with an admissible decentralized control law β. We can define 'back projected' vectors $g_{aoa'} \in \mathcal{G}_{aoa'}$ (see [4]). From these we construct the parsimonious representation

$$\mathcal{V}_a^t = \texttt{Prune} \bigcup_{\beta \in B} \left(\{R^a\} \oplus \mathcal{G}_{ao^1\beta(o^1)} \oplus \cdots \oplus \mathcal{G}_{ao^{|\mathcal{O}|}\beta(o^{|\mathcal{O}|})} \right) \quad (2)$$

where the cross-sum $A \oplus B = \{a + b \mid a \in A, b \in B\}$.

A naive way of performing incremental pruning (IP) [1] is to perform IP for each β. Their number, however, is exponential both in the number of agents and in the number of observations. Moreover, this method performs a lot of duplicate work. E.g., there are many β that specify $\beta(o^1) = a^k, \beta(o^2) = a^l$, but for each of them $\texttt{Prune}(\mathcal{G}_{ao^1a^k} \oplus \mathcal{G}_{ao^2a^l})$ is recomputed. In order to overcome these drawbacks, we propose a different approach: for each β, we directly construct construct the parsimonious representation via a computation tree.

In particular, it is possible to interpret β as a vector of joint actions, $\langle a_{(1)} \ldots a_{(|\mathcal{O}|)} \rangle$, where $a_{(j)}$ denotes the joint action selected for the j-th joint observation. This allows us to decompose the union over β into dependent unions over joint actions, resulting in the computation tree illustrated in Fig. 1 for a fictitious 2-action (x and y) 2-observation (1 and 2) MPOMDP-DC. The root of the tree, $\mathcal{V}_a^t$, is the result of the computation. There are two 2 types of internal, or operator, nodes: cross-sum and union. All the leafs are sets of vectors. An operator node n takes as input the sets from its children, and propagates the result up to its parent. The j-th union node on a path from root to leaf performs the union $\cup_{a_{(j)}}$ and thus has children corresponding to different assignments of a joint action to o^j (indicated by the gray bands). It is important to realize that the options available for $a_{(j)}$ depend on the action choices $\left(a_{(1)}, \ldots, a_{(j-1)}\right)$ made higher up in the tree; given those earlier choices, some $a_{(j)}$ may lead to conflicting individual actions for the same individual observation.

Now, to compute $\mathcal{V}_a^t$ we propose *tree-based (incremental) pruning (TBP)*: it expands the computation tree and, when the results are being propagated to the top of the tree, it prunes dominated vectors at each internal node. However, Fig. 1 shows another important issue: there are identical sub-trees in this computation tree, as indicated by the dashed green ovals, which means that we would be doing unnecessary work. We address this problem by memoization, i.e., caching of intermediate results, and refer to the resulting method as TBP-M.

Problem(h)	TBP-M	Naive IP	TBP-noM
Dec-Tiger(5)	0.13	0.23	0.09
Dec-Tiger(15)	0.98	2.54	1.19
OneDoor(3)	53.64	304.72	56.53
GridSmall(2)	3.93	64.03	3.80
MG2x2(2)	171.07	382093.00	516.03
MG2x2(4)	1115.06		2813.10
D-T Creaks(2)	63.14	109.27	121.99
D-T Creaks(5)	286.53	8277.32	2046.73
Box Push.(2)	132.13	1832.98	1961.38

Table 1: Timing results (in s).

Table 1 shows timing results for six benchmark problems, for a set of planning horizons (depending on the problem). We can see that for all domains TBP-M outperforms Naive IP, often by an order of magnitude and up to 3 orders of magnitude. We also compared against TBP-noM: a strawman version of TBP-M that does not perform any memoization and re-computes duplicate parts of the tree. It allows us to see the effect of tree-based pruning, without the extra speedups provided by memoization: memoization significantly speeds up computations.

4. CONCLUSIONS

We addressed the problem of the additional complexity that the MPOMDP-DC backup exhibits over the backup for the MPOMDP. We showed that the DC backup operator can be represented as a computation tree and presented TBP-M, a method to exploit this tree structure. An empirical evaluation on a number of benchmark problems that indicates that TBP-M can realize speedups of 3 orders of magnitude over the naive IP baseline.

Acknowledgments

Research supported in part by AFOSR MURI project #FA9550-09-1-0538 and NWO CATCH project #640.005.003. M.S. is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF).

5. REFERENCES

[1] A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *UAI*, pages 54–61. Morgan Kaufmann, 1997.

[2] K. Hsu and S. Marcus. Decentralized control of finite state Markov processes. *IEEE Transactions on Automatic Control*, 27(2):426–431, Apr. 1982.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[4] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Dec-POMDPs with delayed communication. In *MSDM (AAMAS Workshop)*, May 2007.

[5] M. T. J. Spaan, F. A. Oliehoek, and N. Vlassis. Multiagent planning under uncertainty with stochastic communication delays. In *ICAPS*, pages 338–345, 2008.

Planning in the Logics of Communication and Change (Extended Abstract)

Pere Pardo
Institut d'Investigació en Intel·ligència Artificial
Campus de la UAB s/n, E-08193
Bellaterra, Spain
pardo@iiia.csic.es

Mehrnoosh Sadrzadeh
Oxford University
Dept. of Computer Science, Wolfson Building
Parks Rd, Oxford OX13QD, UK
mehrnoosh.sadrzadeh@cs.ox.ac.uk

ABSTRACT

We adapt backward planning to Logics of Communication and Change (LCC), that model how do actions, announcements and sensing change facts and agents' beliefs. An LCC planner takes into account the epistemic effects of planned actions upon other agents, if their beliefs are relevant to her goals. Our results include: a characterization of frame axioms as theorems in $*$-free LCC, and soundness and completeness results for deterministic planning and strong planning in the non-deterministic case.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents

General Terms

Algorithms, Theory

Keywords

Dynamic Epistemic Logic, Planning, Communication

1. INTRODUCTION

In the present contribution, we adapt backward planning techniques to the Logics of Communication and Change (LCC). An LCC reasoning agent (who can foresee the possible epistemic effects of her actions and communications) is endowed with planning abilities to achieve some goals by means of LCC action models. This greatly expands on the social complexity of multi-agent planning scenarios.

EXAMPLE 1.1. *Agent* a *is having a party, and would like her friend* b *to assist without their friend* c*. If* b *is secretive, a private announcement to* b *will suffice. However, suppose that* b *tells everything to* c*. Yet, if* a *knows that* c *only assists to parties with beer, while* b*'s interests also include jazz music (the party will include both), a solution may consist in informing* b *only about jazz.*

EXAMPLE 1.2. *Agent* a *just bet agent* b *10 coins that the next coin toss will land heads (h);* a *can sense and even flip the coin without* b *ever suspecting it. A successful plan seems to be:* toss *the coin; if sense that h, then announce it to* b*; otherwise flip the coin and announce h.*

Related Work: [1] studies forward planning in LCC [2], under a semantic approach. Because of the large number of LCC actions available (one announcement per formula) forward planning faces the state-explosion problem. Thus, (deduction-based) backward planning seems appropriate.

2. LCC AND FRAME AXIOMS.

DEFINITION 2.1. *The language $\mathcal{L}_{\text{PDL}}$ of $*$-free* PDL*, for a given sets of atoms $p \in \mathsf{Var}$ and agents $a \in \mathsf{Ag}$ is:*

$$\varphi \quad ::= \quad \mathsf{Var} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi_1 ; \pi_2 \mid \pi_1 \cup \pi_2$$

with the usual abbreviations for $\bot, \vee, \leftrightarrow$ and $\langle\pi\rangle$.

For LCC, we read an atomic program $[a]$ as *a believes that*, composition ";" is nested belief, and $\cup$ defines group belief.

LCC extends $\mathcal{L}_{\text{PDL}}$ with modalities for pointed action models $[\mathsf{U}, \mathsf{e}]$. An action model is $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$, with an action $\mathsf{e} \in \mathsf{E}$ being defined by a *precondition* $\mathsf{pre}(\mathsf{e})$, a LCC-formula, and a *postcondition* $\mathsf{post}(\mathsf{e})$, a substitution $\sigma : p \mapsto \varphi$ expressing that after executing e, the truth-value of p becomes that of φ (before the execution). In the present paper, though, we limit to the case $\sigma(p) \in \{\top, p, \bot\}$, studied in [3]. The *accessibility relations* $\mathsf{e}\mathsf{R}(a)\mathsf{f}$ denote actions f that cannot be distinguished from e by a. The skip action is given by the identity substitution. A truthful (resp. lying) communication of p by agent a to a set of (credulous) agents $B \subseteq \mathsf{Ag}$, denoted $p!^a_B$ (resp. $p\dagger^a_B$) has $\mathsf{pre}(p!^a_B) = p$ (resp. $\mathsf{pre}(p\dagger^a_B) = \neg p$).

We further extend the language of LCC with composition $\otimes$ and choice $\cup$ for action models.

PROPOSITION 2.2. *The axioms of LCC [2] plus the next two axioms are a complete axiomatization of* $\text{LCC} + \{\otimes, \cup\}$.

$$[\mathsf{U}^\cup, \mathsf{e} \cup \mathsf{e}']\varphi \quad \leftrightarrow \quad [\mathsf{U}, \mathsf{e}]\varphi \wedge [\mathsf{U}, \mathsf{e}']\varphi$$
$$[\mathsf{U} \otimes \mathsf{U}, \mathsf{e} \otimes \mathsf{e}']\varphi \quad \leftrightarrow \quad [\mathsf{U}, \mathsf{e}][\mathsf{U}, \mathsf{e}']\varphi$$

Frame axioms describe the conditions for a formula φ to be preserved after executing e. The presence of ontic actions makes LCC frame axioms $\mathsf{FA}(\mathsf{e}, \varphi)$ non-trivial, see Figure 1. The naive form cannot address the cases $p \vee q$ or $[a]p$.

PROPOSITION 2.3. *The frame axioms $\mathsf{FA}(\mathsf{e}, \varphi)$ as in Fig. 1 (Right) can be inductively defined, and are valid in* LCC*: if $\mathsf{cond}(\mathsf{FA}(\mathsf{e}, \varphi))$ holds, then $\models \mathsf{ant}(\mathsf{FA}(\mathsf{e}, \varphi)) \to [\mathsf{U}, \mathsf{e}]\varphi$.*

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

if $\not\models [\mathsf{U},\mathsf{e}]\neg\varphi$	if $\mathsf{cond}(\mathsf{FA}(\mathsf{e},\varphi))$
then $\models \varphi \to [\mathsf{U},\mathsf{e}]\varphi$	then $\models \mathsf{ant}(\mathsf{FA}(\mathsf{e},\varphi)) \to [\mathsf{U},\mathsf{e}]\varphi$

Figure 1: Frame axiom for e,φ: (Left) Naive form. (Right) Correct form.

3. DETERMINISTIC PLANNING.

A planning domain is defined by a set $A \subseteq \mathsf{E}$ of available actions, and a pair (T, G), where $T, G \subseteq \mathcal{L}_{\mathrm{PDL}}$ describe the initial state and goals. Deterministic actions are just some subset $A \subseteq \mathsf{E}$ in LCC + $\{\otimes\}$. Given $\mathsf{e} \in A$, its effects are $\mathsf{X}(\mathsf{e}) = \{\psi \in \mathcal{L}_{\mathrm{PDL}} :\models \mathsf{pre}(\mathsf{e}) \to [\mathsf{U},\mathsf{e}]\psi\}$.

Definition 3.1. *Given a planning domain (T, G), actions A, and a program π, we say π is a* solution *for (T, G) in A iff (1) $\vdash \bigwedge T \to \langle\pi\rangle\top$, and (2) $\vdash \bigwedge T \to [\pi]\bigwedge G$.*

A solution must (1) be executable in T, and (2) lead to G. To solve a planning domain (T, G), we adopt the Breadth First Search (BFS) for incremental backward planning: starting with the empty plan for G, at each step $\pi_k = (\kappa_0, \ldots, \kappa_k)$ we add a step $\pi_{k+1} = (\kappa_0, \ldots, \kappa_k, \kappa_{k+1})$, delete the open goals of π_k enforced by κ_{k+1}, and add as new open goals $\mathsf{pre}(\kappa_{k+1})$. This step κ can be an action step $\mathsf{e} \in A$, or a proof step $\mathcal{A}$. Proof steps split complex goals, e.g. $\varphi\wedge\psi$, into simpler goals φ, ψ each of which can directly be enforced by some action $\mathsf{e} \in A$. This is done by means of a planned LCC-proof $\mathcal{A} = \langle\varphi, \psi, \ldots, \varphi \wedge \psi\rangle$, where $\mathsf{pre}(\mathcal{A}) = \{\varphi, \psi\}$ denotes the (non-tautological) premisses of $\mathcal{A}$ and $\mathsf{X}(\mathcal{A}) = \varphi\wedge\psi$ its conclusion. Action steps must respect the frame axioms $\mathsf{FA}(\mathsf{e}, \varphi)$ for each goals φ in π_k unaddressed by π_{k+1}. That is, for e_{k+1} to refine π_k into a *plan* π_{k+1}, the condition $\mathsf{cond}(\mathsf{FA}(\mathsf{e}_{k+1}, \varphi))$ must be true, and $\mathsf{ant}(\mathsf{FA}(\mathsf{e}_{k+1}, \varphi))$ must be added as an open goal of π_{k+1}. Finally, the set of open goals of π_{k+1} must also be consistent. Similar conditions apply to proof steps $\mathcal{A}$, to make π_{k+1} a plan. Note the plan $\pi = (\mathsf{e}_0, \mathcal{A}_0, \ldots, \mathcal{A}_n, \mathsf{e}_n)$ translates into logical form $[\mathsf{U}, \mathsf{e}_n]$,..., $[\mathsf{U}, \mathsf{e}_0]$, with action steps in inverse order, and where proof steps are omitted (LCC will enforce them anyway).

Theorem 3.2. *Let $(\mathsf{e}_0, \ldots, \mathsf{e}_n)$ be an output of the BFS algorithm for (T, G) in A. Then $[\mathsf{U}, \mathsf{e}_n] \ldots [\mathsf{U}, \mathsf{e}_0]$ is a solution for (T, G). Conversely, suppose some deterministic solution $[\mathsf{U}, \mathsf{e}_n] \ldots [\mathsf{U}, \mathsf{e}_0]$ exists for (T, G) in A. Then the BFS algorithm terminates with a solution for (T, G) in A.*

Planning in others' shoes For multi-agent scenarios, we can define an algorithm that computes the reactions to one's plan by other planner agents. Then, a plan is called *stable* if these reactions do not lead to a state where G is not satisfied.

Example 3.3. *(Cont'd) Recall Example 1.1.* b*'s goals are* $(\mathsf{beer} \vee \mathsf{jazz}) \to @\mathsf{party}(\mathsf{b})$ *as well as* $\{[\mathsf{b}]\varphi \to [\mathsf{c}]\varphi\}_{\varphi\in\mathcal{L}_{\mathrm{PDL}}}$*; and* c *has goal* $\mathsf{beer} \to @\mathsf{party}(\mathsf{b})$. *It can be seen that the naive solution* $\mathsf{beer}!^{\mathsf{a}}_{\mathsf{b}}$ *is not stable: agents' reactions lead to the output* $(\mathsf{beer}!^{\mathsf{a}}_{\mathsf{b}} \otimes \mathsf{beer}!^{\mathsf{b}}_{\mathsf{c}} \otimes \mathsf{go.party}(\mathsf{b}) \otimes \mathsf{go.party}(\mathsf{c}))$*, which makes* $\neg@\mathsf{party}(\mathsf{c})$ *false. In contrast,* $\mathsf{jazz}!^{\mathsf{a}}_{\mathsf{b}}$ *is stable: the output is* $\langle(\mathsf{jazz}!^{\mathsf{a}}_{\mathsf{b}} \otimes \mathsf{jazz}!^{\mathsf{b}}_{\mathsf{c}} \otimes \mathsf{go.party}(\mathsf{b})\rangle$.

4. NON-DETERMINISTIC PLANNING

For planning involving actions with disjunctive effects $\varphi_1 \vee \varphi_2$, one first stipulates actions e_i with $\varphi_i \in \mathsf{X}(\mathsf{e}_i)$. While $\mathsf{e}_1 \cup \mathsf{e}_2 \in A$ is available, individual actions e_i are not: $\mathsf{e}_i \in \mathsf{E} \smallsetminus A$.

A plan is now a 4-tuple: *plan* = (*sequence of actions*, *open goals*, *initial state*, *original goals*). We reduce the problem of building a non-deterministic plan into that of solving a sequence of deterministic planning problems. To do so, we define a plan set Π as a sequence of plans $\Pi = \langle\pi^{\xi}, \pi^{\xi'}, \ldots\rangle$ enumerated by sequences $\xi \in \{\emptyset\} \cup \{1, 2\}^{<\omega}$ and ordered lexicographically, e.g. $\emptyset <_{\mathsf{lex}} 1 <_{\mathsf{lex}} 11 <_{\mathsf{lex}} 12 <_{\mathsf{lex}} 2$. This ordering represents the order in which plans are solved. See Figure 2 for an illustration of the algorithm in Example 1.2. Non-deterministic planning is done by a series of BFS searches.

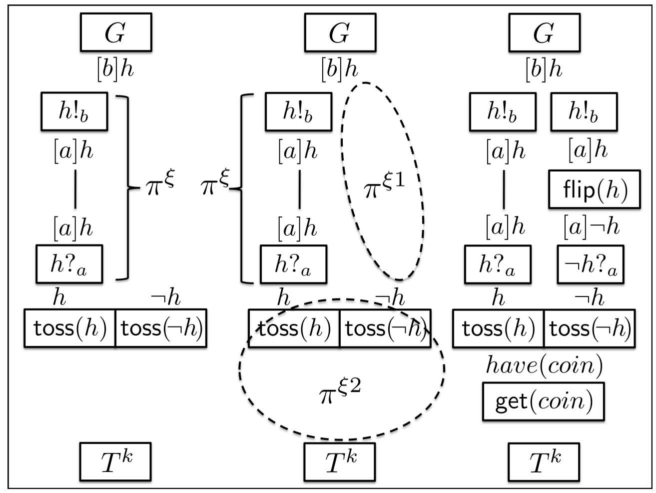


Figure 2: Refining π^{ξ} with $\mathsf{toss}(h) \cup \mathsf{toss}(\neg h)$ splits π^{ξ} into three deterministic plan search problems $\pi^{\xi}, \pi^{\xi 1}, \pi^{\xi 2}$.

To make sure that a plan π with $\mathsf{e}_1 \cup \mathsf{e}_2$ is logically acceptable, a reset action ρ might be needed to harmonize effects: $\mathsf{e}_1 \cup (\mathsf{e}_2 \otimes \rho)$. These ρ do not contribute to the success of π.

Theorem 4.1. *Let π be an output of the BFS for (T, G). Then π is a solution for (T, G) in A. Conversely, if a non-deterministic solution exists, then the BFS output for (T, G) in A is a solution.*

Conclusions and Future Work

We studied backward deterministic and strong planning for LCC logics. Several directions seem interesting: belief revision, the $*$ operator for strong cyclic planning and decidability/complexity issues, among others.

Acknowledgements: This work has been funded by projects AT (CSD 2007-022); LoMoReVI (FFI2008-03126-E/FILO FP006); ARINF (TIN2009-14704-C03-03); and the GenCat 2009-SGR-1434 and EPSRC grant EP/J002607/1.

5. REFERENCES

[1] T. Bolander and M. Birkegaard Andersen *Epistemic planning for single- and multi-agent systems* Journal of Applied Non-Classical Logics, (in press)

[2] J. van Benthem, J. van Eijck and B. Kooi *Logics of Communication and Change* Information and Computation, vol 204 pp. 1620–1662 (2006)

[3] H. van Ditmarsch, B. Kooi *Semantic results for ontic and epistemic change*, in Bonanno, van der Hoek and Wooldridge (eds.), LOFT 7, pp. 87–117 (2008)

Intention-Aware Planning under Uncertainty for Interacting with Self-Interested, Boundedly Rational Agents

(Extended Abstract)

Trong Nghia Hoang and Kian Hsiang Low
Department of Computer Science, National University of Singapore, Republic of Singapore
{nghiaht, lowkh}@comp.nus.edu.sg

ABSTRACT

A key challenge in non-cooperative multi-agent systems is that of developing efficient planning algorithms for intelligent agents to perform effectively among boundedly rational[1], self-interested (i.e., non-cooperative) agents (e.g., humans). To address this challenge, we investigate how intention prediction can be efficiently exploited and made practical in planning, thereby leading to efficient intention-aware planning frameworks capable of predicting the intentions of other agents and acting optimally with respect to their predicted intentions.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent systems

General Terms

Algorithms, Performance, Experimentation, Theory

Keywords

Planning (single and multi-agent), Modeling other agents and self

1. INTRODUCTION

To date, existing planning frameworks for non-cooperative multi-agent systems (MAS) can be generally classified into: (a) *game-theoretic* frameworks rely on the well-established solution concepts of classical game theory to characterize interactions among self-interested agents; (b) *decision-theoretic* frameworks extend single-agent decision-theoretic planning framework (e.g., MDP, POMDP) by considering other agents as a stochastic part of the environment. However, such frameworks suffer from the following drawbacks: (a) the restrictive assumptions on other agents' behaviors, as implied by the solution concepts [3, 4]; (b) the failure in accounting for agents' deliberative and boundedly rational behaviors that cannot be sufficiently modeled as stochastic noise in the transition model. Alternatively, the *interactive POMDP* (I-POMDP) framework [2] attempts to explicitly account for the bounded rationality of self-interested agents by maintaining an agent's interactive beliefs over both the physical states and the other agents' beliefs. As a result, solving I-POMDP requires solving an exponential number of POMDPs [2], which are prohibitively expensive. To resolve the above issues, we propose practical and efficient formal, principled intention-aware planning frameworks for interacting with boundedly rational, self-interested agents:

- *Nested MDP* framework for interacting in fully observable environments (Section 2): inspired by [1], it constitutes a recursive reasoning formalism to predict the other agents' intention efficiently and such predictive information is then exploited to plan our agent's optimal interaction policy. The cost of solving nested MDP is linear in the length of time horizon and the depth of reasoning.
- *Intention-aware POMDP* (IA-POMDP) framework for interacting in partially observable environments (Section 3): it extends nested MDP by integrating it into POMDP for tracking our agent's belief. By exploiting problem structure in terms of the other agents' full observability, IA-POMDP can be solved efficiently in polynomial time.

2. NESTED MDP

Nested MDP constitutes a recursive reasoning process comprising k reasoning levels: at level 0, our agent's best response is computed by considering the other agent's actions as stochastic noise in an MDP's transition model. At level $k \geq 1$, our agent plans its optimal strategy by assuming that the other agent's strategy is based only on lower reasoning levels $0, \ldots, k-1$. Formally, nested MDP at level $k \geq 1$ for agent 1 is a tuple $M_1^k \triangleq (S, U, V, T, R, \{\pi_2^i\}_{i=0}^{k-1}, \phi)$ where

- S is a set of all possible states of the environment;
- U and V are sets of all possible actions available to agents 1 and 2, respectively;
- $T : S \times U \times V \times S \to [0, 1]$ denotes the transition probability of going from state $s \in S$ to state $s' \in S$ using agent 1's action $u \in U$ and agent 2's action $v \in V$;
- $R : S \times U \times V \to \mathbb{R}$ is the reward function of agent 1;
- $\pi_2^i : S \times V \to [0, 1]$ is the reasoning model at level $i < k$ predicting the mixed strategy of agent 2 for each state;
- $\phi \in (0, 1)$ is a discount factor.

The optimal value function of nested MDP M_1^k at level $k \geq 1$ for agent 1 satisfies the following Bellman equation:

$$\begin{aligned} U_1^k(s) &\triangleq \max_{u \in U} \sum_{v \in V} \widehat{\pi}_2^{k-1}(s, v)\ Q_1^k(s, u, v) \\ Q_1^k(s, u, v) &\triangleq R(s, u, v) + \phi \sum_{s' \in S} T(s, u, v, s')\ U_1^k(s') \end{aligned} \tag{1}$$

[1]Boundedly rational agents are subject to limited information, cognition, and time while making decisions.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

where the mixed strategy $\widehat{\pi}_2^{k-1}$ of the other agent 2 is predicted by averaging uniformly over all its reasoning models $\{\pi_2^i\}_{i=0}^{k-1}$ at levels $0, 1, \ldots, k-1$ because its actual level of reasoning is not known to our agent 1:

$$\widehat{\pi}_2^{k-1}(s, v) \triangleq \beta \sum_{i=0}^{k-1} \pi_2^i(s, v) \ . \quad (2)$$

Agent 2's reasoning model π_2^0 at level 0 is induced by solving a conventional MDP that represents agent 1's actions as stochastic noise in its transition model. To obtain agent 2's reasoning models $\{\pi_2^i\}_{i=1}^{k-1}$ at levels $i = 1, \ldots, k-1$, let $Opt_2^i(s)$ be the set of agent 2's optimal actions for state s induced by solving its nested MDP M_2^i, which recursively involves building agent 1's reasoning models $\{\pi_1^l\}_{l=0}^{i-1}$ at levels $l = 0, 1, \ldots, i-1$, by definition. Then,

$$\pi_2^i(s, v) \triangleq \begin{cases} |Opt_2^i(s)|^{-1} & \text{if } v \in Opt_2^i(s), \\ 0 & \text{otherwise.} \end{cases}$$

After predicting agent 2's strategy $\widehat{\pi}_2^{k-1}$ (2), agent 1's optimal policy (i.e., reasoning model) π_1^k at level k can be induced by solving its corresponding nested MDP M_1^k (1).

3. INTENTION-AWARE POMDP

To tackle partial observability, it seems obvious to first consider generalizing the recursive reasoning formalism of nested MDP. This approach yields two practical complications: (a) our agent's belief over both the physical states and the other agent's belief has to be modeled, and (b) the other agent's mixed strategy has to be predicted for each of its infinitely many possible beliefs. The I-POMDP framework faces both difficulties and consequently incurs a prohibitively expensive processing cost that involves solving exponential number of POMDPs [2]. In practice where we are subject to limited information, cognition, and time, we hardly recall performing such sophisticated modeling of our human counterpart during interaction. Instead, we often make satisficing decisions by limiting our predictions of counterpart's strategy to some specific states and considering how likely each state is based on our belief over these states.

To realize this intuition, we propose an alternative *intention-aware POMDP* (IA-POMDP) framework by exploiting the following structural assumption: the environment is fully observable to the other agent. Such an assumption is practical to make when the other agent's sensing capability is superior (e.g., human) or we do not know nor want to underestimate the other agent's sensing capability, especially in competitive scenarios. Surprisingly, this simple assumption alleviates both difficulties faced by I-POMDP, thus making IA-POMDP computationally efficient. Compared to existing game-theoretic frameworks [3, 4], our assumption is far less restrictive. More importantly, though it makes IA-POMDP less expressive than I-POMDP, it significantly boosts the practicality of decision-theoretic planning frameworks for non-cooperative MAS. Formally, IA-POMDP for agent 1 is defined as a tuple $(S, U, V, O, T, Z, R, \widehat{\pi}_2^k, \phi, b_0)$ where

- S is a set of all possible states of the environment;
- U and V are sets of all possible actions available to our agent 1 and the other agent 2, respectively;
- O is a set of all possible observations of our agent 1;
- $T : S \times U \times V \times S \to [0, 1]$ is a transition function that depends on the joint actions of both agents;
- $Z : S \times U \times O \to [0, 1]$ denotes the probability $Pr(o|s', u)$ of making observation $o \in O$ in state $s' \in S$ using our agent 1's action $u \in U$;
- $R : S \times U \times V \to \mathbb{R}$ is the reward function of agent 1;
- $\widehat{\pi}_2^k : S \times V \to [0, 1]$ denotes the predictive probability $Pr(v|s)$ (i.e., predicted mixed strategy) of agent 2 selecting action v in state s and is derived using (2) by solving its nested MDPs at levels $0, \ldots, k$;
- $\phi \in (0, 1)$ is a discount factor; and
- $b_0 \in \Delta(S)$ is a prior belief over the states of environment.

Solving IA-POMDP involves choosing the policy that maximizes the expected total reward with respect to the prediction of agent 2's mixed strategy using nested MDP. The optimal value function of IA-POMDP for our agent 1 satisfies the following Bellman equation:

$$V_{n+1}(b) = \max_u Q_{n+1}(b, u)$$
$$Q_{n+1}(b, u) = R(b, u) + \phi \sum_{v,o} Pr(v, o|b, u) \ V_n(b')$$

where our agent 1's expected immediate payoff is

$$R(b, u) = \sum_{s,v} R(s, u, v) \ Pr(v|s) \ b(s)$$

and the belief update is

$$b'(s') = \beta \ Z(s', u, o) \sum_s T(s, u, v, s') \ Pr(v|s) \ b(s) \ .$$

Like POMDP, the optimal value function $V_n(b)$ of IA-POMDP can be approximated arbitrarily closely (for infinite horizon) by a piecewise-linear and convex function that takes the form of a set V_n[2] of α vectors: $V_n(b) = \max_{\alpha \in V_n}(\alpha \cdot b)$. Thus, solving IA-POMDP is equivalent to computing the corresponding set of α vectors, which grows exponentially with the time horizon: $|V_{n+1}| = |U||V_n|^{|V||O|}$. To avoid this exponential blow-up, IA-POMDP inherits essential properties from POMDP that make it amenable to be solved by existing sampling-based algorithms, such as [5], of POMDP in polynomial time. For interested readers, a further technical discussion of IA-POMDP as well as an empirical evaluation of our proposed frameworks can be found in the extended version of this paper[3].

4. REFERENCES

[1] C. F. Camerer, T. H. Ho, and J. K. Chong. A cognitive hierarchy model of games. *Quarterly J. Economics*, 119(3):861–898, 2004.

[2] P. J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multi-agent settings. *JAIR*, 24:49–79, 2005.

[3] J. Hu and M. P. Wellman. Multi-agent reinforcement learning: Theoretical framework and an algorithm. In *Proc. ICML*, pages 242–250, 1998.

[4] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. ICML*, pages 157–163, 1994.

[5] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. IJCAI*, pages 1025–1032, 2003.

[2] With slight abuse of notation, the value function is also used to denote the set of corresponding α vectors.

[3] http://www.comp.nus.edu.sg/∼lowkh/pubs/aamas2012e.pdf

Delayed Observation Planning in Partially Observable Domains

(Extended Abstract)

Pradeep Varakantham†, Janusz Marecki‡
†School of Information Systems, Singapore Management University, Singapore
‡IBM Watson Research Lab, New York, NY
†pradeepv@smu.edu.sg ‡marecki@us.ibm.com

ABSTRACT

Traditional models for planning under uncertainty such as Markov Decision Processes (MDPs) or Partially Observable MDPs (POMDPs) assume that the observations about the results of agent actions are instantly available to the agent. In so doing, they are no longer applicable to domains where observations are received with delays caused by temporary unavailability of information (e.g. delayed response of the market to a new product). To that end, we make the following key contributions towards solving Delayed observation POMDPs (D-POMDPs): (i) We first provide an parameterized approximate algorithm for solving D-POMDPs efficiently, with desired accuracy; and (ii) We then propose a policy execution technique that adjusts the policy at runtime to account for the actual realization of observations. We then show the performance of our techniques on POMDP benchmark problems with delayed observations where explicit modeling of delayed observations leads to solutions of superior quality.

Categories and Subject Descriptors

G [**3**]: Markov Processes

General Terms

Algorithms

Keywords

Partially Observable Markov Decision Process, Delayed Observations

1. INTRODUCTION

Recent years have seen a rise of interest in autonomous agents deployed in domains ranging from automated trading, traffic control, disaster rescue and space exploration. Simultaneously, research in devising control mechanisms for these agents has progressed significantly. Partially Observable Markov Decision Processes (POMDPs) have received considerable attention, due to their ability to capture the uncertainty of the outcomes of agent actions and the uncertainty in the agent observations of the environment. Research in POMDPs has allowed the POMDP solvers [5] to scale to domains with thousands of states, actions and observations while providing guarantees on solution quality.

Unfortunately, the problem of decision making with delayed observations has received scant attention in POMDP research. Delayed observation reasoning is particularly relevant in providing real time decisions based on traffic congestion/incident information [1] or in making decisions on new products before receiving the market response to a new product. There are always delays in receiving such information due to data fusion, computation, transmission and physical limitations of the underlying process. Existing research [1, 3] has provided (a) models to represent observation delay in the context of Markov Decision Problems; (b) Theoretical properties of the sufficient statistic and value function; and (c) optimal approaches for solving MDPs with fixed observation delays;. While those models and algorithms are extended to POMDPs, the optimal nature of the algorithms and other restrictions (such as fixed observation delays) decreases their scalability and applicability.

In this paper we remedy the shortcomings of the previous work for handling delayed observations, in three key contributions: (i) We provide a parameterized approximate algorithm for solving D-POMDPs with a desired accuracy; (ii) We propose a policy execution technique that adjusts the agent policy corresponding to delayed observations at runtime for improved performance; and (iii) Finally, we provide error bounds, theoretical properties and complexity results for the proposed approaches. In the experimental results, we illustrate that our planning and execution algorithms lead to improved performance in domains with observation delays.

2. MODEL: D-POMDP

We now introduce D-POMDP model to allow for rich modeling of delayed observations, extending the models proposed in [1, 3]. A D-POMDP is a tuple $\langle S, A, \Omega, P, R, O, \mathcal{X}\rangle$ whose only difference from a POMDP is $\mathcal{X}$—a set of random variables $\mathcal{X}_{s,a}(k)$ that specify the probability that an observation is delayed by k decision epochs, when action a is executed in state s. *An example of* $\mathcal{X}_{s,a}$ *would be the discrete distribution (0.5,0.3,0.2), where 0.5 is the probability of no delay, 0.3 is the probability of one time step delay and 0.2 is the probability of two time step delay in receiving the observation in state s on executing action a.* D-POMDPs extend POMDPs by modeling the observations that are de-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

layed and by allowing for actions to be executed prior to receiving these delayed observations. In essence, if the agent receives an observation immediately after executing an action, D-POMDPs behave exactly as POMDPs. However, if an observation does not reach the agent immediately, D-POMDPs behave differently from POMDPs. Rather than having to wait for an observation to arrive, a D-POMDP agent can resume the execution of its policy *prior to* receiving the observation. In short, a D-POMDP agent must balance the trade off of acting prematurely (without the information provided by the observations that have not yet arrived) versus executing stop gap (waiting) actions.

Our introduction of D-POMDPs is accompanied in the next section by a D-POMDP example for a classical "Tiger Domain" [4] wherein $S = \{s_{TigerLeft}, s_{TigerRight}\}$, $A = \{a_{OpenLeft}, a_{OpenRight}, a_{Listen}\}$, $O = \{o_{TigerLeft}, o_{TigerRight}\}$ and the observations $o_{TigerLeft}$, $o_{TigerRight}$ resulting from the execution of action a_{Listen} arrive with a delay sampled from a discrete probability distribution $\mathcal{X}$.

3. SOLVING D-POMDPS

In this paper, we are interested in providing quality bounded and efficient solutions for D-POMDPs. Our approach to solving a D-POMDP consists of two steps: (a) converting the D-POMDP to an approximately equivalent POMDP; and (b) employing an existing POMDP solver to solve the obtained POMDP. The key step is (a) and we provide a parameterized approach for making the conversion from D-POMDP to its approximately equivalent POMDP. The level of approximation is governed by an input parameter, D, which represents the number of delay steps considered in the planning process[1] . The extended POMDP obtained from the D-POMDP is defined as the tuple $\langle \overline{S}, A, \overline{\Omega}, \overline{P}, \overline{R}, \overline{O} \rangle$ where $\overline{S}$ is the set of extended states and $\overline{\Omega}$ is a set of extended observations that the agent receives upon executing its actions in extended states. $\overline{P}, \overline{R}, \overline{O}$ are the extended transition, reward and observations functions.

3.1 Online Policy Modification

The second key contribution of this paper is a technique for modifying the policy of a converted POMDP (from previous section) during execution. We assume here that the employed POMDP solver returns value vectors (along with dominating actions) across the belief space. Typically, the policy execution in a POMDP is initiated by executing the action at the root of the policy tree, selecting and executing the next action based on the received observation and so on. This type of policy execution suffices in normal POMDPs, however, in extending POMDPs corresponding to D-POMDPs, the policy execution can be improved. The key intuition here is that during policy execution the beliefs that an agent has can be outdated (due to not updating the belief once observations are delayed). Hence, the idea is to keep the belief state as updated as possible in an efficient manner, i.e. updating the beliefs as and when the delayed observations are received.

4. EXPERIMENTS

In this paper, we have experimented with different types of problems to evaluate the performance of our planning and execution algorithms. Since state-of-the-art POMDP solvers [5] are already capable of solving problems involving hundreds of thousands of states and we hope to implement our techniques on top of those approaches. However, in this paper, we will be focusing mainly on understanding how much our planning and execution approaches can improve with respect to quality as the delay distribution increases in complexity.

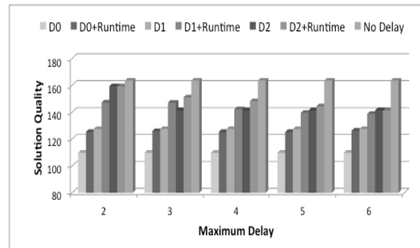


Figure 1: Comparison of solution quality

We experimented with benchmark problems in POMDP literature with different delay distributions. The main problems that we experimented with include: "tiger", "1d maze", "network", "4x4.95.POMDP" and "paint95", taken from Anthony Cassandra's POMDP page. Observation delay is defined by the set of discrete distributions $\mathcal{X}$ in the D-POMDP model. Our planning algorithms are represented as D0, D1, D2 etc., where the number corresponds to the D employed in the conversion of D-POMDP to POMDP. We compare the solution quality obtained by D0, D1 and D2 with and without the online policy modification component (Runtime) against the solution quality obtained if there was no delay in receiving observation. To solve the converted POMDP problems we employ the Point Based Value Iteration solver [2], but any of the existing solvers can be employed. Figure 1 shows the performance of our algorithms as the maximum possible observation delay, Δ, is increased. The problems are categorized according to the value of $\{\mathcal{X}(0) + \mathcal{X}(1)\}$.

5. REFERENCES

[1] J.L. Bander and C.C. White. Markov decision processes with noise corrupted and delayed state observations. *Journal of OR Society*, 50:660–668, 1999.

[2] G. Gordon J. Pineau and S. Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *International Joint Conference in Artificial Intelligence*, 2003.

[3] K.V. Katsikopoulos and S. E. Engelbrecht. Markov decision processes with delays and asynchronous cost collection. *IEEE transactions on automatic control*, 48:568–574, 2003.

[4] R. Nair, D. Pynadath, M. Yokoo, M. Tambe, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, 2003.

[5] T. Smith and R. G. Simmons. Point-based pomdp algorithms: Improved analysis and implementation. In *UAI*, 2005.

[1]At execution time, we can receive observations at delays greater than D

Analysis of Methods for solving MDPs

(Extended Abstract)

Marek Grześ and Jesse Hoey
David R. Cheriton School of Computer Science, University of Waterloo, Canada
{mgrzes, jhoey}@cs.uwaterloo.ca

ABSTRACT

New proofs for two extensions to value iteration are derived when the type of initialisation of the value function is considered. Theoretical requirements that guarantee the convergence of backward value iteration and weaker requirements for the convergence of backups based on best actions only are identified. Experimental results show that standard value iteration performs significantly faster with simple extensions that are investigated in this work.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Experimentation, Theory

Keywords

Policy Iteration, Markov Decision Process, Value Iteration

1 INTRODUCTION

We consider the problem of finding an optimal policy in discrete time, finite state and action, discounted (by factor $\gamma < 1$) as well as undiscounted ($\gamma = 1$) Markov Decision Processes (MDPs) [6]. A standard MDP notation is used from [3]. The following definitions are considered:

DEFINITION 1. *Q is pessimistic if $Q(x, a) \leq Q^*(x, a)$ and optimistic if $Q(x, a) \geq Q^*(x, a)$.*

DEFINITION 2. *Q is monotone pessimistic if $Q(x, a) \leq R_x(a) + \gamma \sum_{x'} T_{x,a}(x')V(x')$ and is monotone optimistic if $Q(x, a) \geq R_x(a) + \gamma \sum_{x'} T_{x,a}(x')V(x')$ for all x and a, where $V(x) = \max_a Q(x, a)$.*

2 ANALYSIS

In our recent work [3], a new backup of the value function was proposed that exploits the idea of updating best actions only (BAO). The approach was shown to be very successful in PAC-MDP reinforcement learning that requires frequent replanning of a changing MDP. The current work investigates how this idea can help in general MDP planning where every MDP is solved once. We also show a new theorem which allows applying the BAO operator in a more general scenario:

THEOREM 1. *Planning based on backups that, in every state, keep updating all best actions until the Bellman error of best actions is smaller than ϵ (BAO) converges to the optimal value function when the initial value function is optimistic.*

Our recent work [3] has identified specific problems with the convergence of backward value iteration (BVI) [2]. Here, we show new, formal theoretical requirements that guarantee that backward value iteration will converge.

THEOREM 2. *In the backward value iteration algorithm specified in [2], the policy induced by the current value function is proper (i.e., every state reaches the goal state with probability 1 [1]) after every iteration when:*

1. *the initial value function is monotone pessimistic, i.e., the conditions of Definition 2 are satisfied*
2. *the initial policy is proper, i.e., at least one goal state is in the policy graph of each state*

When the policy induced by the current value function of the BVI algorithm is proper after every iteration, the algorithm will update all states in every iteration and upon termination the Bellman error satisfies the termination condition on all states.

3 RESULTS

In order to test BAO in general MDPs, the following algorithms are evaluated: (1) VI: standard Gauss-Seidel value iteration [1], (2) MPI(k): modified policy iteration [7] where k is the constant number of iterations in policy evaluation, (3) PI: policy iteration [4], and (4) PS: prioritised sweeping with priority based on the Bellman error [5]. If BAO is applicable, it is used as one of the options and added to the name of the algorithm in the results. Also, a simplified version of BAO is used, named BAOnce, that updates best actions only once during every visit to the state. $V(i)$, V_{max}, V_{min}, V^+, and V^- mean that the value function of a particular algorithm was initialised with i, $R_{max}/(1-\gamma)$, $R_{min}/(1-\gamma)$, and upper and lower bounds on V^* correspondingly. Every domain was evaluated 10 times, for every randomly generated domain 10 instances were generated, the precision ϵ was 10^{-5}, and the standard error of the mean is shown in the results which display the planing time and the number of performed backups (the best results are in boldface).

VI, by default, cannot beat PI/MPI on domains with a high number of actions. For this reason, the first set of domains is generated according to [6] and has a high number of actions: the number of states and actions in every state is 100, and an action can lead to three randomly selected states with a probability sampled from a truncated Gaussian distribution with mean 20 and standard deviation 5 or from a uniform distribution on [1-100].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Nr	Time [ms]		Backups		Algorithm
1	3869.5	± 159.0	7970000.0	± 332699	VI-V(0)
2	3780.1	± 172.2	7662000.0	± 367979	VI-Vmax
3	2546.5	± 127.2	5158000.0	± 251183	VI-V+
4	840.4	± 61.2	641943.6	± 41327	VI-Vmax-BAO
5	**104.1**	**± 3.9**	114576.1	± 4805	**VI-V+-BAO**
6	**91.3**	**± 2.8**	73694.2	± 2044	**VI-V+-BAOnce**
7	5569.2	± 143.0	6421040.0	± 177804	PS-V+
8	1907.7	± 78.3	94820.0	± 3445	MPI(2)-V(0)
9	441.5	± 20.6	99680.0	± 4283	MPI(10)-V(0)
10	238.9	± 10.8	97060.0	± 4028	MPI(20)-V(0)
11	**122.9**	**± 4.3**	255330.0	± 10614	**MPI(500)-V(0)**
12	136.5	± 5.4	309910.0	± 14962	PI-V(0)
13	1079.2	± 58.3	57700.0	± 2579	MPI(2)-V+
14	133.6	± 6.6	303910.0	± 16916	PI-V+

Table 1: Results on non-terminating MDPs, Gaussian rewards and $\gamma = 0.99$

Nr	Time [ms]		Backups		Algorithm
1	3545.9	± 147.0	7526000.0	± 310506	VI-V(0)
2	3024.4	± 127.4	6305000.0	± 255679	VI-Vmax
3	**170.9**	**± 4.6**	172349.5	± 5251	**VI-Vmax-BAO**
4	169.3	± 3.0	127090.0	± 2314	VI-Vmax-BAOnce
5	6958.2	± 142.7	7819750.0	± 155515	PS-Vmax
6	1963.9	± 72.2	96840.0	± 3460	MPI(2)-V(0)
7	431.8	± 14.2	98630.0	± 3279	MPI(10)-V(0)
8	250.6	± 6.8	102980.0	± 2862	MPI(20)-V(0)
9	**101.1**	**± 4.8**	209310.0	± 10885	**MPI(500)-V(0)**
10	**111.4**	**± 5.4**	251550.0	± 12444	**PI-V(0)**

Table 2: Results on non-terminating MDPs, uniformly distributed rewards and $\gamma = 0.99$

The first experiment evaluates domains with Gaussian reward (see Table 1). MPI improves its performance and gets closer to the performance of PI when k grows. All rewards are positive (and similar due to Gaussian distribution) here, and evaluation of every policy makes progress towards an optimal solution, and for that reason it makes sense to advance evaluation of every policy (high k) and do fewer policy updates - the situation where VI is poor. BAO with V_{max} is better than standard VI, but loses against MPI. Only a more informative initialisation, V^+, allowed BAO to be both faster and to reduce the number of backups beyond what was achieved by the best MPI settings. Certainly, one could argue that V^+ is usually not known exactly in the real situation, however sometimes (see the car replacement example below) a bound, far better than V_{max}, can be determined and the discussed experiment shows that such a bound would be very convenient for the BAO update.

Since BAO continuously adapts its evaluated policy, our guess was that it may waste time on evaluating all actions which are similar due to a low variance in the Gaussian rewards. Therefore, the same set of domains was generated with a uniform reward distribution. Results in Table 2 show the evidence that higher variance in values of rewards made BAO perform better even with uninformative V_{max} initialisation. Here, there are actions which are proved to be non-optimal initially and BAO can help.

Car replacement from [4] was evaluated as a realistic domain with many actions: there are 41 states and 41 actions. Results are in Table 3. $\gamma = 0.97$ since in [4], it is justified as having a real meaning of around 12% annual interest rate. Rewards have high variance, but this time there is another property that strongly influences the performance of evaluated algorithms. Specifically, a short horizon policy is very sub-optimal when compared with a long horizon policy. Actions that yield high instantaneous reward are sub-optimal in the long term (selling a good car now and buying a cheap one may result in getting money now but incurs losses in the long term). Hence, BAO first learns actions which seem promising in short term and then unlearns them. The same applies to MPI. Small k makes MPI slower. With sufficiently large k, policies are

Nr	Time [ms]		Backups		Algorithm
1	206.5	± 8.0	591880.1	± 24543	VI-Vmax
2	144.6	± 6.6	429999.8	± 21736	VI-V+
3	169.6	± 5.3	494214.0	± 15791	VI-V(0)
4	123.5	± 8.0	378729.3	± 21790	VI-V-
5	160.2	± 5.6	498248.4	± 18250	VI-Vmin
6	126.8	± 0.8	176371.1	± 592	VI-Vmax-BAO
7	**30.7**	**± 2.0**	46615.6	± 882	**VI-V+-BAO**
8	**55.9**	**± 1.1**	81765.3	± 1350	**VI-V(0)-BAO**
9	124.5	± 3.0	159412.1	± 494	VI-Vmax-BAOnce
10	**25.8**	**± 0.4**	36149.7	± 498	**VI-V+-BAOnce**
11	**48.8**	**± 0.8**	64849.7	± 281	**VI-V(0)-BAOnce**
12	397.1	± 3.1	734117.3	± 3216	PS-Vmax
13	279.6	± 1.1	540306.2	± 2237	PS-V+
14	314.4	± 1.4	596263.0	± 3923	PS-V(0)
15	226.8	± 1.8	447260.8	± 2255	PS-V-
16	277.7	± 1.6	537243.5	± 1959	PS-Vmin
17	**16.1**	**± 0.6**	18158.9	± 487	**MPI(20)-Vmax**
18	**13**	**± 0.2**	14559.1	± 292	**MPI(20)-V+**
19	**13.1**	**± 0.2**	14883	± 251	**MPI(20)-V(0)**
20	**13.7**	**± 0.4**	15293	± 360	**MPI(20)-V-**
21	**15.8**	**± 0.5**	17535.7	± 562	**MPI(20)-Vmin**
22	26.3	± 0.7	80097.6	± 2332	PI-Vmax
23	25.2	± 0.5	76264.1	± 1598	PI-V+
24	26.6	± 0.8	81413.7	± 2403	PI-V(0)
25	25.8	± 0.9	77067.7	± 3203	PI-V-
26	27.8	± 0.4	84660.9	± 1870	PI-Vmin

Table 3: Results on car replacement

evaluated 'almost exactly', and this helps avoiding short horizon policies. This also explains why MPI with lowest k is even slower than BAO because MPI applies full backups during policy improvement. $V(0)$ could be used to initialise the value function in BAO because in this domain there is never a positive long term reward (the possession of a car always incurs costs). With this knowledge, BAO can be competitive even on this challenging domain. If the bound can be improved, BAO gains further speed-up. Thus, $V(0)$, V_{max}, and V^+ yields optimistic initialisation required by BAO, and V_{min} and V^- pessimistic which was originally required by the theory of MPI [7], however the recent literature shows that this requirement can be avoided [1].

4 CONCLUSION

Our experiments have shown that, thanks to BAO updates, the gap between MPI and VI is significantly reduced on challenging domains with many actions. Unpublished comparisons with BVI on stochastic shortest path problems showed that standard VI can also outperform prioritised approaches when BAO is used.

Acknowledgement This research was sponsored by American Alzheimer's Association through the Everyday Technologies for Alzheimer's Care (ETAC) program. The first author was supported by a fellowship from the Ontario Ministry of Research and Innovation.

5 REFERENCES

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control (2 Vol Set)*. Athena Scientific, 3rd ed., 2007.

[2] P. Dai and E. A. Hansen. Prioritizing Bellman backups without a priority queue. In *Proc. of ICAPS*, 2007.

[3] M. Grześ and J. Hoey. Efficient planning in R-max. In *Proc. of AAMAS*, 2011.

[4] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, 1960.

[5] A. W. Moore and C. G. Atkenson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13:103–130, 1993.

[6] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[7] M. L. Puterman and M. C. Shin. Modified policy iteration algorithms for discounted Markov decision problems. *Management Science*, 24:1127–1137, 1978.

Decentralized Multi-agent Plan Repair in Dynamic Environments*

(Extended Abstract)

Antonín Komenda Peter Novák Michal Pěchouček

Agent Technology Center,
Department of Computer Science and Engineering,
Czech Technical University in Prague
{antonin.komenda,peter.novak,michal.pechoucek}@agents.fel.cvut.cz

ABSTRACT

Achieving joint objectives by teams of cooperative planning agents requires significant coordination and communication efforts. For a single-agent system facing a plan failure in a dynamic environment, arguably, attempts to repair the failed plan in general do not straightforwardly bring any benefit in terms of time complexity. However, in multi-agent settings the communication complexity might be of a much higher importance, possibly a high communication overhead might be even prohibitive in certain domains. We hypothesize that in decentralized systems, where coordination is enforced to achieve joint objectives, *attempts to repair failed multi-agent plans should lead to lower communication overhead than replanning from scratch*.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Plan execution, formation, and generation*

General Terms

Algorithms, Experimentation.

Keywords

multi-agent plan repair, decentralized multi-agent planning, communication complexity.

1. MOTIVATION

When an agent is situated in a dynamic environment, occurrence of various unexpected events the environment generates might lead to plan invalidation, a failure. A straightforward solution to this problem is to invoke a planning algorithm and compute a new plan from the state the agent found itself in after the failure to a state conforming with its original objective. In many cases, however, a relatively minor fix to the original plan would resolve the failure, possibly at a lower cost.

*An extended version of this paper was also published as [1].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

In general, plan repair can be seen as planning with re-use of fragments of the old plan. Even though there is a body of research empirically demonstrating that plan repair in some domains performs better than replanning (e.g., [2]), theoretical analysis concluded concluded that plan re-use (repair) in general does not bring any benefit over replanning in terms of computational time complexity [3]. In situated multi-agent systems often it is not the time complexity which is of a primary importance, the *communication complexity* is often a higher priority concern (consider application domains, such as e.g., undersea operations, where the communication links are extremely constrained and expensive). The motivation for our research is the intuition that multi-agent plan repair, even though not always the fastest approach, should under specific conditions generate lower communication overheads in comparison to replanning.

2. MULTI-AGENT PLAN REPAIR

We consider a number of *cooperative* and *coordinated* actors featuring possibly distinct sets of capabilities (actions), which concurrently plan and subsequently execute their local plans so that they achieve a joint goal. An instance of a multi-agent planning problem is defined by: i) an environment characterized by a state space, ii) a finite set of agents, each characterized by a set of primitive actions (or capabilities) it can execute in the environment, iii) an initial state the agents start their activities in and iv) a characterization of the desired goal states. Definitions of the underlying formal framework can be found in Nissim et al. [4]. The core hypothesis of the paper can be then formulated as follows: *multi-agent plan repair approaches producing more preserving repairs than replanning tend to generate lower communication overhead for tightly coupled multi-agent problems.*

We propose three algorithms for solving the plan repair problem. The core idea behind the *back-on-track* (BoT) algorithm is to utilize a multi-agent planner to produce a plan from the failed state to the originally desired state and subsequently follow the rest of the original multi-agent plan from the step in which the failure occurred. In result, the BoT repair tries to preserve a suffix of the original plan and prefix it with a newly computed plan starting in the failure state and leading to some state along the execution of the original plan in the ideal environment.

The second approach, *lazy-repair (LR)*, is designed to to preserve an *executable remainder* of the original multi-agent plan (the actions would remain, if the original plan was executed ignoring non executable actions) and close the gap between the state resulting from the failed plan execution and a goal state of the original planning problem. The lazy approach tries to preserve a partial prefix of the original plan and complete it by a newly planned plan suffix. The algorithm is incomplete, as it might happen that the execution of the executable remainder diverges to a state from which no plan to some goal state exists.

The shortcoming of the LR algorithm is addressed by the *repeated lazy repair (RL).* The idea is that a failure during execution of an already repaired plan makes the previous repair irrelevant and its result can be discarded, unless the failure occurred already in the fragment appended by the previous repair. Note, the repeated lazy repair algorithms enables a plan execution model which preserves significantly longer fragments of the original plan. That is, upon a failure, instead of trying to repair the failed plan directly, as the previous two algorithms, the system can simply proceed with execution of the remainder of the original plan and only after its complete execution the lazy plan repair is triggered. The approach simply ignores the plan failures during the multi-agent plan execution and postpones the repair to the very end of the process, hence the "*lazy*" label for the two algorithms.

3. EXPERIMENTAL VALIDATION

To verify the core hypothesis, we conducted a series of experiments with implementations of the proposed multi-agent plan repair algorithms. Firstly, a multi-agent plan was computed by a distributed multi-agent planner authored by Nissim et al. [4]. Secondly, we executed the multi-agent plan. In the course of the plan execution, we simulated the environment dynamics by producing various plan failures according to a variable failure probability P (with a uniform distribution). The plan execution was monitored and upon a failure detection a plan repair algorithm was invoked. Before execution of each plan step, the joint action of all actions of the particular agents is checked for applicability in the current state. In the case it is not applicable, a plan repair algorithm is invoked and the execution continues on the repaired plan.

We distinguished two types of plan failures: *action failures* and *state perturbations*. An *action failure* is simulated by omitting a randomly chosen (with a uniform distribution) individual agent action from the actual plan step. The other simulated failure type, *state perturbation,* is parametrized by a positive non-zero integer c, which determines the number of randomly chosen (again with a uniform distribution) state terms, which are removed from the current state, as well as the number of terms which are added to it.

The experiments were conducted on three planning domains originating in the standard benchmark single-agent ICP planning domains. Similarly to [4], we chose domains, which are straightforwardly modifiable to multi-agent planning problems: LOGISTICS (3 agents), ROVERS (3 agents), and SATELLITES (2–5 agents). The metrics were i) *execution length* (number of joint actions executed), ii) *planning time* (cumulative time consumed by the underlying planner), iii) *communication* (number of messages passed between the agents).

4. RESULTS AND FINAL REMARKS

The first batch of experiments directly targets validation of the core hypothesis. We used LOGISTICS as a tightly coupled domain (the resulting personal plans often depend on each other) and dynamics of the simulated environment modeled as action failures. Figure 1 depicts the results of the experiment, which *support* our hypothesis. The overall planning time was at 54% (34% at best) and at 51% (12% at best) for Back-on-Track-Repair and Repeated-Lazy-Repair against replanning respectively. The execution length was lower being in average 96% (72% at best, 130% at worst) by Back-on-Track-Repair and significantly lower being 81% (34% at best, 132% at worst) for Repeated-Lazy-Repair.

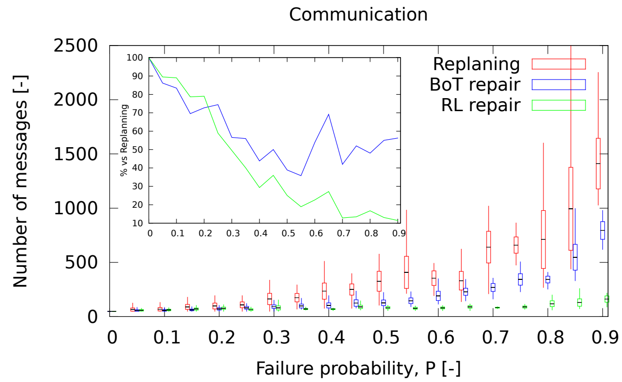


Figure 1: Experimental results for LOGISTICS domain with 3 agents and action failures.

The second batch of experiments focused on boundaries of validity of the positive result presented above, i.e., *with decreasing coupling of the domain, the communication efficiency gains of repairing techniques should decrease*. Experiments performed with the loosely-coupled domain of ROVERS support the claim.

The third batch of experiments targeted the perturbation magnitude of the plan failures, i.e., *communication efficiency gain of plan repairing should decrease as the difference between a nominal and related failed state increases*. The underlying intuition that in the case the dynamic environment generates only relatively small state perturbations and the failed states are "not far" from the actual state was positively supported by results of another LOGISTICS experiment employing state perturbations as the model of the environment dynamics ($c = 1$).

Finally, we conducted a series of experiments with an uncoupled SATELLITES domain. The results show the anticipated lower plan repair communication efficiency in contrast to replanning.

The main difference between our approach and the related work (partial ordered plan monitoring and repairing, conformant and contingency planning, plan re-use and plan adaptation, and finally Markov decision processes) is that the state perturbations utilized in our experiments have *a priori* unknown probabilities.

Acknowledgements

This work was supported by the *U.S. Army CERDEC* grant W911NF-11-1-0252 and by the *Grant Agency of the Czech Technical University in Prague* grant SGS10/189/OHK3/2T/13 .

5. REFERENCES

[1] Antonín Komenda, Peter Novák, and Michal Pěchouček. Decentralized Multi-agent Plan Repair in Dynamic Environments. *CoRR*, abs/1202.2773, February 2012.

[2] Roman van der Krogt and Mathijs de Weerdt. Self-interested planning agents using plan repair. In *Proceedings of the ICAPS 2005 Workshop on Multiagent Planning and Scheduling*, pages 36–44, 2005.

[3] B. Nebel and Koehler J. Plan reuse versus plan generation: a theoretical and empirical analysis. *Artificial Intelligence*, 76(1-2):427–454, July 1995.

[4] Raz Nissim, Ronen I. Brafman, and Carmel Domshlak. A general, fully distributed multi-agent planning algorithm. In *Proceedings of AAMAS'10*, pages 1323–1330, 2010.

Multimodal Trust Formation with Uninformed Cognitive Maps (UnCM)

(Extended Abstract)

Michele Piunti
Reply Whitehall
Rome, Italy
m.piunti@reply.it

Matteo Venanzi
AIC Group, Electronics and Computer Science
University of Southampton
m.venanzi@ecs.soton.ac.uk

Rino Falcone, Cristiano Castelfranchi
Institute of Cognitive Sciences and Technologies, ISTC-CNR
{r.falcone,c.castelfranchi}@istc.cnr.it

ABSTRACT

This work describes a cognitive heuristic allowing agents to assess trust and delegations merging heterogenous information sources. The model is realized through Uninformed Cognitive Maps, based on the combination of: (*i*) categorization abilities (*ii*) history of personal experiences (*iii*) context awareness.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, multiagent systems*

General Terms

Algorithms, Performance, Design, Theory

Keywords

Cognitive models, Trust, Social Systems, Social simulation

1. MULTIMODAL TRUST FORMATION

Crucial abilities for agents engaged in open systems is to decide how to coordinate activities and whether (or not) delegate tasks to other, possibly unknown, agents. Trust based interactions have been proposed as a suitable model to achieve such a subjective coordination. But, placed in the context of open and dynamic systems, the main issue of trust management is a problem *trust formation*. Existing approaches to trust formation refer to *subjective experiences* and *reputation* mainly. Subjective experiences are typically exploited in evaluating the outcomes of previous transactions, and therefore they are limited by the need of multiple and repeated interactions between the same agents. Reputational approaches have been proposed to establish trustworthy interactions with possibly unknown counterparts [7, 5]. The downside is the need of a network of reputation providers, being each reputational information possibly biased or corrupted. Other approaches push on the multi-faceted relationship between environments, context awareness and trust management. Finally, the relevance of categories for trusting strangers has been remarked in the work of Falcone et al [4]. Categorial reasoning for trust formation has also been recently explored by Burnett et al. [2]. In their work, they propose the notion of stereotypical trust (stereotrust) as a categorial prejudice that agents may take into account in order to assess trust *in absence* of direct evidences. The mechanism adopts data mining techniques applied over the database of past transactions.

The approach proposed in this research aims at combining three different information sources into a unique reasoning process. Multimodal trust formation is realized through a novel mechanism called *Uninformed Cognitive Map* (UnCM), where the introduction of learning mechanisms further allows to establish a series of emergent relations between a rich set of information sources and the trustworthiness of unknown trustees. In doing so, we rely on the socio cognitive theory of trust [3], according to which trust is grounded on detectable cognitive ingredients.

2. UNINFORMED COGNITIVE MAPS

In cognitive agents, the problem of trust formation can be translated in the problem of retrieving the constituent beliefs of trust. Cognitive trust is treated as a relational construct between a trustor (trust giver, ag_i) and a trustee (trust receiver, ag_j) which can be established in a given environment/context E, and about a defined task to be fulfilled (τ): $Trust(ag_i, ag_j, E, \tau)$. Trust is then *graded* over multiple dimensions. The degree of trust (DoT) comes from a series of cognitive primitives, which can be summarized in terms of trustor's beliefs and goals. The approach takes into account the three contributions that play a crucial role in trust formation: *Bel*($Can_{ag_j}(\tau)$), that is trustor believes that ag_j is potentially able to fulfill τ (i.e., ag_j has the skills, the competences, the necessary instruments for realizing that task τ); *Bel*($Will_{ag_j}(\tau)$), that is trustor believes that ag_j is potentially willing and persistent in fulfilling τ (i.e., ag_j has the motivational attitudes sufficient to perform the task τ); *Bel*($ExtFact_{ag_j}(\tau)$), that is trustor believes that the external conditions are not preventing the execution of τ by ag_j (or even: ag_i believes that ag_j will perform the task τ in an environment presenting positive or negative interferences to ag_j's behavior in order to achieve the task τ). Summing up, an agent ag_i trusts ag_j about a task τ and in the conditions E, if ag_i's DoT overcomes a given threshold σ: $DoT_{ag_j,E,\tau} > \sigma$. The model resembles the notion of *Krypta* and *Manifesta*, according to which agents' manifesta are signals, or observable traces, recalling agents' krypta, which are the internal properties (*qualities*, *abilities* or *powers*) finally determining agents' behaviors on specific tasks

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

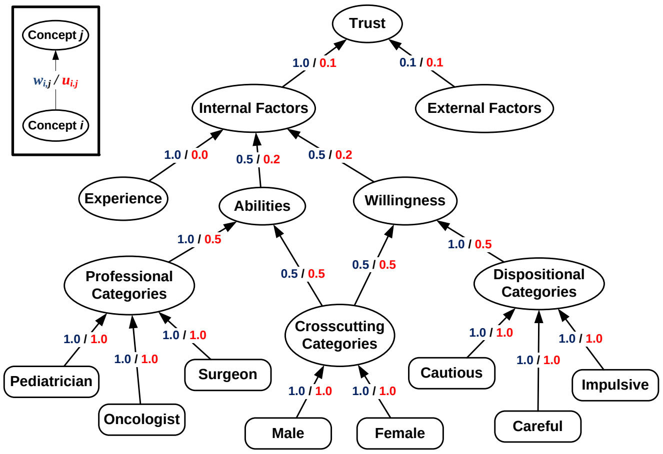


Figure 1: UnCM implementing the socio-cognitive trust model with multiple dimensions.

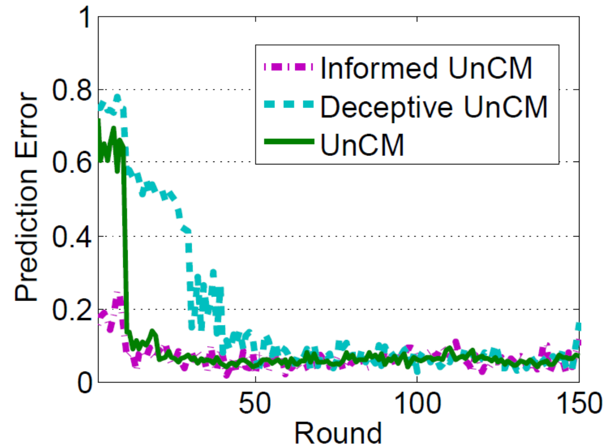


Figure 2: Plot of the prediction error for the UnCM, Informed UnCM, Deceptive UnCM over 150 simulation rounds.

or contexts [1].

Uninformed Cognitive Maps (UnCM) are a novel approach hybridizing cognitive modeling and learning. They are based as an extension of Fuzzy Cognitive Maps, a computing technique successfully applied in several domains for modeling knowledge-based systems [6]. An UnCM is as a graph modeling causal processes by means of concepts and causal relations placed on different dimensions (Fig. 1). The UnCM layout is designed by domain experts using an off-line setting. At design time, the relevant concepts of a problem domain are identified, and their reciprocal influences are quantitatively modeled by weighted connections. The causal impact between two concepts A_i and A_j is then measured by the weight of the connection $w_{i,j}$, taken in the interval $[-1, 1]$.

3. EXPERIMENTAL EVALUATION

Evaluation concerned a simulated agent society in a medical domain, with 100 trustees having krypta randomly selected from a repository of 2500 profiles. Every profile is characterized by three types of categories, which can be of *professional, dispositional* or *crosscutting*. The experiment discussed here used the *pneumonia* task, for which the best categorial profile is assumed to be $\langle$*pediatrician, cautious, female*$\rangle$. The outcome of trustee execution is referred in terms of *score*, while the accuracy of trust formation is measured in terms of *prediction error* as the distance of the predicted DoT from the real delegation outcome: $error = |DoT_{ag_j} - score|$. Setting also takes into account the environmental influences, defined as a ρ parameter indicating the contribution of situated conditions to the executor's performance. Hence, each task execution may receive an influence randomly distributed in the range $[-\rho, +\rho]$ System openness is determined by the parameter δ, which determines the number of trustees replaced at each round. Finally, L sets the interval rounds after which the trustors update their learning model over the experiences history. The model has beens compared with well-established approaches to data analysis and decision making, as neural networks (Neural agents) and agents using stereotypes and data mining mechanisms (Stereotrust agents). Experiments pointed out the abilities of UnCM strategies to perform task delegation based on multimodal trust attribution. Either context awareness and experiences play a pivotal role in trust formation in open and dynamic systems. The adopted UnCM, in particular, allows to learn to which extent the single categories fit for a given tasks, thus drastically enhancing delegation-making.Fig. 2 shows the performance of UnCM in minimizing errors: experiments show that categorial evidences *emerge* with respect to the ongoing tasks—also *without* requiring any initial categorial knowledge. The mechanism manages in a unique function heterogeneous information sources, ranging from personal experiences, to manifesta and external influences. Thanks to the UnCM learning algorithm, categories are revised, or devised from scratch, and the categorial information is combined to personal experiences and environmental conditions encountered. Differently from Neural and Stereotrust agents , the UnCM agents are also able to maintain a meaningful *semantic* of influences between concepts and their connections. Influences of the single categories on a given task represent a key aspect and, using UnCM this information is explicitly readable and updated *online*. Limitations of the current approach pave the way to future work. To evaluate the scalability of the proposed approach, applications in different domain as social networks will be devised.

4. REFERENCES

[1] M. Bacharach and D. Gambetta. Trust as Type Detection. In *Trust and deception in virtual societies*, 2001.

[2] C. Burnett, T. Norman, and K. Sycara. Bootstrapping Trust Evaluations through Stereotypes. In *Autonomous Agents and Multiagent Systems (AAMAS-10)*, pages 241–248, 2010.

[3] C. Castelfranchi and R. Falcone. *Trust Theory. A Socio-Cognitive and Computational Model*. Wiley, 2010.

[4] R. Falcone, M. Piunti, M. Venanzi, and C. Castelfranchi. From manifesta to krypta: The relevance of categories for trusting others. *ACM Transactions on Intelligent Systems and Technology.*, 2011.

[5] T. G. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated Trust and Reputation model for Open Multi-Agent Systems. *Journal of Autonomous Agent and Multi-Agent Systems*, 13:119–154, 2006.

[6] B. Kosko and J. Burgess. Neural Networks and Fuzzy Systems. *The Journal of the Acoustical Society of America*, 103:3131, 1998.

[7] B. Yu and M. P. Singh. An evidential model of distributed reputation management. In *Autonomous Agents and Multiagent Systems (AAMAS-02)*, pages 294–301, 2002.

Modeling Deep Strategic Reasoning by Humans in Competitive Games

(Extended Abstract)

Xia Qu
Dept. of Computer Science
University of Georgia
Athens, GA 30602
quxia@uga.edu

Prashant Doshi
Dept. of Computer Science
University of Georgia
Athens, GA 30602
pdoshi@cs.uga.edu

Adam Goodie
Dept. of Psychology
University of Georgia
Athens, GA 30602
goodie@uga.edu

ABSTRACT

The prior literature on strategic reasoning by humans of the sort, *what do you think that I think that you think*, is that humans generally do not reason beyond a single level. However, recent evidence suggests that if the games are made competitive and therefore representationally simpler, humans generally exhibited behavior that was more consistent with deeper levels of recursive reasoning. We seek to computationally model behavioral data that is consistent with deep recursive reasoning in competitive games. We use generative, process models built from agent frameworks that simulate the observed data well and also exhibit psychological intuition.

Categories and Subject Descriptors

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

General Terms

Experimentation, Performance

Keywords

recursive reasoning, human decision making, modeling, games

1. INTRODUCTION

We model human judgment and behavioral data, reported by Goodie et al. [4], that is consistent with *three* levels of recursive reasoning in the context of fixed-sum games. In doing so, we investigate principled modeling of behavioral data consistent with levels rarely observed before. A previous model utilized underweighted belief learning, parameterized by γ, and a quantal response choice model [5] for the subject agent, parameterized by λ, within the framework of interactive partially observable Markov decision process (I-POMDP) [3]. We extend this model to make it applicable to games evaluating up to level 3 reasoning. Although it employs an empirically supported choice model for the subject agent, it does not ascribe plausible choice models to the opponent who in the experiments is also projected as being human. We hypothesize that an informed choice model for the opponent supports more nuanced explanations for observed opponent actions leading to improved performance. Hence, our second candidate model generalizes the previous by intuitively utilizing a quantal response choice model for selecting the opponent's actions at level 2. Finally, our third candidate model deviates from using I-POMDPs by utilizing weighted fictitious play [1], which predominantly relies on the past pattern of the opponent's observed actions to form a judgment about what the opponent will do next. This model differs from the previous two in that it does not seek to ascertain the mental models of the opponent but instead bases itself on the observed frequency of empirical play. The strictly competitive nature of the game discourages the influence of essentially cooperative social constructs such as positive reciprocity and altruism, otherwise observed in strategic games. While other processes such as inequality aversion may apply, an analysis of the data reveals that it did not play a role here.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. COMPUTATIONAL MODELING

In order to computationally model the data, a multiagent decision making framework that integrates recursive reasoning in the decision process is needed. A finitely-nested I-POMDP$_{i,l}$ [3] for agent i with a strategy level l represents a choice which meets the requirements of explicit consideration of recursive beliefs and decision making based on such beliefs.

Because the opponent is thought to be human and guided by payoffs, we focus on intentional models only. Given that expectations about the opponent's action by the participants showed consistency with the opponent types used in the experimentation, intuitively, model set, $\Theta_j = \{\theta_{j,0}, \theta_{j,1}, \theta_{j,2}\}$, where $\theta_{j,0}$ is the level 0 (*myopic*) model of the opponent, $\theta_{j,1}$ is the level 1 (*predictive*) model and $\theta_{j,2}$ is the level 2 (*super-predictive*) model. Parameters of these models are analogous to the I-POMDP for agent i.

We observed that some of the participants learn about the opponent model as they continue to play. However, in general, the rate of learning is slow. This is indicative of the cognitive phenomenon that the participants could be underweighting the evidence that they observe. We may model this by augmenting normative Bayesian learning in the following way:

$$b'_{i,l}(s, \theta_{j,l-1}|o_i; \gamma) = \alpha\, b_{i,l}(s, \theta_{j,l-1}) \Big\{ \sum_{a_j} O_i(o_i|a_i, a_j, s') \times Pr(a_j|\theta_{j,l-1}) \Big\}^{\gamma} \tag{1}$$

where α is the normalization factor, state s corresponds to A and s' to B, action a_i is to move, and if $\gamma < 1$, then the evidence $o_i \in \Omega_i$ is underweighted while updating the belief over j's models. Furthermore, we observed significant rationality errors in the participants' decision making. We utilize the *quantal response* model [5] to simulate human non-normative choice. This model is based on the finding that rather than always choosing the optimal action which maximizes the expected utility, individuals are known to select actions proportionally to their utilities. The quantal response model assigns a probability of choosing an action as a sigmoidal function of how close to optimal is the action. Previously, Doshi et al. [2] augmented I-POMDPs with both these models in order to simulate human recursive reasoning up to level two. As they continue to apply to our data, we extend the I-POMDP model to the longer games and label it as I-POMDP$_{i,3}^{\gamma,\lambda}$.

The methodology for the experiments reveals that the participants are deceived into thinking that the opponent is human. *Therefore, participants may justify unexpected actions of the opponent as errors in their decision making rather than due to their level of reasoning.* Hence, we generalize the previous model by attributing quantal response choice to opponent's action selection as well. Let λ_1 be the quantal response parameter for the participant and λ_2 be the parameter for the opponent's action. Then,

$$Q(a_i^*;\gamma,\lambda_1,\lambda_2) = \frac{e^{\lambda_1 \cdot U(b'_{i,3},a_i^*;\gamma,\lambda_2)}}{\sum_{a_i \in A_i} e^{\lambda_1 \cdot U(b'_{i,3},a_i;\gamma,\lambda_2)}} \tag{2}$$

parameters, λ_1, $\lambda_2 \in [-\infty,\infty]$; a_i^* is the participant's action and $Q(a_i^*)$ is the probability assigned by the model. $U(b'_{i,3},a_i;\gamma,\lambda_2)$ is the utility for i on performing action, a_i, given its updated belief, $b'_{i,3}$, with λ_2 parameterizing j's action probabilities, $Pr(a_j|\theta_{j,l-1})$, present in Eq. 1 and in computation of the utility. We label this model as I-POMDP$_{i,3}^{\gamma,\lambda_1,\lambda_2}$.

A different reason for participant behavior that relies more heavily on past patterns of observed actions of the opponent, instead of ascertaining the mental models of the opponent as in the previous I-POMDP based models, is applicable. A well-known learning model in this regard is weighted (generalized) fictitious play [1]. Let $E_i(a_j)$ be the observed frequency of opponent's action, $a_j \in A_j$. We update this as:

$$E_i^t(a_j;\phi) = I(a_j,o_i) + \phi\, E_i^{t-1}(a_j) \quad t = 1,2,\ldots \tag{3}$$

where parameter, $\phi \in [0,1]$, is the weight put on the past observations; $I(a_j,o_i)$ is an indicator function that is 1 when j's action in consideration is identical to the currently observed j's action, o_i, and 0 otherwise. Due to the presence of rationality errors in the data, we combine the belief update of Eq. 3 with quantal response. We label this model as wFP$_i^{\phi,\lambda}$.

3. EVALUATION

To learn λ_2 in I-POMDP$_{i,3}^{\gamma,\lambda_1,\lambda_2}$, we use the expectations data of the "catch" games only. These are games in which no matter the type of the opponent, the rational action for the opponent is to move. Hence, expectations of opponent staying by the participants in the catch trials would signal non-normative action being attributed. This also permits learning a single λ_2 value across the three opponent types. However, this is not the case for the other parameters: for different opponent types, the learning rate is different. Also, we observed that the rationality errors differ considerably between the participant groups experiencing different opponent types. Therefore, we learn parameters, γ and λ_1 given the value of λ_2 (and λ in I-POMDP$_{i,3}^{\gamma,\lambda}$), separately from each group's diagnostic games. Analogously, we learn ϕ and λ for wFP$_i^{\phi,\lambda}$ from the diagnostic games as well. We report the learned parameters in Table 1.

We utilize the learned values in Table 1 to parameterize the underweighting and quantal responses within the I-POMDP based models and fictitious play. We cross-validated the models on the

model	param.	myopic	pred	super-pred
I-POMDP$_{i,3}^{\gamma,\lambda_1,\lambda_2}$	λ_2	1.959		
	γ	0.164	0.049	0.221
	λ_1	3.259	3.906	3.768
I-POMDP$_{i,3}^{\gamma,\lambda}$	γ	0.232	0.079	0.357
	λ	2.985	3.826	3.667
wFP$_i^{\phi,\lambda}$	ϕ	0.999	0.999	0.150
	λ	2.127	3.107	3.165

Table 1: Average group-level parameter values learned from the training folds of the experiment data for the three candidate models.

Opponent type	Mean Squared Error (MSE)			
	Achievement score		Prediction score	
	myopic	super-pred	myopic	super-pred
Random	0.0041	0.4502	0.0035	0.3807
I-POMDP$_{i,3}^{\gamma,\lambda_1,\lambda_2}$	0.0014	0.0009	0.0020	0.0010
I-POMDP$_{i,3}^{\gamma,\lambda}$	0.0025	0.0008	0.0016	0.0014
wFP$_i^{\phi,\lambda}$	0.0123	0.0082	0.0103	0.0120

Table 2: MSE of the predictions by the different models.

test folds. Using a participant's actions in the first 5 trials, we initialized the prior belief distribution over the opponent types. We measure the goodness of the fit by computing the mean squared error (MSE) of the prediction by the models, and compare it to those of a random model (null hypothesis) for significance. We show the MSE in the achievement and prediction scores, as defined in Goodie et al. [4], based on the models in Table 2.

Notice from Table 2 that both I-POMDP based models have MSEs that are significantly lower than the random model. The difference in MSE of the achievement score for the myopic group between the two is significant (Student's paired t-test: p = .015). However, other MSE differences between the two models are insignificant and do not distinguish one model over the other across the scores and groups. Although attributing non-normative action selection to the opponent did not result in significantly more accurate expectations for any group, we think that it allowed the model to generate actions for agent i that fit the data better by supporting an additional account of j's (surprising) myopic behavior. Of course, this positive result should be placed in the context of increased expense of learning an additional parameter, λ_2. Large MSE of wFP$_i^{\phi,\lambda}$ reflects its weak simulation performance although it does improve on the par set by random for the super-predictive group.

Acknowledgment We acknowledge grant support from NSF, CAREER #IIS-0845036, and from USAF, #FA9550-08-1-0429.

4. REFERENCES

[1] Y.-W. Cheung and D. Friedman. Individual learning in normal-form games. *Games and Econ Behav*, 19:46–76, 1997.

[2] P. Doshi, X. Qu, A. Goodie, and D. Young. Modeling recursive reasoning in humans using empirically informed interactive POMDPs. In *AAMAS*, pages 1223–1230, 2010.

[3] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *JAIR*, 24:49–79, 2005.

[4] A. S. Goodie, P. Doshi, and D. L. Young. Levels of theory-of-mind reasoning in competitive games. *Behav Dec Making*, 24:95–108, 2012.

[5] R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and Econ Behav*, 10:6–38, 1995.

Coalitional Agency and Evidence-Based Ability

(Extended Abstract)

Nicolas Troquard*
Laboratory for Applied Ontology (ISTC-CNR)
Trento, Italy
troquard@loa.istc.cnr.it

ABSTRACT

The logics of "bringing it about" have been part of a prominent tradition for the formalization of individual and institutional agency. Our objective here is to extend Elgesem's logic of individual agency and ability to coalitions.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.2.4 [**Knowledge representation formalisms and methods**]: Modal logic

General Terms

Theory

Keywords

logic, bringing-it-about, coalitions, agency, ability

1. EXTENDED ABSTRACT

This extended abstract aims to contribute to the literature that views an action as the mere result of the activity of an agent. It is generally acknowledged that this tradition dates back at least to St. Anselm who claimed that the phenomenon of an action is better explained by what is brought about. This is to be distinguished from other traditions of logic of action talking explicitly about action terms: for instance, Dynamic Logics in computer science, or the study of action sentences in philosophy using first-order theories.

Pörn, Elgesem ([1]), and others, have studied the modality of agency in the Anselmian tradition. The bringing-it-about modality E_x has been quite popular in the MAS community. (E.g., [4]). It has been used to model the actions and responsibilities of acting entities x: the formula $E_x\varphi$ traditionally reads "x brings it about that φ". In the literature, x has been either an individual agent, or an institutional agent. An institution can involve several agents, each playing a specific role in it. But institutions are not groups or coalitions. Our contribution is an extrapolation of a theory of coalitional agency and ability from Elgesem's account of individual agency. That is, we study the logic of the operator E_x where x is a set of agents, along with a coalitional operator of ability.

Individual agency and ability. Elgesem's logic was a fresh look at a long tradition of philosophical logic of action, where the traditional modality of bringing-it-about is studied alongside related modalities of action. The logic still admits the core principles that are generally assumed for agency:

- all substitution instances of classical tautologies
- $\vdash \neg E_x\top$
- $\vdash E_x\varphi \wedge E_x\psi \rightarrow E_x(\varphi \wedge \psi)$
- $\vdash E_x\varphi \rightarrow \varphi$
- if $\vdash \varphi \leftrightarrow \psi$ then $\vdash E_x\varphi \leftrightarrow E_x\psi$

Following Sommerhoff, Elgesem argues that agency is the actual bringing about of a goal towards which an activity is oriented. An agent acts to achieve a goal. But an agent is not necessarily aware of his goals, at least not in the sense that he is consciously committed to achieve them. Elgesem also leans on Frankfurt according to whom, the pertinent aspect of agency is the manifestation of the agent's guidance towards a goal; not necessarily the intentional action. Here, we understand intention in agency as a motivated goal, possibly long pondered and rational. Elgesem seeks a more general notion of goal that guides agency.

He observes that the manifestation of guidance is the exercise of a power to bring about something. Therefore, the notion of potential guidance, or ability, of an agent for a goal should be integrated in a theory of agency. Elgesem argues, much contradicting Kenny, that we should not deny the possibility of abilities that are exercised only once, giving the example of Bob Beamon, who jumped 8.90 m (long jump) in the 1968 Olympics. If Beamon jumped that far it is that he was exercising guidance towards a goal. Even though this goal was probably not intentionally to jump 8.90 m, we would not take back from Beamon that on that day he brought about the fact that he jumped that far and that he had the ability to do it.

Elgesem then suggests that there is a more basic notion of ability than an intention-based one, and that this non-intentional notion of ability is a necessary condition for agency. By bringing about something, an agent *shows* that he is indeed able to do so. We advance an interpretation of *evidence-based* ability.

Governatori and Rotolo proved the completeness of Elgesem's logic ([2]). The principles are the following, where $C_x\varphi$ reads "the acting entity x is able to bring about φ".

*Supported by a Marie Curie fellowship.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

- all the previous principles
- $\vdash \neg C_x \bot$
- $\vdash \neg C_x \top$
- $\vdash E_x\varphi \rightarrow C_x\varphi$
- if $\vdash \varphi \leftrightarrow \psi$ then $\vdash C_x\varphi \leftrightarrow C_x\psi$

The logic of C_x is then rather weak. The only certainty one can have about the presence of an ability to bring about φ is in the presence of an actual bringing about of φ.

The notion of ability captured by Elgesem is nevertheless very appealing because it is one where *the observation of an evidence* induces the existence of an ability. Imagine a repository of web services that are acting in some ways upon their environment and can be queried. Whenever a request is successfully fulfilled, the ability of a service for a particular query can be logged and the couple service/query can be offered as a *suggestion* for later use. This *evidence-based* perspective of ability is strikingly weak in the individual case. Nevertheless, we will see that extending the logic to coalitions can offer more flexibility for the suggestion of potentially successful acting entities, even for *complex goals* that have never been brought about.

Joint actions. We will identify a group with an arbitrary subset of agents. Joint actions are a species of actions involving a group that acts towards a shared goal. Despite resorting to some notion of shared goal, Miller ([3]) argues that we-intentions are not a necessary element of joint actions. When two scholars start chatting at a conference break and somewhat start to take a walk in the park, they respect their turn in the conversation, they synchronize their pace, and take a direction in the park without having previously agreed on it. Similar to the individual case (Beamon's jump), this suggests that there is a more basic notion of coalitional goal-directed agency than an intentional one. Again in analogy with the individual case, that means that there is a basic notion of coalitional ability that is a necessary condition for coalitional agency. In particular, at a given time and from the evidence of actual agency of some coalitions for some goals, we will be able to infer the potential ability of larger coalitions for more complex goals. To come back to our example of web services, this suggests an incremental procedure for web service discovery. This evidence-based perspective may actually provide a practical alternative to the computationally costly orchestration procedures in web service composition.

Since there is a basic notion of coalitional agency, like Elgesem for individual agency and ability, we can therefore focus on the principles of pure agency and ability without having to struggle with the formation of we-intentions.

Empty coalition. We first look at the empty group that is the simplest group, though degenerate. Our notion of agency is one that is goal-directed, and our notion of ability is one of potential guidance towards a goal. It would not be right to give to the empty group a status of true coalition with a goal and a potential guidance for it. Hence

$$\vdash \neg C_\emptyset\varphi.$$

Together with the principle $\vdash E_x\varphi \rightarrow C_x\varphi$ adopted above, it follows from it that $\vdash \neg E_\emptyset\varphi$, too.

Evidence of coalitional ability. If a coalition G_1 brings about φ and a coalition G_2 brings about ψ, had they acted as the coalition $G_1 \cup G_2$ they would have together brought about $\varphi \wedge \psi$. Our evidence-based perspective of ability suggests that as they showed evidence, they are deemed able. In formula:

$$\vdash E_{G_1}\varphi \wedge E_{G_2}\psi \rightarrow C_{G_1\cup G_2}(\varphi \wedge \psi).$$

It is a powerful formal device for our theory of evidence-based ability since it allows to deduce potential abilities of coalitions of agents from smaller "successes" in the society of agents. We can use the information of actual agency and suggest that the group of agents $G_1 \cup G_2$ could potentially be solicited to bring about the goal $\varphi \wedge \psi$, for instance in a context of web services orchestration.

The logic of coalitional agency and ability. Our methodology to finding the coalitional version of Elgesem's logic rather naïvely consists in thinking of a principle and trying to show that it is not acceptable in some scenario. If no counterexample is found, we must accept it at that stage. We found only the previous two principles that we think are adequate with Elgesem's philosophy and our analysis above.

The logic of coalitional agency and ability can be conveniently presented as a Hilbert system. For all groups G, G_1, and G_2 and formulas φ and ψ:

Ax0 $\vdash \varphi$, when φ is a tautology in propositional logic
Ax1 $\vdash E_G\varphi \wedge E_G\psi \rightarrow E_G(\varphi \wedge \psi)$
Ax2 $\vdash E_G\varphi \rightarrow \varphi$
Ax3 $\vdash E_G\varphi \rightarrow C_G\varphi$
Ax4 $\vdash \neg C_G\bot$
Ax5 $\vdash \neg C_G\top$
Ax6 $\vdash \neg C_\emptyset\varphi$
Ax7 $\vdash E_{G_1}\varphi \wedge E_{G_2}\psi \rightarrow C_{G_1\cup G_2}(\varphi \wedge \psi)$
ERE if $\vdash \varphi \leftrightarrow \psi$ then $\vdash E_G\varphi \leftrightarrow E_G\psi$
ERC if $\vdash \varphi \leftrightarrow \psi$ then $\vdash C_G\varphi \leftrightarrow C_G\psi$

From here, one can provide a class of models for which the logic is sound and complete. It can be proved that the decision problem of satisfiability checking within the logic can be solved in space polynomial.

Towards stronger logics. From this minimal logic, one can strengthen it and adapt it to more specific application domains. For instance, the language of our logic talking about coalitions, allows to formulate a variant to the controversial principle of law $E_xE_y\varphi \rightarrow E_x\varphi$ that states that the delegating entity x is responsible for what the delegate y brings about. Elgesem rejected it. Instead we could adopt $E_{G_1}E_{G_2}\varphi \rightarrow E_{G_1\cup G_2}\varphi$, which only attributes a shared responsibility to the delegating entity. Nevertheless, $E_{G_1}E_{G_2}\varphi \rightarrow E_{G_2}\varphi$ remains true in virtue of **Ax2**.

2. REFERENCES

[1] D. Elgesem. The modal logic of agency. *Nordic J. Philos. Logic*, 2(2), 1997.

[2] G. Governatori and A. Rotolo. On the Axiomatisation of Elgesem's Logic of Agency and Ability. *Journal of Philosophical Logic*, 34:403–431, 2005.

[3] S. Miller. *Social Action (A Teleogical Account)*. Cambridge University Press, 2001.

[4] F. Santos, A. Jones, and J. Carmo. Responsibility for Action in Organisations: a Formal Model. In G. Holmström-Hintikka and R. Tuomela, editors, *Contemporary Action Theory*, volume 1, pages 333–348. Kluwer, 1997.

Strategic voting and the logic of knowledge

(Extended Abstract)

Hans van Ditmarsch
Logic
University of Seville
hvd@us.es

Jérôme Lang
LAMSADE
Université Paris Dauphine
lang@irit.fr

Abdallah Saffidine
LAMSADE
Université Paris Dauphine
abdallah.saffidine@gmail.com

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Theory

Keywords

Social choice, Dynamics, Epistemic logic

1. INTRODUCTION

A well-known fact in social choice theory is that strategic voting, also known as manipulation, becomes harder when voters know less about the preferences or votes of other voters. Standard approaches to manipulation in social choice theory [6] as well as in computational social choice [3] assume that the manipulating voter or the manipulating coalition knows perfectly how the other voters will vote. Some approaches [2] assume that voters have a probabilistic prior belief on the outcome of the vote, which encompasses the case where each voter has a probability distribution over the set of profiles. A recent paper [5] extends coalitional manipulation to incomplete knowledge, by distinguishing manipulating from non-manipulating voters and by considering that the manipulating coalition has, for each voter outside the coalition, a set of possible votes encoded in the form of a partial order over candidates. Uncertainty of voters about the uncertainties of other voters, i.e., higher-order beliefs of voters, has not been treated in full generality.

We model how uncertainty about the preferences of other voters may determine a strategic vote, and how a reduction in this uncertainty may change a strategic vote. A link between epistemic logic and voting has been given in [4]—they use knowledge graphs to indicate that a voter is uncertain about the preference of another voter. A more recent approach, within the area known as social software, is [8]. The recent [5] walks a middle way namely where equivalence classes are called information sets, as in treatments of knowledge and uncertainty in economics, but where the uncertain voter, or coalition, does not take the uncertainty of other voters into account.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. KNOWLEDGE AND VOTING

We assume *voters* $\mathcal{N} = \{1, \ldots, n\}$, *candidates* $\mathcal{C} = \{a, b, c, \ldots\}$, and *votes* $V_i \subseteq \mathcal{C} \times \mathcal{C}$ that are linear orders. If agent i prefers candidate a to candidate b, we write $a \succ_i b$. A *profile* P is a collection $\{V_1, \ldots, V_n\}$ of n votes, and a *voting rule* is a function $F : O(\mathcal{C})^n \to \mathcal{C}$ from the set of profiles to the set of candidates. We may further assume a tie-breaking mechanism. If $F(P[V_i/V_i']) \succ_i F(P)$, then V_i' is a *successful manipulation.* Given a profile P, a profile P' is an *equilibrium profile* iff no agent has a successful manipulation.

We model uncertainty about voting as incomplete knowledge about profiles. This terminology is standard in modal logic. The novelty consists in taking models with *profiles* instead of *valuations of propositional variables.*

Definition 1 (Knowledge profile). *A* profile model *is a structure* $\mathcal{P} = (S, \{\sim_1, \ldots, \sim_n\}, \pi)$*, where* S *is a* domain *of abstract objects called* profile names*; where for* $i = 1, \ldots, n$*,* $\sim_i$ *is an* indistinguishability relation*, that is, an equivalence relation; and where* valuation $\pi : S \to O(\mathcal{C})^n$ *assigns a profile to each profile name. A* knowledge profile *is pointed structure* $\mathcal{P}_s$ *where* $\mathcal{P}$ *is a profile model and* s *is a profile name in the domain of* $\mathcal{P}$*.*

Definition 2 (Knowledge). *Given a knowledge profile* $\mathcal{P}_s$ *and a proposition* q*,* agent i knows that q *if and only if* q *holds for all profile names in* $\mathcal{P}$ *indistinguishable for* i *from* s *(i.e., for all* $s' \in \mathcal{P}$ *such that* $s \sim_i s'$*).*

Propositions like 'voter i knows the profile' or even 'voter i knows that P is an equilibrium profile' have a precise formal description in this framework.

Under conditions of incomplete knowledge it may be that voter i (or coalition G) can manipulate the outcome of a profile P but does not know that, because she considers another profile (name) possible that she cannot manipulate. Such situations call for more refined notions of manipulation, that also involve knowledge. They can be borrowed from the knowledge and action literature [9, 7]. Our main interest is when voters know the manipulation.

Definition 3 (Knowledge of manipulation). *Given a knowledge profile* $\mathcal{P}_s$*. Voter* i *knows* **de re** *that she can* strongly successfully *manipulate* $\mathcal{P}_s$ *if there is a vote* V_i' *such that for all* t *such that* $s \sim_i t$*,* $F(P[V_i'/V_i]) \succ_i F(P)$*, where* t *has profile* P*.*

In the presence of knowledge, the definition of an equilibrium extends naturally. The trick is that for each agent, the combination of an agent i and an equivalence class $[s]_{\sim_i}$ for that agent (for some state s in the knowledge profile) defines a *virtual agent*. Thus, agent i is multiplied in as many virtual agents as there are equivalences classes for $\sim_i$ in the model. An equilibrium is then a combination of votes such that none of the virtual agents has an interest to deviate. An intuitively more appealing solution than virtual agents, applied in [1], is to stick to the agents we already have, but change the set of votes into a larger set of *conditional votes* — where the conditions are the equivalence classes for the agents. This we will now follow in the definition below. For *risk averse* voters (this criterion fits best our probability-free and utility-free model — it was also chosen in [5]) we can effectively determine if a conditional profile is an equilibrium without taking probability distributions into account, unlike in the more general setting of Bayesian games that it originates with.

Definition 4 (Conditional equilibrium). *Given is a knowledge profile model $\mathcal{P}$. For each agent i, let CV_i be the set of all* conditional votes *for that agent. A* conditional vote *is a function $CV_i : S/{\sim_i} \to O(\mathcal{C})$, i.e., a function that assigns to each equivalence class for that agent a vote. A* conditional profile *is a collection of n conditional votes, one for each agent. A conditional profile is an* equilibrium *iff no agent has a successful manipulation. A conditional profile is a* strong equilibrium *iff no coalition has a successful manipulation.*

3. EXAMPLE

Consider two voters a, b, four candidates $1, 2, 3, 4$, and three profile names s, t, u (for two profiles P and P') as below. The profile name s is assigned to profile P, wherein $a \succ_1 c \succ_1 b \succ_1 d$ and $d \succ_2 c \succ_2 b \succ_2 a$, etc. Profile names that are indistinguishable for a voter i are linked with an i-labelled edge. The partition for 1 on the domain is therefore $\{\{s,t\},\{u\}\}$, and the partition for 2 on the domain is $\{\{s\},\{t,u\}\}$.

1	2		1	2		1	2
a	d		a	d		d	d
c	c	—1—	c	c	—2—	c	c
b	b		b	b		b	b
d	a		d	a		a	a
s, P			t, P			u, P'	

Note that the names s and t are assigned to the same profile. However, s and t have different epistemic properties. In s, 2 knows that 1 prefers a over d, whereas in t 2 does not know that.

Consider a plurality vote with a tie-breaking rule $b \succ a \succ c \succ d$. If there had been no uncertainty, then in profile P, if 1 votes for her preference a and 2 votes for his preference d, then the tie prefers a, 2's least preferred candidate. If instead 2 votes c, a will still win. But if 2 votes b, b wins. We observe that (a, b) and (b, b) are equilibria pairs of votes, and that for 1 voting a is dominant. If there had been no uncertainty, then in profile P' pair (d, d) is the dominant equilibrium.

This situation changes when we take the uncertainty of the voters into account. There are two equilibria that we can associate with this knowledge profile model. Below, the conditional vote for 1 in the first equilibrium actually is defined as (given that $\pi(t) = P$ and $\pi(u) = P'$): $CV_1(\{t\}) = V_1$ and $CV_1(\{u\}) = V_1'$; the vote for 2 is conditional to one equivalence class — in other words, it is unconditional. The equivalent verbose formulation is more intelligable:

- (if 1 prefers a then 1 votes a and if 1 prefers d then 1 votes d, 2 votes b),
- (if 1 prefers a then 1 votes b and if 1 prefers d then 1 votes d, 2 votes b).

Unfortunately for voter 2, if the actual profile is P' so that d is his equilibrium vote, he will still not be inclined to cast that vote because he considers it possible that the profile is P, where, if 2 votes d and 1 votes a, a gets elected, voter 2's least preferred candidate. As 2 is risk averse his (known) equilibrium vote is therefore b.

If P' is the case, voter 1 has an incentive to make her true vote (i.e., her intention) known to 2, and even to declare her vote prior to 2.

4. DYNAMICS

The modal logical setting for voting and knowledge can be extended with dynamic logical operations. Three examples are: *deliberation of a coalition*, *public announcement of a proposition* (such as an agent revealing her true preference), and *declaring a vote*. These can be formalized as semantic operations $P_s \mapsto P_s|G$, $P_s \mapsto P_s|p$ (for proposition p), and $P_s \mapsto P_s|\mathsf{d}(V_i)$, respectively. All these correspond to standard dynamic epistemic logical operations [10].

5. REFERENCES

[1] T. Ågotnes and H. van Ditmarsch. What will they say? - Public announcement games. *Synthese*, 179(S.1):57–85, 2011.

[2] S. Barbera, A. Bogomolnaia, and H. van der Stel. Strategy-proof probabilistic rules for expected utility maximizers. *Mathematical Social Sciences*, 35(2):89–103, 1998.

[3] J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[4] S. Chopra, E. Pacuit, and R. Parikh. Knowledge-theoretic properties of strategic voting. *Proceedings of JELIA*, pages 18–30, 2004. LNCS 3229.

[5] V. Conitzer, T. Walsh, and L. Xia. Dominating manipulations in voting with partial information. *Proceedings of AAAI*, 2011. To appear.

[6] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.

[7] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63:185–219, 2004.

[8] R. Parikh, C. Tasdemir, and A. Witzel. The power of knowledge in games. *Proceedings of the Workshop on Reasoning About Other Minds*, 2011. CEUR Workshop Proceedings. Volume: 751.

[9] J. van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[10] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.

Exclusivity-based Allocation of Knowledge

(Extended Abstract)

Madalina Croitoru
University of Montpellier 2
France
croitoru@lirmm.fr

Sebastian Rudolph
Karlsruhe Institute of Technology
Germany
rudolph@kit.edu

ABSTRACT

The classical setting of query answering either assumes the existence of just one knowledge requester, or the knowledge requests from different parties are treated independently from each other. This assumption does not always hold in practical applications where requesters often are in direct competition for knowledge. We propose a formal model for this type of scenario by introducing the Multi-Agent Knowledge Allocation (MAKA) setting which combines the fields of query answering in information systems and multi-agent resource allocation.

Categories and Subject Descriptors

K.6.0 [**Management of Computing and Information Systems**]: General—Economics

General Terms

Economics, Theory

Keywords

Auction and mechanism design

1. INTRODUCTION

Conjunctive query answering (between a knowledge requester and a knowledge provider) constitutes the de-facto standard of interacting with resources of structured information: databases or ontological information systems. The classical setting in query answering is focused on the case where just one knowledge requester is present. In case multiple requesters are present, the queries posed by different parties are processed and answered as independent from each other, thus making the multi-requester scenario a straightforward extension of the individual case.

While the above practice is natural in some cases, the assumption that queries can be processed independently clearly does not always hold in practical applications where the requesters are in direct competition for information. Let us consider for instance a multi-agent setting, with requester agents concurrently demanding information from a provider agent (example scenarios include military applications, news agencies, intelligence services, etc.). Of course, in this context, requester agents will not be willing to share "sensitive" information with other agents.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

A structurally related problem is the multi-agent resource allocation (MARA) setting [2]. However, in such a setting (i) the agents ask for resources (not knowledge) and (ii) agents *a priori* know the pool of available resources. Work in this field either aims at bidding language expressiveness or algorithmic aspects of the allocation problem (see for instance [5, 1, 4] and others). The notion of multiplicity of resources, or resources used exclusively or shared has also been recently investigated in a logic-based language [6].

In the proposed multi-agent knowledge allocation (MAKA) setting, the n requester agents, at some given time (in a single-step), ask for knowledge (and not resources). They express their requests in the form of conjunctive queries that are endowed with exclusivity constraints and valuations, which indicate the subjective value of potentially allocated answers. Knowledge allocation poses interesting inherent problems not only from a bidding and query answering viewpoint, but also in terms of mechanism design.

The aim of this paper is to motivate and introduce the novel problem of Multi-Agent Knowledge Allocation and lay down future work directions opened by this setting: increased expressivity, dynamic allocations, fairness, multiple providers etc.

2. QUERYING WITH EXCLUSIVITY CONSTRAINTS

In [3] we fully introduce our framework of exclusivity-aware querying as a basis for the MAKA bidding formalism. In the following, we will just provide an intuitive overview of this work by the means of an example. Consider the following predicates: `actor`, `director`, `singer` (all unary), `marriage` and `act` (binary) and five constants `AJ` (Angelina Jolie), `BP` (Brad Pitt), `MMS` (Mr. and Ms. Smith), `JB` (Jessica Biel), `JT` (Justin Timberlake). A *knowledge base* consists of ground facts such as:

actor(AJ)	director(AJ)	marriage(AJ, BP)
actor(BP)	singer(JT)	act(AJ, MMS)
actor(JB)		act(BP, MMS)

If we consider a set of variables $V = \{x, y\}$ and the set of constants $C = \{\mathtt{AJ}, \mathtt{BP}, \mathtt{MMS}, \mathtt{JB}, \mathtt{JT}\}$, then $\mathtt{actor}(x)$, $\mathtt{act}(y, \mathtt{MMS})$, $\mathtt{marriage}(\mathtt{AJ}, \mathtt{BP})$ are all *atoms* over the sets P and C.

Since in the MAKA scenario, requesters might be competing for certain pieces of knowledge, we have to provide them with the possibility of asking for an atom exclusively (exclusive) or not (shared). This additional information is captured by the notion of *exclusivity-annotated* atoms, ground facts and queries.

Some exclusivity-annotated atoms would for instance be: $\langle \mathtt{actor}(x), \mathsf{sh} \rangle$, $\langle \mathtt{marriage}(\mathtt{AJ}, \mathtt{BP}), \mathsf{exc} \rangle$ etc.

Note that the idea of exclusivity annotation is a novel concept going beyond the classical query answering framework. We assume an order exclusive $\succeq$ shared being used for query answering. It allows to specify concisely that an answer delivered exclusively is suitable for a knowledge requester who demanded that information shared (but not vice versa).

For example, a query asking exclusively for marriages between actors and directors (where only the "marriage" itself is required as exclusive information, but the "actor" and "director" knowledge is sharable with other knowledge requester agents) is:

$$\langle \texttt{marriage}(x,y), \text{exclusive}\rangle \wedge \big((\langle \texttt{actor}(x), \text{shared}\rangle \wedge \langle \texttt{director}(y), \text{shared}\rangle) \vee (\langle \texttt{actor}(y), \text{shared}\rangle \wedge \langle \texttt{director}(x), \text{shared}\rangle)\big).$$

There is only one answer to this query w.r.t. our previously introduced knowledge base: $\mu = \{x \mapsto \texttt{AJ}, y \mapsto \texttt{BP}\}$. This means that $\texttt{marriage}(\texttt{AJ},\texttt{BP})$ can only be exclusively allocated (as $\langle \texttt{marriage}(\texttt{AJ},\texttt{BP}), \text{exclusive}\rangle$) but the $\texttt{director}(\texttt{AJ})$ and $\texttt{actor}(\texttt{BP})$ atoms can be either "shareably" allocated with other requesters ($\langle \texttt{actor}(\texttt{BP}), \text{shared}\rangle$) or exclusively allocated only to one requester agent ($\langle \texttt{director}(\texttt{AJ}), \text{exclusive}\rangle$).

3. THE KNOWLEDGE ALLOCATION PROBLEM DEFINED

Multi Agent Knowledge Allocation (MAKA) can be interpreted as an abstraction of a *market-based* centralized distributed knowledge-based system for query answering. In such a MAKA system, there is central node a, the *auctioneer* (or the *knowledge provider*), and a set of n nodes, $I = \{1, \ldots, n\}$, the *bidders* (or the *knowledge requesters*), which express their information need (including exclusivity requirements) via queries, which are to be evaluated against a knowledge base $\mathcal{K}$, held by the auctioneer.

The auctioneer asks bidders to submit in a specified common language, the *bidding language*, their *knowledge request*: $\langle q, \varphi\rangle$ where q is an exclusivity-annotated query and $\varphi : \mathbb{N} \to \mathbb{R}_+$ is a valuation function.

Following the ongoing example in the paper, a knowledge request for an exclusively known marriage between a known actor and a known director, where each such marriage information is paid 30 units for would be the singleton set $\{\langle q, \varphi\rangle\}$ with

$$\begin{aligned} q = \quad & \langle\langle \texttt{marriage}(x,y), \text{exclusive}\rangle \wedge \\ & \big((\langle \texttt{actor}(x), \text{shared}\rangle \wedge \langle \texttt{director}(y), \text{shared}\rangle) \vee \\ & (\langle \texttt{actor}(y), \text{shared}\rangle \wedge \langle \texttt{director}(x), \text{shared}\rangle)\big), \\ \varphi = \quad & k \mapsto 30 \cdot k. \end{aligned}$$

The valuation function $\varphi : \mathbb{N} \to \mathbb{R}_+$ can be defined in several ways. Assuming that $val_q^i \in \mathbb{R}_+$ denotes a bidder i's interest to obtain a single answer to a query q, standard valuation options are:

- naive valuation: $\varphi^n(|S|) = |S| \cdot val_q^i$,
- threshold valuation: $\varphi^t(|S|) = |S| \cdot val_q^i$ if $|\ S\ | \leq threshold_{q_i}^i$ and $|S| \cdot (val_q^i - discount_q^i)$ otherwise,
- budget valuation: $\varphi^b(|S|) = min\{\varphi_i(|S|), budget_i\}$ where φ_i can either be φ_i^n or φ_i^t.

Based on bidders' valuations, the auctioneer will determine a *knowledge allocation*, specifying for each bidder her obtained knowledge bundle and satisfying the *exclusivity constraints* (expressing that exclusivity annotations associated to atoms in the respective bundle are indeed complied with).

Given a knowledge base and a set of n bidders, a *knowledge allocation* is an n-tuple of subsets of the exclusivity-enriched knowledge base (i.e., the knowledge base atoms annotated with both exclusive and shared). An allocation needs to satisfy two conditions: First, we cannot allocate the same atom as both shared and exclusive. Second, an exclusive atom can only be allocated to one agent.

Given a knowledge allocation, one can compute its *global value* by summing up the individual prizes paid by the bidders for the share they receive. Obviously, the knowledge allocation problem aims at an *optimal allocation*, which maximizes this value.

Please see [3] providing more details and a full formalisation of the above intuitions, as well as a network representation of the problem, such that the winner determination can be cast into a max flow problem on the proposed graph structure.

4. CONCLUSION AND FUTURE WORK

We have introduced the problem of Multi-Agent Knowledge Allocation by drawing from the fields of query answering in information systems and combinatory auctions. To this end, we have sketched a bidding language based on exclusivity-annotated conjunctive queries. This approach opens up interesting work directions such as:

- **Extending the bidding language**: One straightforward extension would be to allow not just for ground facts (like $\texttt{marriage}(\texttt{AJ},\texttt{BP})$) to be delivered to the requester but also for "anonymized" facts (like $\texttt{marriage}(\texttt{AJ},*)$ or, more formally $\exists x.\texttt{marriage}(\texttt{AJ},x)$), which require handling adaption.

- **Extending knowledge base expressivity**: On one hand, the knowledge base formalism could be extended to cover not just ground facts but more advanced logical statements such as Datalog rules (used in deductive databases) or ontology languages. In that case, a distinction has to be made between propositions which are explicitly present in the knowledge base and those entailed by it.

- **Covering Dynamic Aspects of Knowledge Allocation**: In particular in the area of news, dynamic aspects are of paramount importance: news items are annotated by time stamps and their value usually greatly depends on their timeliness. Moreover we can assume the information provider's knowledge pool to be continuously updated by incoming streams of new information.

- **Multiple Providers**: Finally, it might be useful to extend the setting to the case where multiple agents offer knowledge; in that case different auctioning and allocation mechanisms would have to be considered. This would also widen the focus towards distributed querying as well as knowledge-providing web-services.

Acknowledgments. We thank Angelina Jolie and Brad Pitt for serving as a source of our inspiration.

5. REFERENCES

[1] C. Boutilier and H. Holger. Bidding languages for combinatorial auctions. In *Proc. IJCAI'01*, pages 1211–1217, 2001.

[2] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[3] M. Croitoru and S. Rudolph. Towards multi-agent knowledge allocation. Technical Report RR-12003(18), LIRMM, http://hal-lirmm.ccsd.cnrs.fr/lirmm-00657422/en/, 2012.

[4] A. Giovannucci, J. Rodriguez-Aguilar, J. Cerquides, and U. Endriss. Winner determination for mixed multi-unit combinatorial auctions via petri nets. In *Proc. AAMAS'07*, 2007.

[5] N. Nisan. Bidding and allocations in combinatorial auctions. In *Proc. EC-2000*, 2000.

[6] D. Porello and U. Endriss. Modelling combinatorial auctions in linear logic. In *Proc. KR'10*, 2010.

Role Selection in Ad Hoc Teamwork

(Extended Abstract)

Katie Genter, Noa Agmon, and Peter Stone
Department of Computer Science
The University of Texas at Austin
{katie,agmon,pstone}@cs.utexas.edu

ABSTRACT

An ad hoc team setting is one in which teammates must work together to obtain a common goal, but without any prior agreement regarding how to work together. In this work we introduce a *role-based approach* for ad hoc teamwork, in which each teammate is inferred to be following a specialized role that accomplishes a specific task or exhibits a particular behavior. In such cases, the role an ad hoc agent should select depends both on its own capabilities and on the roles currently selected by other team members. We present methods for evaluating the influence of the ad hoc agent's role selection on the team's utility and we examine empirically how to choose the best suited method for role assignment in a complex environment. Finally, we show that an appropriate assignment method can be determined from a limited amount of data and used successfully in new tasks that the team has not encountered before.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Ad Hoc Teamwork, Agent Cooperation, Coordination

1. INTRODUCTION

Ad hoc teamwork is a relatively new research area [1, 4, 5] that examines how an agent ought to act when placed on a team with other agents such that there was no prior opportunity to coordinate behaviors. This is in contrast to most prior multiagent teamwork research, which often requires explicit coordination protocols, languages, and/or shared assumptions (e.g. [3, 6]).

In some team domains, the team behavior can be broken down into *roles*. In such domains, an ad hoc agent's main task is to decide which role to assume, such that the team's performance is maximized. The decision of which role an ad hoc agent should assume is situation-specific: it depends on the task the team performs, the environment in which it operates, and the capabilities of the team members. One trivial approach to the problem is for an ad hoc agent to assume the role at which it is most *individually* capable. However, the choice of optimal role—one that results in highest *team* utility—rarely depends only on the ad hoc agent, but also on the ability and behavior of the other team members. We therefore examine the contribution of an ad hoc agent to the team by the measure of *marginal utility*, which is the increase in a team's utility when an ad hoc agent is added to the team and assumes a particular role. An *optimal mapping* of an ad hoc agent to a role is, therefore, one that maximizes the marginal utility, hence maximizing the contribution of the ad hoc agent to the team's utility.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. PROBLEM DEFINITION

An ad hoc teamwork problem is one in which several agents find themselves in a situation where they all have perfectly aligned goals, yet they have had no previous opportunity to coordinate their teamwork [5]. In this work we introduce the *role-based* ad hoc teamwork problem, which is one that requires or benefits from dividing the task at hand into roles. Throughout this paper we refer to the agents that make up a team as either *ad hoc agents* or *teammates*. Ad hoc agents are agents whose behavior we can control, while teammates are agents that we have no control over, potentially because they were programmed at a time when future collaboration with our agents was unforeseeable.

Formally, let task d have m roles $R(d) = \{r_0, ..., r_{m-1}\}$. Let $\mathbf{A} = \{a_0, ..., a_{n-1}\}$ be the set of ad hoc agents and $\mathbf{B} = \{b_0, ..., b_{k-1}\}$ be the set of teammates such that $T = A \cup B$ is the team that is to perform task d. Let mapping $\mathbf{P} : B \rightarrow R(d)$ be the mapping of B to roles $\{r_0, ..., r_{m-1}\}$ and let mapping $\mathbf{S} : A \rightarrow R(d)$ be the mapping of A to roles $\{r_0, ..., r_{m-1}\}$. Finally, let mapping $\mathbf{SP} : T \rightarrow R(d)$ be the combination of mappings S and P. A team score $U(SP, d, T)$ results when the set of agents T perform a task d, with each $t_j \in T$ fulfilling some role $r_i \in R(d)$ under mapping SP. The marginal utility $MU(S, P)$ obtained by mapping S, assuming P is the mapping of B to roles, is the score improvement obtained when S maps A to roles. Hence, marginal utility $MU(S, P) = U(SP, d, T) - U(P, d, B)$.

Given that mapping P is fixed, the role-based ad hoc team problem is to find a mapping S that maximizes marginal utility. Although for the remainder of this paper we focus on the case where $A = \{a_0\}$, the problem definition provided above is valid for any number of ad hoc team agents.

3. MODELS FOR CHOOSING A ROLE

The gold standard way for an ad hoc agent to determine the marginal utility of selecting a particular role is to determine $U(SP, d, T)$ for each possible role it could adopt. However, in practice, the ad hoc agent must *predict* its marginal utility for all possible roles and then select just *one* role to adopt. Here we lay out three possible models with which the ad hoc agent could do this prediction.

Unlimited Role Mapping Model The value received by the team for an agent performing a role is not dependant on the roles fulfilled by other teammates.

Limited Role Mapping Model The benefit the team recieves for an agent performing role r_i is dependent on the number of agents performing r_i. The team recieves no benefit for an additional agent performing r_i if this results in less than (greater than) r_i^{min} (r_i^{max}) agents performing r_i.

Incremental Role Mapping Model The value added by an agent performing a role is correlated with the number of agents performing that role via a (1) logarithmic, (2) exponential, or (3) sigmoidal function.

4. MODEL EVALUATION

We examine each of the three models described above in a capture-the-flag style variant of Pacman [2]. The Pacman map is divided into two halves and two teams compete by attempting to eat the food on the opponent's side of the map while defending the food on their side. A team wins by eating all but two of the food pellets on the opponent's side or by eating more pellets than the opponent before time expires. The result of each game is a *score differential*—the difference between the number of pellets protected by the team and the number of pellets protected by the opponent.

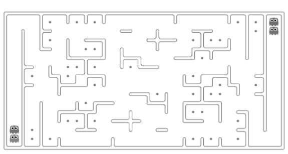


Figure 1: Sample Pacman capture-the-flag map.

4.1 Determining the Best-Suited Model

We use three tasks to determine which of the models best represents the marginal utility of a role selection for the Pacman Capture-the-Flag environment, where a *task* is defined by the number of opponents and the map. In each task we consider two roles that could be performed: $R = \{$offense, defense$\}$.

We start by gathering full sets of *gold standard data*. In particular, we gather score differentials over one thousand games for each team of zero to six offensive agents and zero to six defensive agents (49 teams). In order to emphasize differences in score differentials close to zero, we input the score differential from each game into the sigmoid function $1/1 + e^{-0.13*scoreDifferential}$ and average the results to obtain gold standard data. Then we use the gold standard data to determine the *gold standard decision* of whether an ad hoc agent should perform an offensive role or a defensive role on any team composed of zero to five offensive agents and zero to five defensive agents. To determine the gold standard decision we look at whether the gold standard data is greater for the team with one extra defensive player or the team with one extra offensive player.

For each of the model functions, we input the gold standard data and the model function into a least squares curve fitting algorithm and obtain *fitted parameters* for the model function. We then use the fitted parameters to calculate *fitted results* for all 49 teams. Lastly, we translate these fitted results into *fitted decisions* using the same methodology used to translate the gold standard data into gold standard decisions. Then we compare the number of times the gold standard decision does not match the fitted decision for a particular team arrangement —in other words, the number of *incorrect decisions*. Our experiments showed that the exponential and sigmoidal functions of the incremental model made the fewest incorrect decisions across the three tasks. Hence we conclude that in the Pacman Capture-the-Flag domain, at least on the maps and opponents studied, the incremental model using an exponential or sigmoidal function most accurately models team utility.

4.2 Predictive Modeling

Once a model type has been selected for a domain, the ad hoc agent can use this model to *predict* the marginal utility of role selection on new tasks for which we have limited gold standard data. We do this by choosing fitted parameters for the new task based on available data. Our experiments showed that the prediction accuracy of each incremental model function variation improved as more data was available, and that some variations did surprisingly well even when provided extremely sparse data.

5. FUTURE WORK

This research is among the first to study role-based ad hoc teams. As such, there are many potential directions for future work. We plan on expanding our work into more complicated environments with more than two potential roles to fulfill and more than one ad hoc agent. Additionally, we wish to consider the case in which the ad hoc agents encounter teammates that are running unfamiliar behaviors, forcing the ad hoc agents to model their teammates.

6. ACKNOWLEDGEMENTS

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), and the FHWA (DTFH61-07-H-00030).

7. REFERENCES

[1] M. Bowling and P. McCracken. Coordination and adaptation in impromptu teams. In *AAAI'05*, pages 53–58, 2005.

[2] J. DeNero and D. Klein. Teaching introductory artificial intelligence with pac-man. In *Proceedings of the Symposium on Educational Advances in Artificial Intelligence*, 2010.

[3] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *AIJ*, 86(2):269 – 357, 1996.

[4] E. Jones, B. Browning, M. B. Dias, B. Argall, M. M. Veloso, and A. T. Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *ICRA'06*, pages 570 – 575, 2006.

[5] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI'10*, 2010.

[6] M. Tambe. Towards flexible teamwork. *JAIR*, 7:83–124, 1997.

Integrating Self-organisation into Dynamic Coalition Formation

(Extended Abstract)

Dayong Ye
University of Wollongong
NSW 2522 AU
dy721@uow.edu.au

Minjie Zhang
University of Wollongong
NSW 2522 AU
minjie@uow.edu.au

Danny Sutanto
University of Wollongong
NSW 2522 AU
danny@elec.uow.edu.au

ABSTRACT

In some real systems, e.g., sensor networks, individual agents will often need to form coalitions to accomplish complex tasks. Due to communication or computation constrains, it is infeasible for agents to directly interact with all other peers to form coalitions. Most current coalition formation works, however, overlooked this aspect. Those works usually did not provide an explicitly modeled agent network or assumed agents in a fully connected network, where an agent can communicate with all other agents. Thus, to alleviate this problem, it is necessary to provide a neighbourhood system within which agents can directly interact only with their neighbours. Towards this end, in this paper, we propose a dynamic coalition formation mechanism, incorporated with self-organisation, in a structured agent network. Based on self-organisation principles, this mechanism enables agents to dynamically adjust their degrees of involvement in different coalitions and to join new coalitions at any time.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Coalition Formation, Self-organisation

1. INTRODUCTION

In many applications of multi-agent systems, agents will need to dynamically join together in a coalition to complete a complex task which none of them can complete independently. Recently, many efforts have been done on coalition formation and have achieved very great results. There is a common assumption in these studies that the agent network underlying structure is either not explicitly modeled or the network structure is based on some regular structures, e.g., a fully connected network or a hierarchical network. However, in many real circumstances, particularly in large and distributed environments, it is infeasible for each individual agent to consider all the other agents to form coalitions due to time, communication and computation constrains [4]. One approach to overcome this limitation is to impose some sort of network structure on the agents and require that agents can directly communicate only with their neighbours when forming coalitions. Gaston and desJardins [2, 3], and Glinton et al. [4] made many efforts in this way. The common limitation in [2, 3, 4] is that an agent can join only one coalition and once a coalition is formed for a task, the coalition is fixed and agents cannot leave the coalition, until the task is finished. Against this background, in this paper, our research concentrates on designing a dynamic coalition formation mechanism in a structured agent network, where each agent has only a limited view about its neighbours in the environment and makes decisions based only on this view. In addition, we integrate self-organisation notion into coalition formation which enables agents to dynamically adjust their degrees of involvement in different coalitions and to join new coalitions, via negotiation, at any time if necessary. In that case, agents have more autonomy and flexibility when they execute tasks.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. COALITION FORMATION

In the agent network, agents make decisions based only on local information about the system, and the decision making process of agents is autonomous without external control. Hence, we define a set $P = \{P_1, ..., P_n\}$. P is defined as a *partition* of the *Compatible Relation* R, where $\langle a_i, a_j \rangle \in R$ if and only if a_j is a neighbour of a_i. Accordingly, it can be obtained that $\bigcup_{1 \leq i \leq n} P_i = R$ and $\forall P_i, P_j \in P : i \neq j \Rightarrow P_i \cap P_j = \emptyset$. The set P can be generated by using **Algorithm 1**.

Algorithm 1: Create a partition P on relation R
begin:
(1) **for each** a_i, $a_i \in A$, in sequential order
(2)   **if** $\exists a_j \in A : \langle a_i, a_j \rangle \in R$ **then**
(3) $P_i \leftarrow P_i \cup \{\langle a_i, a_j \rangle\}$;
end

The coalition formation mechanism is illustrated in **Algorithm 2** as follows.

Algorithm 2: Coalition Formation Mechanism
begin:
(1)**Call Algorithm 1** to generate P;
(2)**for each** θ_i, $\theta_i \in \Theta$, in sequential order /*θ_i is a subtask of Θ*/
(3)   randomly select an *IDLE* agent, $a_i \in A$, as *Initiator*;

(4) $State(a_i) \leftarrow BUSY$;
(5) **while** $t < DL(\theta_i)$ **do** * t is the real time * \
(6) **for each** $a_j \in A : \langle a_i, a_j \rangle \in P_i$
(7) **if** $\exists r^l_{\theta_i} \in R(\theta_i) : r^l_{\theta_i} = r_{a_j}$ **and** $r^l_{\theta_i}$ is unsatisfied **then**
(8) **Negotiate**(a_i, a_j);
(9) **end if**
(10) **end for**
(11) **if** $\forall r^l_{\theta_i} \in R(\theta_i) : r^l_{\theta_i}$ is satisfied **then**
(12) **break**;
(13) **else**
(14) **select** a_k as *Mediator* based on the number of a_k's neighbours, where $\langle a_i, a_k \rangle \in P_i$
(15) $State(a_k) \leftarrow BUSY$;
(16) $P_i \leftarrow P_i \circ P_k$;
(17) **end if**
(18) **end while**
(19) **end for**
end

2.1 The Negotiation Protocol

In order to operate the coalition formation mechanism, we need another important component, i.e., a negotiation protocol. The coalition formation problem can be modeled as a negotiation process between an *Initiator* and a *Participant*, where an *Initiator* acts as a *buyer* and a *Participant* plays as a *seller*. The negotiation focuses on a single issue, i.e., the degree of involvement of a *Participant* into a coalition which is being formed by an *Initiator*. Some constrains are listed as follows, with which each agent should comply.

1. An agent, except *Initiator*, can dynamically join multiple coalitions with different degrees of involvement.
2. Temporary agreements can be canceled by either *Initiator*s or *Participant*s without paying penalty.
3. Both *Initiator*s and *Participant*s cannot cancel final agreements, but *Participant*s can adapt the degrees of involvement in their joined coalitions by paying penalty to *Initiator*s and *Participant*s can join other coalitions if necessary.
4. The degree of involvement of an *Initiator* in its initiated coalition is postulated to be 1 and cannot be adapted.

The negotiation protocol employed in this paper extends the alternating offers protocol [5] by allowing an agent to make multiple agreements with other agents and to cancel temporary agreements without paying penalty. Rubinstein's protocol [5] has been widely used for bilateral bargaining, e.g., An et al. [1]. Other more complex negotiation protocols may be also available for our problem, but based on our investigation, Rubinstein's protocol is enough for our problem and it is easy to implement.

There are some possible actions of *buyer* (*Initiator*) and *seller* (*Participant*) agents.

• $offer[o]$, where o is *buyer*'s offer to a *seller*. An offer is determined by four factors, which are the pressure of deadline, the payment of the resource paid by the *buyer* to the *seller*, the duration of using the resource, and the demand/supply ratio of the *buyer*'s required resource.

• $accept[o]$. When a *seller* receives an offer o, it can accept the offer which results in a temporary agreement made with the *buyer*.

• $counter\_offer[o']$. If a *seller* is not happy with an offer o, it can send back a counter-offer o' for its available resource. A counter-offer o' is determined by three aspects, which include the current state of the *seller*, e.g., whether it has joined other coalitions and the degrees of involvement into those coalitions, the payment received by the *seller* from the *buyer*, and the demand/supply ratio of the *seller*'s available resource.

• $cancel[o]$. After a temporary agreement is achieved by a *buyer* and a *seller*, any one of them can cancel the agreement without paying penalty. A final agreement, however, cannot be canceled by either of a *buyer* or a *seller*.

The negotiation protocol, displayed in Line 8 of **Algorithm 2**, is shown in **Algorithm 3** as follows.

Algorithm 3: Negotiate(a_i, a_j)
* a_i is the *buyer* and a_j is the *seller* * \
begin:
while $t <$ predefined period **do** * t is the real time * \
a_i generates an offer o to a_j;
if a_j accepts o **then**
$\mathcal{A}^{\mathcal{T}}(a_i) \leftarrow \mathcal{A}^{\mathcal{T}}(a_i) \cup \{o\}$;
$\mathcal{A}^{\mathcal{T}}(a_j) \leftarrow \mathcal{A}^{\mathcal{T}}(a_j) \cup \{o\}$;
$State(a_j) \leftarrow BUSY$;
return;
else
a_j generates a counter-offer o' to a_i;
if a_i accepts o' **then**
$\mathcal{A}^{\mathcal{T}}(a_i) \leftarrow \mathcal{A}^{\mathcal{T}}(a_i) \cup \{o'\}$;
$\mathcal{A}^{\mathcal{T}}(a_j) \leftarrow \mathcal{A}^{\mathcal{T}}(a_j) \cup \{o'\}$;
$State(a_j) \leftarrow BUSY$;
return;
else
continue;
end if
end if
end while
end

3. CONCLUSION

This paper provided a self-organisation based dynamic coalition formation mechanism which enables agents to dynamically adjust their degrees of involvement in different coalitions to achieve efficient task allocation. This mechanism considers the existence of an underlying network structure and integrates the self-organisation concept. To realise the self-organisation concept, a negotiation protocol is employed. This research can be exploited for completing shared tasks in many distributed systems where resources are distributed and agents are highly autonomous, such as distributed agent-based grid systems, service-oriented computing and distributed sensor networks.

4. REFERENCES

[1] B. An, V. Lesser, D. Irwin, and M. Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *AAMAS'10*, pages 981–988, Toronto, Canada, May 2010.
[2] M. E. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *AAMAS'05*, pages 230–237, Utrecht, Netherlands, Jul. 2005.
[3] M. E. Gaston and M. desJardins. The effect of network structure on dynamic team formation in multi-agent systems. *Computational Intelligence*, 24(2):122–157, 2008.
[4] R. Glinton, K. Sycara, and P. Scerri. Agent organized networks redux. In *AAAI'08*, pages 83–88, 2008.
[5] A. Rubinstein. Prefect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.

An Analysis of Constructive Network Formation Models (Extended Abstract)

Gary Fredericks
University of South Carolina
Columbia, SC 29208
frederga@email.sc.edu

José M. Vidal
University of South Carolina
Columbia, SC 29208
vidal@sc.edu

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Intelligent Agents, Multiagent Systems.*

General Terms

Algorithms, Economics

Keywords

Network formation

1. OVERVIEW

Network formation was originally studied by Jackson and Wolinsky [1]. We analyze a constructive model of network formation that is particularly suited to exhaustive computation.

Our model deals with a fixed-size network where each individual in the network wants to minimize its distance to the other individuals. Distance is measured as simply the number of hops between individuals, and is considered infinite if they are not connected. Building a link has a fixed cost (α) which is in the same units as the measure of distance. We assume an individual will want to build a link iff their decrease in total distance to the rest of the network is greater than or equal to the cost to them for the link.

The network formation process begins with an empty graph on N vertices and at each step a random not-yet-existing edge is added from the set deemed **feasible** by the payment rule. This is repeated until there are no more feasible edges to add. A **payment rule** is a rule for which vertices have to pay for the edge, how they split the cost, and how they decide whether or not to do it. The payment rule determines which edges can be added in a given situation. If a graph can be reached through a network formation process (i.e., can result from a series of feasible transitions), then we say it is **reachable**. The probability that a reachable graph will appear is its **reachability**. Finally, any reachable graph to which no further edges can be added (given a payment rule and α) is called a **sink-graph**. Since every network formation process will necessarily end with a sink-graph, our primary strategy for comparing payment rules in Section 3 is to compare the attributes of the sink-graphs for each payment rule.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

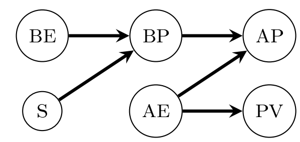


Figure 1: Partial ordering on rules.

We consider six different payment rules. Two natural choices are that the edge might be paid for by one of the two vertices it connects (S), or by both of them equally (BE). It also may be split equally among the whole network (AE). We may also modify the second two so that the cost is divided up fairly, in proportion to how much each vertex benefits from the new edge (BP and AP, respectively). In all the previous payment rules we assume the edge is not built unless all payers agree. Finally, since the AE rule turned out to be trivial (see Section 2), we added a modified payment rule that only requires a strict majority of the network to be in favor of building the edge (PV). In each of the global payment rules (AE, AP, and PV), we also assume that any edge that connects previously disconnected components can be added, for otherwise the process would never get past the empty graph.

2. THEORETICAL ANALYSIS

We have proven several interesting facts about these payment rules. We showed that the AE rule always deadlocks (and is therefore absent from Section 3). We then demonstrated a partial ordering of the six rules under a certain kind of subset relation. Finally, we proved the presence or absence of a certain extreme sink-graph under different rules.

The AE payment rule requires that for an edge to be built each node in the network should be willing to pay $\frac{\alpha}{N}$ for the edge, which means that their expected immediate value for it should be at least $\frac{\alpha}{N}$. However, we have proven that for any minimally-connected graph and not-yet-existing edge there must be a vertex that cannot benefit from adding that edge, and so will not be willing to pay $\frac{\alpha}{N}$. The result is that once the network formation process reaches a tree it cannot proceed further. Therefore AE is omitted in most of what follows.

We have also been able to prove the partial ordering shown in Figure 1. The partial order is a kind of subset relation. Given two payment rules X and Y, we say that $X \subseteq Y$ if for all α the feasibility of any transition under X implies its feasibility under Y. This implies that for a fixed α, any graph reachable under X is also reachable under Y.

All the arrows in Figure 1 have been proven, and all the pairs of payment rules that are incomparable in the diagram have been shown to be incomparable by example.

Finally, we have shown that a certain extreme type of graph is a sink graph for some of the rules but not others. We use the term "Lollipop Graph" to refer to any graph for which $N-1$ vertices form a clique, and the remaining vertex has only a single edge. This graph is the largest possible graph that can still be disconnected by removing a single edge, and so seems inefficient for most applications. Surprisingly, we have shown that for BE lollipop graphs are never sink-graphs, for PV they are never sink-graphs if $N > 5$, but for AP, BP, and S the lollipop graphs are sink-graphs for certain ranges of α. This is unexpected both because it is a sink-graph at all, and also because there are local and global payment rules in each category.

3. STATISTICAL TESTS

We also investigate the payment rules statistically by exhaustively computing properties of all graphs on ten or fewer vertices, and try to identify patterns that seem most amenable to extrapolation to larger graphs. Most of our experimentation was done by treating the network formation process as an acyclic Markov chain, where the Markov states are graphs, the transitions represent adding an edge to the graph, and the transition probabilities are conditioned on α. We used the `nauty` program to populate our database with all 12,293,431 unlabeled graphs for $3 \leq N \leq 10$, and all 251,463,867 transitions between them. Unlabeled graphs were used as an optimization (there are over 35 trillion labeled graphs of the same orders), and extra calculation involving symmetries was necessary to ensure we were computing the same probabilities as would be obtained with labeled graphs.

We used dynamic programming to compute the feasibility thresholds for each transition and payment rule. Using these we then computed the reachability probabilities for each graph and payment rule, conditioned on α. Having this data, it was easy to identify sink-graphs, and to compute the expected value of various graph attributes given a payment rule and α. We compared the total count of reachable graphs for each payment rule as α varies, the expected connectivity and unfairness of the sink-graphs, and the probability that the sink-graph reached is regret-free.

The number of reachable graphs for each payment rule for $N = 1$ is plotted in Figure 2. The subset relationship described in Section 2 is evident in the dominance relationships between the lines in the plot. The line for AP dominates the BP line, which itself dominates both S and BE. The pairs of payment rules that are incomparable in Figure 1 have intersecting lines in Figure 2.

The connectivity of a graph is, formally, the minimum number of vertices that must be removed to disconnect it, so it measures the stability of the network against vertex failures. A disconnected graph has connectivity 0, a minimally connected graph (e.g., a tree) has connectivity 1, and a complete graph has connectivity $N-1$. When examining the expected connectivity of the sink-graphs for each payment rule, we found that (not surprisingly) the connectivity decreases as α increases (though there are curious slight exceptions for each payment rule). Furthermore we found that

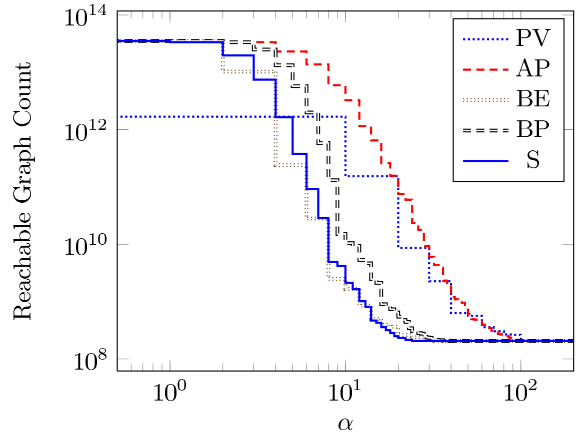


Figure 2: Number of reachable order-10 graphs for each payment rule as α changes. Since the x-axis is plotted logarithmically, the lines extend infinitely to the left; at $\alpha = 0$ all lines meet at the same value.

the S rule gives reliably less connectivity than the other four rules, which had mixed relationships amongst themselves.

We define unfairness as the maximum difference between any two vertices' total distances to the rest of the graph. When analyzing the expected unfairness of the sink-graphs for each payment rule, we found that the S rule was worst for nearly all α. The other local payment rules (BE and BP) were better for small α (< 6), while the global payment rules (AP and PV) were better for large α (> 20).

The final characteristic we analyzed was one we called "regret-free", which describes a graph for which all of the existing edges are still worth adding. That is, for all edges ij of the graph g, the transition from $(g - ij)$ to g is feasible. We computed the probability that the sink-graphs for each payment rule would be regret-free. We found that the probabilities are quite low for small α (in fact as the order increases, the minimum probability seems to approach zero for all payment rules), and slowly build up towards 1 as α increases (eventually the only reachable graphs are trees, which are trivially regret-free). The probabilities for the three local payment rules seem to rise much faster than the two global payment rules.

We also noted the strange fact that all of the regret-free sink-graphs for $N \leq 10$ for all of our payment rules have some symmetry in them (i.e., there is a non-trivial permutation of the vertices that produces the same graph), which is a relatively uncommon attribute in general.

4. CONCLUSION

Our work has so far dealt with only a small set of payment rules, and only networks with relatively few individuals. Future work could explore through sampling whether the statistical results in Section 3 apply to larger networks, and could try to characterize other payment rules or families of payment rules.

5. REFERENCES

[1] Matthew O. Jackson and Asher Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71(1):44–74, 1996.

On Deconflicting Local Coordination Among Agents

(Extended Abstract)

Manh Tung Pham and Kiam Tian Seow
School of Computer Engineering, Nanyang Technological University, Singapore 639798
pham0028,asktseow@ntu.edu.sg

ABSTRACT

For conflict resolution between local coordination modules of distributed agents, synthesized based on inter-agent constraints, two original ideas are proposed. The first is a designer-comprehensible Distributed Constraint Specification Network (DCSN) for describing the constraint relationships among agents. The second is an algorithm using cut-set theory for generating, from a given DCSN, an AND/OR graph compactly representing all conflict resolution plans. A case study is presented.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents, Multiagent Systems

General Terms

Algorithms, Design, Theory

Keywords

Multiagent Coordination, Conflict Resolution

1. DCSN

We shall use small letters such as n, m, k, to denote integers, and for an integer $n \geq 1$, I_n denotes the set $\{1, 2, ..., n\}$.

Definition 1. *Let $n \geq 2$, $m \geq 1$. A distributed constraint specification network (DCSN) $\mathcal{N}$ is a tuple $(\mathcal{A}, \mathcal{C})$, where $\mathcal{A} = \{A_i \mid i \in I_n\}$ is an agent set of size n and $\mathcal{C} = \{C^k_{J_k} \mid k \in I_m, J_k \subseteq I_n\}$ is an inter-agent constraint set of size m, where $C^k_{J_k} \in \mathcal{C}$ is a constraint specified for the group of agents $\mathcal{A}_{J_k} = \{A_i \mid i \in J_k\}$. In other words, for all $k \in I_m$, the agents in $\mathcal{A}_{J_k}$ must coordinate among themselves to respect $C^k_{J_k}$.*

Each $C^k_{J_k} \in \mathcal{C}$ in a DCSN $\mathcal{N}$ is said to be a relevant constraint for agents in the group $\mathcal{A}_{J_k} = \{A_i \mid i \in J_k\}$. Without loss of generality, assume henceforth that $\bigcup_{k \in I_m} J_k = I_n$, i.e., every agent in $\mathcal{A}$ is in $\mathcal{A}_{J_k}$ for some k, and so every agent needs to coordinate. Then a DCSN can be redefined as $\mathcal{N} = \{(J_k, C^k_{J_k}) \mid k \in I_m, J_k \subseteq I_n\}$. An element $\mathcal{N}^k_1 = (J_k, C^k_{J_k})$ of $\mathcal{N}$ is then called a basic subnet of $\mathcal{N}$; and a nonempty $\mathcal{N}^{\mathcal{S}_r}_r \subseteq \mathcal{N}$ consisting of $r = |\mathcal{S}_r| \geq 1$ basic subnets is called a r-constraint subnet of $\mathcal{N}$ with constraint subset $\{C^k_{J_k} \mid k \in \mathcal{S}_r\}$. Where the constraint subset is arbitrary, a r-constraint subnet is simply denoted by $\mathcal{N}_r$.

A DCSN can be graphically represented by an undirected graph with agents represented by rectangular nodes, and each constraint relevant for an agent group by an oval hyperedge with arcs connecting it to all the agents in the group. A r-constraint subnet $\mathcal{N}_r$ of $\mathcal{N}$ is said to be constraint-connected if the graph representing $\mathcal{N}_r$ is a connected graph.

Given a DCSN $\mathcal{N} = (\mathcal{A}, \mathcal{C})$, the problem of interest is to synthesize local plans and coordination strategies, called coordination modules (CM's) henceforth, for every agent in $\mathcal{A}$ to respect every constraint in $\mathcal{C}$. For this problem, we propose a compositional synthesis approach:

- **Step 1 Basic Subnet Synthesis:** Synthesize for every agent a set of local CM's, one for each of the agent's relevant constraints.
- **Step 2 Subnet Composition**
  -Step 2.1 Conflict Resolution Plan Generation: Generate a conflict resolution plan for the given DCSN.
  -Step 2.2 Conflict Resolution Plan Execution: Compose subnets with conflict resolution by following a precedence order of subnet composition operations in the plan. This is to completely deconflict the local CM's synthesized in Step 1. Each subnet composition operation entails designing deconflicting CM's for the agents concerned to ensure nonconflictingness in the composed subnet.

Henceforth, we present the theory for representing conflict resolution plans using AND/OR graphs [1]. We assume that a subnet composition is an operation on two subnets.

2. PLAN REPRESENTATION

A conflict resolution plan for a DCSN is a finite number of subnet composition operations, with ordering constraints between them. Such a plan may encompass several complete planning sequences, each of which is an ordered sequence of the subnet composition operations that satisfies all the ordering constraints. Executing a given plan means following one of its complete planning sequences to successively compose (the solutions of) different pairs of subnets to form (solutions of) larger subnets, starting with all basic subnets "disconnected" from each other, and ending with all of them correctly composed to form the DCSN.

Observe that a conflict resolution planning sequence for a DCSN $\mathcal{N}$ is a reversal of a successive decomposition, starting with $\mathcal{N}$, of constraint-connected component subnets until only basic subnets remain. This suggests that the forward search problem of generating conflict resolution plans $\mathcal{N}$ can be addressed as a backward search problem of successively decomposing $\mathcal{N}$ into pairs of constraint-connected component subnets until only basic subnets are left.

Definition 2. *The AND/OR graph of conflict resolution plans*

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

for a DCSN $\mathcal{N}$ is a hyper-graph $T_{\mathcal{N}} = (S_{\mathcal{N}}, H_{\mathcal{N}})$, where 1) $S_{\mathcal{N}}$ is the set of nodes of $T_{\mathcal{N}}$ and defined as $S_{\mathcal{N}} = \{\mathcal{N}_r \subseteq \mathcal{N} \mid \mathcal{N}_r$ is constraint-connected $\}$, and 2) $H_{\mathcal{N}}$ is the set of hyper-edges of $T_{\mathcal{N}}$ and defined as $H_{\mathcal{N}} = \{(\mathcal{N}_{r_1}, (\mathcal{N}_{r_2}, \mathcal{N}_{r_3})) \in S_{\mathcal{N}} \times (S_{\mathcal{N}} \times S_{\mathcal{N}}) \mid \mathcal{N}_{r_2} \cap \mathcal{N}_{r_3} \neq \emptyset$ and $\mathcal{N}_{r_1} = \mathcal{N}_{r_2} \cup \mathcal{N}_{r_3}\}$.

The nodes in the AND/OR graph $T_{\mathcal{N}}$ represent constraint-connected subnets of $\mathcal{N}$, and each of the hyper-edges is a pair $(\mathcal{N}_{r_1}, (\mathcal{N}_{r_2}, \mathcal{N}_{r_3}))$ denoting the decomposition of subnet $\mathcal{N}_{r_1}$ into two component subnets $\mathcal{N}_{r_2}$ and $\mathcal{N}_{r_3}$, or equivalently, the composition of $\mathcal{N}_{r_2}$ and $\mathcal{N}_{r_3}$ into $\mathcal{N}_{r_1}$. A hyper-edge points from a node representing a subnet to two nodes representing the component subnets. The node that represents the complete DCSN $\mathcal{N}$ is referred to as the root node and denoted by n_{root}, and the nodes representing basic subnets of $\mathcal{N}$ are referred to as the leaf nodes. The set of all leaf nodes of $T_{\mathcal{N}}$ is denoted by Θ_{leaf}. In what follows, a conflict resolution plan for $\mathcal{N}$ is represented by a tree in $T_{\mathcal{N}}$ that starts at n_{root} and terminates at Θ_{leaf}.

3. AND/OR GRAPH PLAN GENERATION

To generate an AND/OR graph representation of conflict resolution plans, the idea is to enumerate all possible decompositions of a DCSN $\mathcal{N}$ into two constraint-connected component subnets. Each such decomposition corresponds to an edge of the AND/OR graph $T_{\mathcal{N}}$ connecting the root node representing $\mathcal{N}$ to two nodes, with each representing a component subnet. The same decomposition process is then repeated for each of the component subnets until only basic subnets are left.

Definition 3. *The constraint relational network (CRN) $\mathcal{CRN}_r$ of a r-constraint subnet $\mathcal{N}_r^{\mathcal{S}_r} = \{(J_k, C^k_{J_k}) \mid k \in \mathcal{S}_r\}$ is a tuple $(\mathcal{C}_r, \mathcal{R}_r)$, where $\mathcal{C}_r = \{C^k_{J_k} \mid k \in \mathcal{S}_r\}$ is the constraint set of size r in $\mathcal{N}_r$ and $\mathcal{R}_r \subseteq \mathcal{C}_r \times \mathcal{C}_r$ is a relation over $\mathcal{C}_r$, such that $(\forall C^k_{J_k}, C^h_{J_h} \in \mathcal{C}_r)[(C^k_{J_k}, C^h_{J_h}) \in \mathcal{R}_r \Leftrightarrow (J_k \cap J_h \neq \emptyset)]$.*

Observe that enumerating all possible decompositions of a subnet $\mathcal{N}_r$ into two constraint-connected subnets can be done by enumerating all possible cut-sets of its CRN $\mathcal{CRN}_r$. Specifically, consider a cut-set $(\mathcal{C}_x, \mathcal{C}_y)$ that decomposes $\mathcal{CRN}_r$ into two parts, where $\mathcal{C}_x$ and $\mathcal{C}_y$ are the two disjoint sets of vertices of $\mathcal{CRN}_r$ belonging to these two parts. Write $\mathcal{N}_x \sim \mathcal{C}_x$ and $\mathcal{N}_y \sim \mathcal{C}_y$ to denote respectively that $\mathcal{N}_x$ and $\mathcal{N}_y$ are the component subnets induced by $\mathcal{C}_x$ and $\mathcal{C}_y$. Then $\mathcal{N}_x$ and $\mathcal{N}_y$ are two constraint-connected component subnets decomposed from $\mathcal{N}_r$. Conversely, any decomposition of $\mathcal{N}_r$ into two constraint-connected component subnets $\mathcal{N}_x$ and $\mathcal{N}_y$ corresponds to a cut-set $(\mathcal{C}_x, \mathcal{C}_y)$ of $\mathcal{CRN}_r$, with $\mathcal{N}_x \sim \mathcal{C}_x$ and $\mathcal{N}_y \sim \mathcal{C}_y$.

Procedure *GenerateANDORGraph*($\mathcal{N}$)

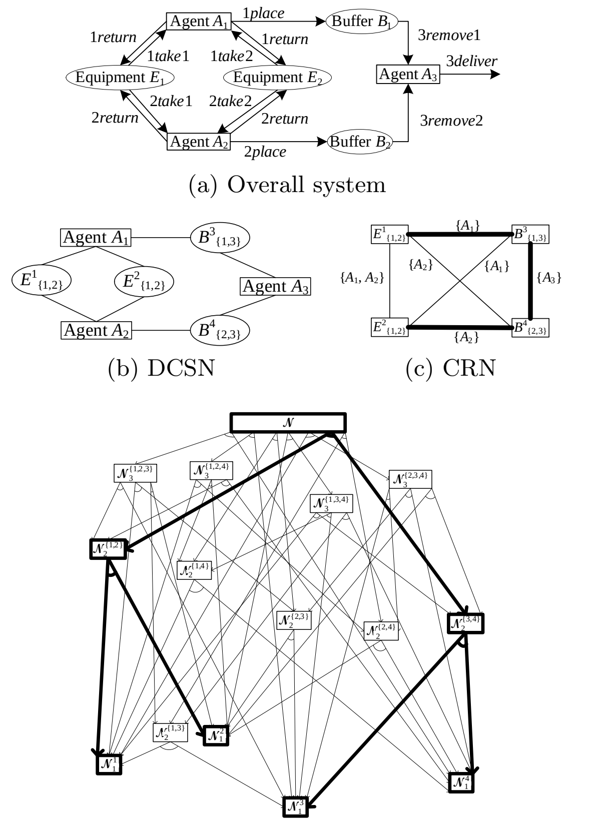
Output: An AND/OR graph $T_{\mathcal{N}} = (S_{\mathcal{N}}, H_{\mathcal{N}})$ of conflict resolution plans for $\mathcal{N}$, initialized with $S_{\mathcal{N}} = \emptyset$ and $H_{\mathcal{N}} = \emptyset$

begin

- **Step 1**: If $\mathcal{N}$ contains only one basic subnet then return; otherwise, convert $\mathcal{N}$ into a $\mathcal{CRN} = (\mathcal{C}, \mathcal{R})$;
- **Step 2**: Compute *CutSets* as the set of all cut-sets of $\mathcal{CRN}$;
- **Step 3 while** *CutSets* $\neq \emptyset$ **do**
  - **Step 3a** Remove a cut-set $(\mathcal{C}_x, \mathcal{C}_y)$ from *CutSets*. Let $\mathcal{N}_x \sim \mathcal{C}_x$ and $\mathcal{N}_y \sim \mathcal{C}_y$;
  - **Step 3b** Add nodes and an edge to T: $S_{\mathcal{N}} = S_{\mathcal{N}} \cup \{\mathcal{N}_x, \mathcal{N}_y, \mathcal{N}_x \cup \mathcal{N}_y\}$, $H_{\mathcal{N}} \cup \{(\mathcal{N}_x \cup \mathcal{N}_y, \mathcal{N}_x, \mathcal{N}_y)\}$;
  - **Step 3c** For $r \in \{x, y\}$, *GenerateANDORGraph*($\mathcal{N}_r$);

4. A CASE STUDY

The system under study [Fig. 1(a)] consists of three agents A_1, A_2 and A_3, and four constraints $E^1_{\{1,2\}}$, $E^2_{\{1,2\}}$, $B^3_{\{1,3\}}$ and $B^4_{\{2,3\}}$. A_1 and A_2 are producer agents that continually follow a production plan: Acquire manufacturing equipment E_1 and E_2 in either order, produce a workpiece, return the equipment to their initial location, move to the buffers' location, place the finished workpiece into the respective one-slot buffer B_1 and B_2, and finally return to the initial state for a new production cycle. A_3 is a delivery agent that continually takes a work piece from either buffer B_1 or B_2, processes, and delivers it to customers. The four constraints $E^1_{\{1,2\}}$, $E^2_{\{1,2\}}$, $B^3_{\{1,3\}}$ and $B^4_{\{2,3\}}$ are formulated to respectively ensure mutual exclusion of equipment use, and no overflow or underflow of buffers.



(a) Overall system

(b) DCSN

(c) CRN

(d) AND/OR graph plans

Figure 1: A manufacturing system.

Fig. 1 shows the DCSN [Fig. 1(b)], CRN [Fig. 1(c)] and AND/OR graph representation of conflict resolution plans [Fig. 1(d)] for the manufacturing system. This AND/OR graph can be automatically generated by applying the procedure *GenerateANDORGraph* to decompose the CRN in Fig. 1(c) recursively.

With appropriate weights assigned to hyper-edges of an AND/OR graph, an A^* search [2] can return a solution tree with minimum depth, namely, a plan that allows maximal simultaneity in executing deconflicting operations, such as the tree highlighted in Fig. 1(d). Presented elsewhere [3], a complete CM solution for this case study can be efficiently computed by executing this highlighted plan tree.

5. REFERENCES

[1] N.Deo. *Graph Theory with Applications to Engineering and Computer Science*. New York: Prentice-Hall, 1974.

[2] S.Russell and P.Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[3] M. T. Pham. *Discrete-event Multiagent Coordination: Framework and Algorithms*. PhD Thesis, NTU, 2011.

Hierarchical Clustering and Linguistic Mediation Rules for Multiagent Negotiation

(Extended Abstract)

Enrique de la Hoz, Miguel Angel Lopez-Carmona
Computer Engineering Department, Universidad de Alcala
Alcala de Henares,Spain
enrique.delahoz@uah.es

Mark Klein
Center for Collective Intelligence, MIT Sloan School of Management, Massachusetts Institute of Technology
m_klein@mit.edu

Ivan Marsa-Maestre
Computer Engineering Department, Universidad de Alcala
Alcala de Henares,Spain
ivan.marsa@uah.es

ABSTRACT

We propose a framework based on Hierarchical Clustering (HC) to perform multiagent negotiations where we can specify the type of agreements needed in terms of utility sharing among the agents.The proposed multi-round mediation process is based on the analysis of the agents' offers at each negotiation round and the generation of a social contract at each round as a feedback to the agents, which explore the negotiation space to generate new offers. This mechanism efficiently manages negotiations following predefined consensus policies avoiding zones of no agreement.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*coherence and coordination*

General Terms

Algorithms, Design, Experimentation

Keywords

Teamwork, coalition formation, coordination, negotiation

1. INTRODUCTION

The type of consensus employed to reach and agreement should be taken into consideration as an integral part when building multiparty negotiation protocols. In this paper, we propose HCPMF, a Hierarchical Consensus Policy based Mediation Framework for Multi-Agent Negotiation. Globally, HCMPF allows to efficiently search for agreements following predefined consensus policies.The protocol is designed to minimize the revelation of private information.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. THE NEGOTIATION PROTOCOL

Each agent sends the mediator an ***initial contract offer***. Based on the received offers, the mediator applies the ***HC algorithm*** [2] to form *clusters of agents*. The cluster with the highest number of agents is selected. Then, the mediator applies the ***OWA operator*** to the offers in the selected cluster to obtain a *feedback contract*. The OWA operator synthesizes the consensus policy to apply. Finally, the mediator verifies if the deadline has been reached. If so, negotiation ends with an agreement on the feedback contract. Otherwise, the mediator computes the ***group distance***, which is a distance estimate to the current feedback contract from the offers in the cluster. If the group distance is below a threshold the negotiation ends with an agreement on the feedback contract. If it is not, the mediator proposes the ***feedback contract*** to the agents. Each agent performs a *local exploration* of the negotiation space using a variation of ***GPS*** [1] to generate a *new offer*. The agent's exploration considers the feedback contract and its utility. The new offer is sent to the mediator, which iterates the process.

3. THE MEDIATION MECHANISMS

The goal of the mediation process is to provide useful feedback to the agents to guide the joint exploration of the negotiation space. This feedback is represented by the *feedback contract*. For the contracts in the highest sized cluster O_{kc}, the centroid $\vec{c_k}$, we compute the distances D_{kc} from the contracts to the centroid and the set of direction vectors R_{kc} from the centroid to the contracts. The OWA operator will be applied to these values in order to obtain the feedback contract. To assess the convergence to a solution the mediator also computes the *group distance* as the OWA-weighted distances to the feedback contract. While the purpose of HC is to avoid zones of no agreement, the aim of using OWA operators is to apply a predefined consensus policy.

Our goal is to elicit a function M, the *mediation rule*, which takes $\vec{c}_k$, D_{kc} and R_{kc} in order to obtain a feedback contract following a consensus policy. M describes the process of combining the individual agents' preferences. Our final objective is to define consensus policies in the form of a linguistic agenda. For example, the mediator could make decisions following mediation rules like *"Most* agents must be satisfied by the contract".

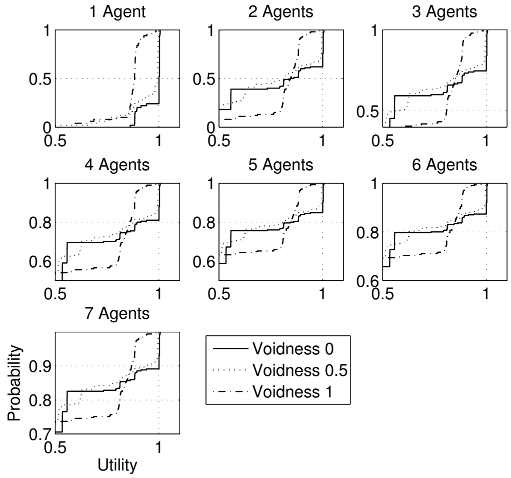


Figure 1: Cumulative distributions of utilities for the *complex negotiation scenario.*

The above statements are examples of *quantifier guided aggregations*. Any relative linguistic quantifier can be expressed as a fuzzy subset Q of the unit interval $I = [0, 1]$ [3]. It has been shown [3] that the OWA weights can be parametrized using this kind of functions. Under the quantifier guided mediation approach a group mediation protocol is expressed in terms of a linguistic quantifier Q indicating the proportion of agents whose agreement if necessary for a solution to be acceptable. First, we will express the mediation rule using the proper Q and then we will derive the OWA weights from Q. One feature which distinguishes the different types of mediation rules is the power of an individual agent to eliminate an alternative. In order to capture this idea, we use the *Value Of Individual Disapproval* (VOID) [3], which is defined as $VOID(Q) = 1 - \int_0^1 Q(y)dy$.

Finally, the feedback contract at round k is generated in the direction pointed by $\vec{v}$ from the origin $\vec{c}_k$, where vector $\vec{v}$ results from applying the vectorial OWA operator to the direction vectors. The distance at which the feedback contract is generated is obtained by applying the scalar OWA operator to the distances to the centroid. The group distance is a measure of closeness to an agreement. We take the distance to the offers in the cluster from the feedback contract to estimate the group distance. Again, we use W to OWA-weight the distance estimate and consider the consensus policy. If the group distance falls below a threshold, the negotiation ends with an agreement on the feedback contract.

4. EXPERIMENTAL EVALUATION

In the first experimental setup we have considered 7 agents. Utility functions are built using an aggregation of two randomly located *Bell functions*.The radius and height of each bell are randomly distributed within the ranges $r_i \in [20, 35]$ and $h_i = [0.1, 1]$. The probability for an agent to concede (i.e. to attend exclusively the feedback contract) is modelled for each agent using a probability value obtained from a uniform distribution between 0.25 and 0.5. We tested the performance of the protocol for 3 different consensus policies using the quantifier $Q_p(y) = y^p$.

Each experiment consist of 100 negotiations where we capture the utilities achieved by each agent. To analyze the results we first build a 7 agents$\times$100 negotiations utility matrix where each row provides each agent's utilities and each column is a negotiation. The matrix is then reorganized such that each column is individually sorted from higher to lower utility values. Given the matrix, we form 7 different utility groups: a first group named *group level 1* where we take the highest utility from each negotiation (i.e. the first row), a second group named *group level 2* with the two first rows and so on. We have used the Kaplan-Meier estimate of the cumulative distribution function (*cdf*) of agents' utilities for each group. The *cdf* estimates the probability of finding agent's utilities below a certain value. The rationale behind using grouping in the analysis is to evaluate the ability of the protocol to find solutions which satisfy groups of agents.

The results also show that as VOID increases, the mediator biases the search for agreements where more agents are satisfied at the expense of the individual satisfaction level. In general, it is worth noting that the application of a consensus policy may incur in a cost in terms of social welfare. In a second experimental setup we have considered 7 agents, 2 issues and 4 different types of negotiation spaces in increasing complexity to evaluate this issue. Figure 2 shows the social welfare measurements (sum of utilities) for different VOID degrees.

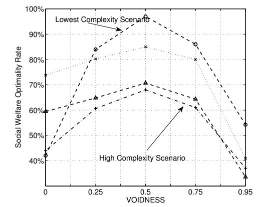


Figure 2: Social Welfare Optimality Rate vs VOID.

5. CONCLUSION

The negotiation framework presented opens the door to a new set of negotiation algorithms where consensus criteria may play an important role. HCPMF allows to perform multiparty negotiations where mediator guides the joint exploration of a solution by using aggregation rules which take the form of linguistic expressions. These rules are applied over the agents' offered contracts in order to generate a feedback contract which is submitted to the agents in order to guide their exploration, using HC To avoid zones of no agreement the mediator. We showed empirically that HCPMF efficiently manages negotiations following predefined consensus policies, which has been modelled using OWA operators.

6. REFERENCES

[1] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124:191–207, 2000.

[2] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[3] R. Yager and J. Kacprzyk. *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, 1997.

An Information Sharing Algorithm For Large Dynamic Mobile Multi-agent Teams

(Extended Abstract)

Linglong Zhu[†], Yang Xu[†], Paul Scerri[‡], Han Liang[†]
[†] University of Electronic Science and Technology of China,
[‡] Robotics Institute of Carnegie Mellon University
xuyang@uestc.edu.cn, pscerri@cs.cmu.edu

ABSTRACT

In large-scale multi-agent systems, communicating effectively is necessary for agents to cooperatively achieve joint goals. Despite significant progress on the multi-agent information sharing problem, existing research has not adequately dealt with the case of very large teams coordinating using a wireless network with changing team structure and density, where messages are *broadcast* to multiple members of the team. In this paper, we developed a compact and effective information sharing approach for teams with a dynamically changing, broadcast communication medium. By using a matrix representation of information status, the network structure and information needs, the model allows efficient reasoning about communication in a single computation. Empirical simulation results show that the approach performs well in large team, and effectively balances sharing key information with minimizing communication costs.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Communication, Broadcast, Teamwork, Decision-making.

1. MOTIVATION

Large teams of mobile robots are an attractive, emerging approach to a range of interesting applications. Typically, communication is required for best performance especially in complex environments. The exact medium that robots use to communicate varies from domain to domain, but will typically consist of some sorts of wireless broadcast within a local area, with robots required to rebroadcast messages to have the information reach its consumers. In many robot teams, the available bandwidth will be dramatically less than the volume of potentially useful messages. In this paper, we model the information sharing problem on an ad hoc wireless network with dynamically changing network structure and density.

A large number of distributed agents $\{a_1, .., a_i, ...\}$ in a team A are required to move around to observe or gather information in environment to act towards their common goal. $I = \{I_1, .., I_j, ...\}$ represents the available discrete pieces of information. Agents communicate via a wireless network $N(t) = \bigcup_{a \in A(t)} n(a,t)$, where $n(a,t)$ is defined as all agents b who have positive probability $Pr(a,b)$ of getting a broadcast message from agent a at time t which depends on the communication medium, signal strength and physical distance between agents but is independent of the information being communicated. When a set of related information $g_i = \{I_{i1}, I_{i2}, ..., I_{ik}\}$ comes to a single agent, a rational joint activities can be carried out in the team toward a reward $R(g_i)$. Information sharing is when an agent gets some information, how team members decide whether to broadcast or rebroadcast it on the network to make the best tradeoff between sharing information to get team reward and minimizing the communication cost.

2. SHARING MODEL WITH BROADCAST

To share information over broadcast media in large multi-agent teams, agents independently make decision on broadcasting information they have so that the team reward can be maximized. We use a simple matrix-based calculation, called *State-Communication-Reward* (SCR) that can be done distributedly, but approximates the complex decision calculation. In this matrix model, one matrix encodes the state of the team (S), one encodes the communication network (C) and one encodes the rewards for agents receiving specific information (R). A single multiplication of these matrices and a comparison to the current communication cost is all an agent needs to do to decide what to communicate. Instead of the traditional decisions which have to decide whether to broadcast the information in the sending queue piece by piece, this is a lightweight way of making the complex communication calculation for each agent.

Agent a's local model of deciding whether to broadcast a piece of information I_h is written as $< S, \Sigma_a^{I_h}, T, R >$. State matrix $S : A \times I_h$ models information distribution over the team. Specifically, the global team state consists of all the local states of each agent, $S = \bigcup_{a \in A} L_a$ where L_a represents the local state of information agent a has received or sensed. $\Sigma_a^{I_h} : I_h \rightarrow \{1, 0\}$ denotes the action that agent

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

a broadcasts I_h. The value of $\Sigma_a^{I_h}$ is 1 if the information is broadcasted by a, otherwise, the value is 0. $T : S \times \Sigma_a^{I_h} \times S \rightarrow [0, 1]$ models the transition function to S' when a executes $\Sigma_a^{I_h}$ on S. The transition probability is purely based on how agents are connected, whatever the information content is, so $T(L_{a_i}, \Sigma_a^{I_h}, L'_{a_i}) = Pr(L'_{a_i} | L_{a_i}, \Sigma_a^{I_h}) = Pr(a, a_i)$. To capture agents' view of the network, we define the matrix $C : A \times A \rightarrow [0, 1]$, where each element $C[a_i][a_j]$ represents agent a's estimate of whether a link exists between a_i and a_j, $C[a_i][a_j] = Pr(a_i, a_j)$. Agent a's decision is to take an optimal policy to the next team states that can maximize the team utility, $\pi^* = argmax_{\Sigma_a^{I_h}}(EU(S') - EU(S))$. When a broadcasts I_h, only agents in a's coverage can potentially get it, and the expected utility depends on the needs of all of potential receivers a_i which is based on what information a_i has.

$$EU(\Sigma_a^{I_h}) = \sum_{a_i \in n(a,t)} Pr(a, a_i) \cdot (EU(L_{a_i} \cup I_h) - EU(L_{a_i}))$$

For example, $g_5 = \{I_3, I_6, I_7, I_9\}$, $R(g_5) = 100$, $L_{a_i} = \{I_3, I_6, I_9\}$ and the expected utility of a given information set is a value iteration of $R(g_i)$, say, $EU(L_{a_i}) = 60$ is a value iteration of $R(g_5) = 100$. $EU(L_{a_i} \cup I_7) - EU(L_{a_i}) = 40$ denotes that the expected utility credited to the team is 40 when a_i receives I_7. Therefore, we setup a reward matrix $R : I \times S \rightarrow \mathbb{R}$ where each element $R[I_h][L_a]$ defines the expected reward of receiving I_h when agent a's local information set is L_a.

By using C to determine the state transition function T, the compact decision model SCR is written as: $< S, C, R, \Sigma >$. The complex information sharing decisions can be substituted with a simple matrix computation of S, C and R. The expected utilities of all possible broadcast decisions for agents in team A can be calculated as: $U = C \cdot S \cdot R$ where each element $U[a][I_h]$ denotes the expected utility for agent a to send I_h. On broadcast media, agents must balance better between providing useful information and causing network congestion. The actual cost of communication is very small, written as *sendcost*, and the real cost is in overloading the network and preventing other information getting through. To address this problem, we consider message collision caused by heavy traffic and model the cost as: $commcost^t = reccost \cdot p^t_{coll} + sendcost$ where *sendcost* and *reccost* are constants predefined according to the actual medium. According to the research in literature [1], we assume that agent can locally estimate the collision probability p^t_{coll} based on the number of messages it currently receives. A piece of information will be broadcasted if its expected utility is higher than the cost it occurs. When the information utilities are fixed, our dynamic model of communication cost makes the sender keep silent if the network is busy but broadcast the information when the network is relatively idle. In this way, SCR model can be adaptive to the dynamically changing network traffic and make the most use of limited network bandwidth.

3. EXPERIMENTAL RESULT

Our experiment simulates a group of 100 decentralized robots executing tasks. These robots expanded from a 150^2 to a 400^2 units area of region. 500 pieces of randomly distributed information were available to be sensed and communicated. Robots communicated with each other by broadcasting within a circle. The receive probability was inversely proportional to the distance between two robots, $P_{rec} = 1 - d/120$ where $d \in [0, 120]$. The relationship between pieces of information and their corresponding reward values were predefined according to power law distribution[1]. Moreover, we set $sendcost = 10, rececost = 10$.

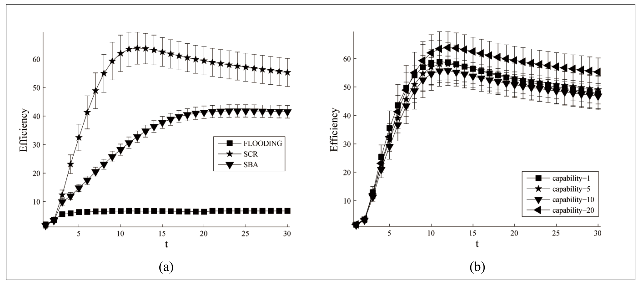


Figure 1: Sharing information with 100 robots. (a) The efficiencies of different algorithms. (b) The efficiencies of SCR algorithm when robots have one, five, ten and twenty different types.

The ratio of accumulative reward and corresponding number of messages sent by the team, which is $efficiency = R_{get}/num_{send}$, measures the algorithm's performance of balancing between sharing information and minimizing communication costs. Figure 1a describes the sharing performances of 100 robots with 20 different types by using the Flooding [2], SBA [3] and our SCR algorithm. The reward matrixes were constructed differently for robots with different capabilities according to $P(k_2)$[1]. As shown in Figure 1a, the efficiency of SCR algorithm is far better than Flooding and SBA. The reason is that robots' local SCR models effectively constrain broadcast of useless messages. Figure 1b shows that the efficiency of SCR algorithm changes little with more and more heterogeneous robot teams.

4. ACKNOWLEDGEMENTS

This research has been sponsored in part by National Natural Science Foundation of China 60905042 and 60950110354, and Research Funds for Central Universities ZYGX2011X013.

5. REFERENCES

[1] Y. Tay and Chua C. A capacity analysis for the IEEE 802.11 MAC protocol. *Wireless networks*, 7, 2001.

[2] C. Ho and et al. Flooding for reliable multicast in multi-hop ad hoc networks. In *Discrete algorithms for mobile computing and communications*, 1999.

[3] W. Peng and X.C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Mobile and Ad Hoc Networking and Computing*, 2002.

[1]Distribution $P(k_1) = 0.3868{k_1}^{-1.3}$ describes the percent of information which has relation with k_1 pieces of other information and $k_1 \in [1, 20]$. The information sets can be constructed according to this distribution when the number of information is known. The reward value of information sets k_2 is deferred to distribution $P(k_2) = 0.3769{k_2}^{-1.3}$ where $k_2 \in [1, 200]$.

Global Constraints in Distributed Constraint Satisfaction

(Extended Abstract)

Christian Bessiere
Université Montpellier 2, LIRMM-CNRS
Montpellier, France
bessiere@lirmm.fr

Ismel Brito, Patricia Gutierrez, Pedro Meseguer
IIIA, CSIC, Universitat Autonoma de Barcelona
Bellaterra, Spain
{ismel|patricia|pedro}@iiia.csic.es

ABSTRACT

Global constraints have been crucial for the success of centralized constraint programming. Here, we propose the inclusion of global constraints in distributed constraint satisfaction. We show how this inclusion can be done, considering different decompositions for global contraints. We provide experimental evidence of their benefits on several benchmarks solved with the ABT algorithm.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Distributed constraint satisfaction, global constraints

1. INTRODUCTION

Global constraints have been crucial in the development of efficient constraint solvers [5]. They allow to capture global properties on an unbounded set of variables. In many cases, the exploitation of the semantic associated with each global constraint allows to codify propagators able to reach local consistency levels (typically generalized arc consistency, GAC) with polynomial complexity. This is a great advantage with respect to GAC propagators for generic non-binary constraints, which have complexity exponential in the constraint arity.

Often, it is implicitly assumed that distributed constraint reasoning precludes the use of global constraints. With the usual assumption that each agent contains a single variable (so agents and variables can be used interchangeably), an agent knows the constraint with each one of its neighbors, and nothing else [6]. These constraints are obviously binary. But this interpretation is too restrictive because there are distributed applications for which it is natural to use global constraints.

When adding global constraints in distributed reasoning we obtain several benefits. First, the expressivity of distributed constraint reasoning is enhanced since there are relations among several variables that cannot be expressed as a conjunction of binary relations (most global constraints are not binary decomposable). Second, the solving process can be done more efficiently. Local consistency can be more efficiently achieved when global constraints are involved [5]. Assuming a solving strategy maintaining some kind of local consistency, using global constraints improves its efficiency.

Accepting the interest of global constraints in distributed constraint reasoning, another question naturally follows: since some global constraints can be decomposed in simpler constraints, is it more efficient, to leave the global constraint as it was initially posted or to decompose it? If several decompositions are possible, which offers the best performance? We provide some answers to these questions, exploring two decompositions (binary [1] and nested for contractible constraints [4]) against the global constraint without decomposition, in two contexts: complete distributed search with / without unconditional GAC maintenance [3].

We assume that readers are familiar with constraint reasoning, specially with distributed constraint satisfaction problems (DisCSP) and the ABT algorithm [6].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ADDING GLOBAL CONSTRAINTS

A global constraint C is a class of constraints defined by a Boolean function f_C whose arity is not fixed. Constraints with different arities can be defined by the same Boolean function. For instance, *alldifferent*(x_1, x_2, x_3) and *alldifferent* (x_1, x_4, x_5, x_6) are two instances of the *alldifferent* global constraint, where $f_{alldifferent}(T)$ returns true iff $x_i \neq x_j, \forall x_i, x_j \in T$. A global constraint C is *contractible* iff for any tuple t on $x_{i_1}, \ldots, x_{i_{p+1}}$, if t satisfies $C(x_{i_1}, \ldots, x_{i_{p+1}})$ then the projection $t[x_{i_1}, \ldots, x_{i_p}]$ of t on the first p variables satisfies $C(x_{i_1}, \ldots, x_{i_p})$ [4]. A global constraint C is *binary decomposable without extra variables* iff for any instance $C(T)$ of C, there exists a set S of binary constraints involving only variables in T such that the solutions of S are the solutions of $C(T)$ [1]. S is a *binary decomposition* of $C(T)$. In the following, we write C for a global constraint, while $C(T)$ means a particular instance of that global constraint on the set of variables T.

We consider three different representations for a global constraint instance: *direct*, *nested* and *binary*. In the *direct representation*, $C(T)$ is posted as a single constraint that allows all tuples on T satisfying C. Each agent in T includes $C(T)$ in its constraint set. The *nested representation* is applicable to all contractible global constraints. The nested representation of $C(T)$ with $T = (x_{i_1}, \ldots, x_{i_p})$ is the set of constraints $\{C(x_{i_1}, \ldots, x_{i_j}) \mid j \in 2 \ldots p\}$. Each agent in T includes all constraints of the nested representation of $C(T)$ that involve its variable in its constraint set. The *binary representation* is applicable to all global constraints that are binary decomposable. The binary representation of $C(T)$ is the set of constraints of its binary decomposition. Each agent in T includes all constraints of the binary decomposition of $C(T)$ that involve its variable in its constraint set. The three representations for the

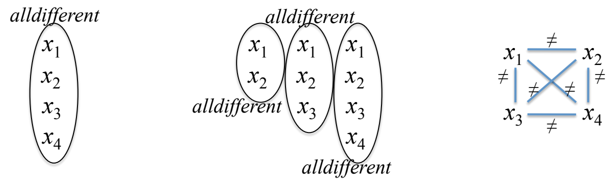


Figure 1: Representations for *alldifferent*(x_1, x_2, x_3, x_4): (left) direct, (center) nested, (right) binary.

alldifferent(x_1, x_2, x_3, x_4) global constraint appear in Figure 1.

Considering ABT as the solving algorithm, it is worth noting that ABT –originally proposed for binary constraints– can be easily generalized to handle constraints of any arity [2]. We assume that our ABT version contains such generalization.

In the direct representation, $C(T)$ is posted as a single constraint. Each agent in T knows it. The lowest priority agent of T in the ABT order is in charge of evaluating it. Other agents in T put a link between themselves and that agent. In the nested representation, $C(T)$, $T = (x_{i_1}, \ldots, x_{i_p})$, is represented by the set of constraints $\{C(x_{i_1}, \ldots, x_{i_j}) \mid j \in 2 \ldots p\}$. Thanks to the extra constraints that are posted, the checking of $C(T)$ is not postponed to the last agent in T. In the binary representation, $C(T)$ is represented by the set of constraints of its binary decomposition. These three representations of a global constraint instance are equivalent from the semantic point of view (they produce the same solutions). But they cause different ABT executions, so they can be seen as different models with dissimilar efficiency.

3. PROPAGATING GLOBAL CONSTRAINTS

Independently of the way a global constraint is included into ABT, this algorithm can be enhanced maintaining some form of local consistency during search. This was already investigated in [3], where limited/full forms of arc consistency (AC) were maintained during ABT execution for binary DisCSPs. While in [3] a limited form of AC causing unconditional deletions and full AC causing conditional deletions were considered, in this paper we only maintain the limited form of GAC that causes unconditional deletions (GAC because constraints may have arity higher than 2). Clearly, this limited GAC, that from now on we call UGAC, is less powerful than full GAC. Maintaining full GAC in the distributed context would cause a substantial load of extra messages which could overcome the benefits of domain pruning. We enforce UGAC on each considered global constraint by adapting the methods achieving GAC on them –developed in the centralized case– to this distributed setting, making them work inside each agent.

Before search, a suitable preprocess makes the problem GAC (before search any value deletion is unconditional, so GAC is equivalent to UGAC). During search, UGAC is enforced as follows: in ABT execution, if agent $self$ receives a nogood message justifying the removal of its value v where the nogood has an empty left-hand side (see [6, 3] for details), v can be unconditionally deleted from its domain. A deletion in the domain of x_{self} is propagated maintaining UGAC on the constraints connecting x_{self} with other variables, which may cause further deletions. Since the initial deletion is unconditional, deletions caused by the propagation are also unconditional.

To maintain UGAC during ABT search, some modifications are needed over the ABT algorithm: (1) the domain of variables constrained with $self$ has to be represented in $self$; (2) only the agent owner of a variable can modify its domain; if agent i deduces that a value could be deleted from the domain of x_j, it does nothing because that deduction will be done by agent j at some point; (3) there is a new message DEL to notify of value deletions: DEL$(self, k, v)$ –informing that $self$ removes v from the domain of x_{self}– is sent from $self$ to every agent k constrained with it; (4) a suitable preprocess makes all constraints GAC before ABT starts. These changes do not modify ABT correctness and completeness.

4. EXPERIMENTS AND SUMMARY

We evaluated the impact of the addition of global constraints on random binary DisCSPs including instances of *alldifferent* and *atmost* global constraints. For ABT on DisCSPs with loose binary constraints, the most efficient representation is the binary one, followed by nested and finally direct. For ABT on DisCSPs with tight binary constraints, the most efficient representation is the direct one, followed by nested and finally binary. The same pattern appears considering both the number of exchanged messages and the number of non-concurrent constraint checks (NCCCs). For ABT-UGAC, enforcing UGAC propagation during search causes a drastic efficiency improvement from medium to high tightness, while the ranking of representations remains the same.

As summary, in this paper we propose the use of global constraints in distributed constraint reasoning, considering three different ways to represent global constraints. We evaluate the performance of ABT with or without UGAC maintenance on random DisCSPs containing some global constraints. We conclude that UGAC propagation of global constraints is never harmful in terms of messages, and in some cases it can significantly reduce the search space. Regarding the different representations of global constraints, the direct representation often is the less efficient one. For DisCSPs with loose binary constraints, the binary representation wins but for DisCSPs without solution, this representation degrades quickly generating too many nogood messages. The nested representation seems to offer a good compromise: it is never worse than direct, and in some cases it is better than binary. This is good news: there are many more constraints that are contractible than constraints that are binary decomposable.

5. ACKNOWLEDGMENTS

Christian Bessiere is partially supported by the FP7-FET ICON project 284715. Christian Bessiere and Pedro Meseguer are partially supported by the CNRS-CSIC integrated action 2010FR0040. Ismel Brito, Patricia Gutierrez and Pedro Meseguer are partially supported by the project TIN2009-13591-C02-02. Patricia Gutierrez has an FPI scholarship BES-2008-006653.

6. REFERENCES

[1] C. Bessiere and P. Van Hentenryck. To be or not to be ... a global constraint. In *Proc. CP-03*, pages 789–794, 2003.

[2] I. Brito and P. Meseguer. Asynchronous backtracking for non-binary DisCSP. In *ECAI-06 DCR workshop*, 2006.

[3] I. Brito and P. Meseguer. Connecting ABT with arc consistency. In *Proc. CP-08*, pages 387–401, 2008.

[4] M.J. Maher. Open contractible global constraints. In *Proc. IJCAI-09*, pages 578–583, 2009.

[5] W. J. van Hoeve and I. Katriel. *Global Constraints*, chapter 6 of Handbook of Constraint Programming, pages 169–208. Elsevier, 2006.

[6] M. Yokoo, E. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Tr. Know. Data Engin.*, 10:673–685, 1998.

Multi-Agent A* for Parallel and Distributed Systems

(Extended Abstract)

Raz Nissim
Ben-Gurion University of the Negev
Be'er-Sheva, Israel
raznis@cs.bgu.ac.il

Ronen I. Brafman
Ben-Gurion University of the Negev
Be'er-Sheva, Israel
brafman@cs.bgu.ac.il

ABSTRACT

Search is among the most fundamental techniques for problem solving, and A* is probably the best known heuristic search algorithm. In this paper we adapt A* to the multi-agent setting, focusing on multi-agent planning problems. We provide a simple formulation of multi-agent A*, with a parallel and distributed variant. Our algorithms exploit the structure of multi-agent problems to not only distribute the work efficiently among different agents, but also to remove symmetries and reduce the overall workload. Given a multi-agent planning problem in which agents are not tightly coupled, our parallel version of A* leads to super-linear speedup, solving benchmark problems that have not been solved before. In its distributed version, the algorithm ensures that private information is not shared among agents, yet computation is still efficient – sometimes even more than centralized search – despite the fact that each agent has access to partial information only.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Distributed Search, Parallel search, Multi-Agent Planning.

1. INTRODUCTION

A* is probably the most celebrated heuristic search algorithm. Its good theoretical properties make it the favorite algorithm when searching for a provably optimal solution. The main contribution of this paper is MA-A*, a multi-agent formulation of A*. MA-A* attempts to make the most of the parallel nature of the system, i.e., the existence of multiple computing agents, while respecting its distributed nature, when relevant, i.e., the fact that some information is local to an agent, and cannot be shared. It is not a shallow parallelization or distribution of A*, as some successful parallel implementations of A* [4]. Rather, it is structure-aware, using the distinction between local and globally relevant actions and propositions to focus the work of each agent, dividing both states and operators among the agents, and exploiting symmetries that arise from the multi-agent structure. Moreover, MA-A* reduces exactly to A* when there is a single agent, unlike existing multi-core search methods [2]. MA-A* comes in two flavors, a parallel one and a distributed one, that differ only in the nature of the heuristic functions used.

To evaluate MA-A* we apply it to a number of multi-agent planning problems, comparing its performance to the best current optimal centralized planner and to the best (non-optimal) distributed planner. In the parallel case, our preliminary experiments show super-linear speed-up, as opposed to sublinear speedup by the best parallel planner, on problems in which agents are not tightly coupled. This stems from the fact that our algorithm is able to exploit the internal structure of the problem, and not only the added computational power. Using this variant, we were able to solve a number of planning problems that were so far beyond the reach of the best centralized optimal planners, and show up to $\times 20$ speedup on problems solved by both systems. In the distributed case, the agents are constrained to use only information that is directly accessible to them, i.e., information about their own operators and non-private aspects of the operators of other agents. Thus, this variant is truly distributed, and private information is not shared. In that setting, one would hope that the distributed algorithm would do not much worse than the centralized one (which has access to all information, but less computing power). Here, we see that the lack of global information is costly. Yet, even now, as long as the system is somewhat decoupled, the distributed algorithm can outperform the centralized one.

2. MULTI-AGENT A*

A MA-STRIPS problem [1] for a set of agents $\Phi = \{\varphi_i\}_{i=1}^k$ is given by a 4-tuple $\Pi = \langle P, \{A_i\}_{i=1}^k, I, G\rangle$, where P is a finite set of propositions, $I \subseteq P$ and $G \subseteq P$ encode the initial state and goal, respectively, and for $1 \leq i \leq k$, A_i is the set of actions agent φ_i is capable of performing. Each action $a = \langle pre(a), eff(a)\rangle$ is given by its preconditions and effects.

The MA-STRIPS model distinguishes between private and public variables and operators. A *private* variable of agent φ is required and affected only by the actions of φ. An action is *private* if all variables it affects and requires are private. All other actions are classified as *public*. That is,

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

φ's private actions affect and are affected only by φ, while its public actions may require or affect the actions of other agents. For ease of presentation we assume that all actions that achieve a goal condition are considered *public*.

MA-A*, presented in algorithms 1-3, is a distributed variation of A*, which maintains a separate search space for each agent. Each agent maintains an *open list* of states that are candidates for expansion and a *closed list* of already expanded states. It expands the state with the minimal $f = g + h$ value in its open list. When an agent expands state s, it uses its own operators only. This means that two agents expanding the same state will generate *different* successor states.

Algorithm 1 MA-A* for Agent φ_i

while did not receive **true** from a solution verification procedure **do**
  for all messages m in message queue **do**
    process-message(m)
  $s \leftarrow extract - min(openlist)$
  expand(s)

Algorithm 2 process-message$(m = \langle s, g_{\varphi_j}(s), h_{\varphi_j}(s)\rangle)$

if s is not in open or closed list **or** $g_{\varphi_i}(s) > g_{\varphi_j}(s)$ **then**
  add s to open list **and** calculate $h_{\varphi_i}(s)$
  $g_{\varphi_i}(s) \leftarrow g_{\varphi_j}(s)$
  $h_{\varphi_i}(s) \leftarrow max(h_{\varphi_i}(s), h_{\varphi_j}(s))$

Algorithm 3 expand(s)

move s to closed list
if s is a goal state **then**
  broadcast s to all agents
  initiate verification of stable property $f_{lower-bound} \geq g_{\varphi_i}(s)$
  return
for all agents $\varphi_j \in \Phi$ **do**
  if the last action leading to s was public **and** φ_j has a public action for which all public preconditions hold in s **then**
    send s to φ_j
apply φ_i's successor operator to s
for all successors s' **do**
  update $g_{\varphi_i}(s')$ and calculate $h_{\varphi_i}(s')$
  if s' is not in closed list **or** $f_{\varphi_i}(s')$ is now smaller than it was when s' was moved to closed list **then**
    move s' to open list

Since no agent has complete knowledge of the entire search space, messages must be sent, informing agents of open search nodes relevant to them. Agent φ_i characterizes state s as relevant to agent φ_j if φ_j has a public operator whose public preconditions (the preconditions φ_i is aware of) hold in s. In principle, a relevant state *must* be sent to φ_j (and this is what A* would effectively do). However, in some cases, this can be avoided, and there is also some flexibility as to when precisely the message will be sent. We discuss these finer details later, and for now, assume a relevant state is sent once it is generated.

The messages sent between agents contain the full state s, i.e. including both public and private variable values, as well as the cost of the best plan from the initial state to s found so far, and the sending agent's heuristic estimate of s. When agent φ receives a state via a message, it checks whether this state exists in its open or closed lists. If it does not appear in these lists, it is inserted into the open list. If a copy of this state with higher g value exists in the open list, its g value is updated, and if it is in the closed list, it is reopened. Otherwise, it is discarded. Whenever a received state is (re)inserted into the open list, the agent computes its local h_φ value for this state, and assigns the maximum of its h_φ value and the h value in the received message.

Once an agent expands a solution state s, it sends s to all agents and initiates the process of verifying its optimality. When the solution is verified as optimal, the agent initiates the trace-back of the solution plan. This is also a distributed process, which involves all agents that perform some action in the optimal plan. When the trace-back phase is done, a terminating message is broad-casted.

Termination detection is done using Chandy and Lamport's *snapshot algorithm* [3], which enables a process to create an approximation of the global state of the system, without "freezing" the distributed computation.

In the parallel setting, MA-A* allows each agent complete knowledge of both private and public operators of all agents. Thus, all agents compute (and can share) a single, global heuristic function, meaning that $h_{\varphi_i}(s) = h_{\varphi_j}(s)$ for all agents $\varphi_i, \varphi_j \in \Phi$ and for all states s. In the distributed setting, we assume that agents have access to public information and their own private information only. Because each agent has different information, it must compute its own local heuristic function. Thus, each agent can compute its heuristic estimate using a domain description that contains its own actions, as well as all public actions projected to public variables. The algorithm is completely agnostic as to how the agent uses this description to compute its private heuristic function. This allows us great flexibility, since different agents may use different heuristics. In fact, this is the *essence* of distributed search – each agent is a separate entity, capable of making choices regarding how it performs search.

Detailed experimental results, as well as proof of correctness and optimality are available in a technical report [5].

3. REFERENCES

[1] R. I. Brafman and C. Domshlak. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, pages 28–35, 2008.

[2] E. Burns, S. Lemons, R. Zhou, and W. Ruml. Best-first heuristic search for multi-core machines. In *IJCAI*, pages 449–455, 2009.

[3] K. M. Chandy and L. Lamport. Distributed snapshots: Determining global states of distributed systems. *ACM Trans. Comput. Syst.*, 3(1):63–75, 1985.

[4] A. Kishimoto, A. S. Fukunaga, and A. Botea. Scalable, parallel best-first search for optimal sequential planning. In *ICAPS*, 2009.

[5] R. Nissim and R. Brafman. Multi-agent a* for parallel and distributed systems. Technical Report 12-03, Department of Computer Science, Ben-Gurion University of the Negev, 2012.

Partial Cooperation in Multi-agent Search

(Extended Abstract)

Roie Zivan⋆, Alon Grubshtein†, Michal Friedman⋆ and Amnon Meisels†
⋆Industrial Engineering and Management department,
†Computer Science department
Ben Gurion University of the Negev,
Beer-Sheva, Israel
⋆{zivanr,friedmmi}@bgu.ac.il
†{alongrub,am}@cs.bgu.ac.il

ABSTRACT

Multi-agent systems usually address one of two pure scenarios, completely competitive agents that act selfishly, each agent maximizing its own gain from the interaction or multiple agents that operate cooperatively in order to achieve a common goal.

The present paper proposes a paradigm for multiple agents to solve a distributed problem, acting partly cooperatively and keeping a limited form of their self-interest. The proposed framework has multiple agents solving an asymmetric distributed constraints optimization problem (ADCOP), where agents have different personal gains from any mutual assignment. Three modes of cooperation are proposed – Non-cooperative, Guaranteed personal gain, and λ-cooperation (where agents' willingness to suffer relative loss is parametrized by λ). The modes of cooperation are described, as well as their realization in search algorithms.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: [Multiagent systems]

General Terms

Algorithms, Experimentation

Keywords

Distributed Search, Cooperation, Self interest

1. INTRODUCTION

Most studies investigating multi agent systems consider either fully cooperative agents which are willing to exchange information and take different roles in the process of achieving a common global goal (cf. [1]), or self interested agents which are considered to be rational when they take actions that will increase their personal gains (cf. [3]).

When one considers the standard working environment in which employees perform tasks for the benefit of the organization they work for and get a pay check in return, it seems that this most common situation is not covered by any of the two models described above. The agents in this working environment are naturally self interested and often have the option to increase their own benefit within the organization, even when benefits are non monetary. However, the success of the organization, and ultimately of the agents themselves, requires that the agents act loyally to increase the organizational profit (e.g., optimize some global goal).

In such real world situations, agents need to collaborate in finding the best (or a good) solution to the problem in a global perspective, despite having personal goals which may be in conflict.

Combinatorial optimization problems in which agents have personal gains can naturally be represented as Asymmetric Distributed Constraint Optimization Problems (ADCOPs) [2].

Previous studies of ADCOPs considered full cooperation of the agents. In contrast, the scenarios described above have agents that are cooperative only when some conditions are satisfied. This generic situation of multi-agent complex interactions raises the need to investigate modes of collaboration for self-interested agents solving combinatorial problems.

The present paper focuses on two new and fundamental questions regarding asymmetric multi agent optimization:

- What are the basic modes of collaboration one can define for agents solving an ADCOP?
- What are the relevant search methods for exploiting such modes of collaboration?

To address these questions three degrees or categories of cooperation are proposed for agents that have different personal gains (interests) in an interaction process. The degrees are defined as a function of the personal outcomes that agents can expect of the process relatively to the expected result of a non-cooperative interaction and on their willingness to sacrifice for the common good.

In this study, the set of possible outcomes that can be reached in a search process and its dependency on the level of cooperation is investigated and a standard DCOP algorithm is adjusted in order to apply to the proposed model according to its different levels of cooperation.

2. PARTIAL COOPERATION

Three increasing degrees of cooperation are proposed for agents: Non-cooperative, Guaranteed Personal Benefit col-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

laboration (GPB) and λ-cooperation. These degrees of cooperation affect the possible outcome of an *Interaction Process* among agents in an ADCOP. An *Interaction Process* IP of n agents is a predefined sequence of events that upon termination has each agent select value assignments for its variables.

The expected outcome of an IP in the non-cooperative setting depends on the details of the interaction. For a multi-step interaction one can expect the end result to be some form of equilibrium if the problem includes such a state and the interaction process allows convergence to it. A GPB solution is defined relatively to the non-cooperative (NC) solution which serves as a baseline. In a GPB setting the outcome of a sequence of actions must be a state which is weakly superior for each agent (Pareto improves the outcome of the NC process).

λ-cooperation allows agents to consider solutions with high global quality, which are not a Pareto improvement of the baseline state. The λ-cooperation class is based on the amount of risk, in the form of personal losses, that agents are willing to undertake in order to satisfy the global objective of the organization. The following definitions are used for the definition of λ-cooperation.

If all agents in the λ-cooperation class have the possibility to approve or reject any outcome which is proposed by the interaction process, then $O^{feasible}$ defines the set of outcomes approved by all agents. However, if the interaction process requires agents to perform actions which can result in other outcomes, agents may not be willing to take any risk and perform these actions. In this case, the set of outcomes considered in the interaction process is a subset (may be empty) of $O^{feasible}$.

3. CONDITIONAL COOPERATIVE SEARCH

The simplest (and most restrictive) translation of λ-cooperation to the distributed search setting is that the set of actions that agent i is willing to perform during search includes only actions which cannot lead to an outcome whose quality is not within λ_i from the quality of the NC outcome for i.

This approach is rather conservative (risk averse). It limits to an extreme extent the outcomes that will be considered by the agents and can prevent the search process from exploring Pareto improving solutions of high quality. To overcome this shortcoming one can extend the definition of λ-cooperation search to include agents beliefs about the future actions taken by other agents. Agents exclude in their considerations states which they believe that will not be selected by other agents. This can allow agents to ignore the threat of undesirable outcomes that, according to their belief, have low probability.

Figure 1 presents the results for three versions of the Distributed Synchronous Branch and Bound algorithm when solving random minimization problems in comparison with the baseline solution and the optimal solution (which ignores the personal thresholds). The baseline non-cooperative solution was selected by using a simple greedy interaction process. The first version of the algorithm allows agents to reject any solution reached. Thus, the algorithm selects the best solution in global terms that satisfies all local λ thresholds. This version is termed *No-Commitment*. On the other hand, in the second version that is referred as *Full-Commitment* an agent must consider any outcome that may result from its action. The balanced version which is based

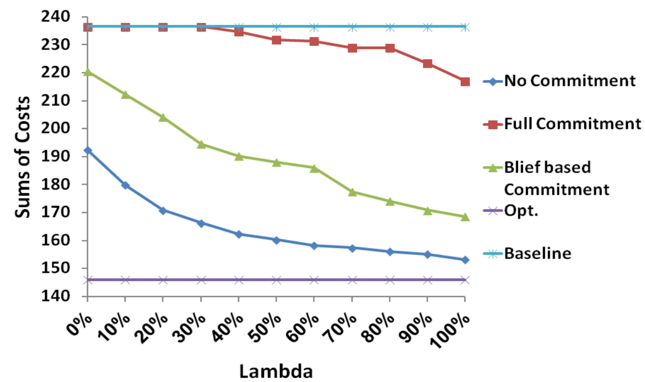


Figure 1: Solution cost of the Synch_BnB versions when solving random problems ($p_1 = 0.3$)

on the agent's belief is termed *Belief based Commitment*.

The global quality of all three versions of the algorithm improves when the λ value grows. The No-Commitment version produces solutions with lower global costs than the other two versions of the algorithm, although failing to find the globally optimal solution. Interestingly, the belief based version produces solutions whose costs are closer to the costs of the solutions found by the No-Commitment version than to the costs of solutions found by the Full-Commitment version.

4. DISCUSSION

A formalism that extends the ADCOP framework to include agents which are partially cooperative is proposed. Three modes of cooperation among agents were proposed - Non-Cooperative, Guaranteed Personal Benefits collaboration and λ-cooperation, where the willingness of agents to suffer a relative loss is parametrized by λ. The outcome of the non-cooperative mode serves as a baseline upon which the partial cooperative model is constructed. Agents seek alternatives which will satisfy their thresholds and improve the global outcome.

The set of possible solutions which are explored in the proposed partial cooperative model depends on the ability of agents to reject unsatisfying outcomes. If the search algorithm enables agents to reject unsatisfying outcomes, then the entire set of solutions which Pareto improve the non-cooperative baseline can be considered. On the other hand, if agents must commit to assignments they perform during search, they may refrain from assignments that lead to unsatisfying outcomes and thus, prevent the pursue of high quality solutions. A balanced compromise of these two extremes that is based on belief was proposed. Agents calculate the loss that they are expected to suffer and are thus able to ignore improbable outcomes with low quality.

5. REFERENCES

[1] A. Gershman, A. Meisels, and R. Zivan. Asynchronous forward bounding. *J. of Artificial Intelligence Research*, 34:25–46, 2009.

[2] A. Grubshtein, R. Zivan, T. Grinshpoun, and A. Meisels. Local search for distributed asymmetric optimization. In *AAMAS*, pages 1015–1022, 2010.

[3] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

Prioritized Shaping of Models for Solving DEC-POMDPs*

(Extended Abstract)

Pradeep Varakantham†, William Yeoh†, Prasanna Velagapudi‡, Katia Sycara‡, Paul Scerri‡
†School of Information Systems, Singapore Management University, Singapore 178902
‡Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15217, USA
†{pradeepv, williamyeoh}@smu.edu.sg ‡{pkv, katia, pscerri}@cs.cmu.edu

ABSTRACT

An interesting class of multi-agent POMDP planning problems can be solved by having agents iteratively solve individual POMDPs, find interactions with other individual plans, shape their transition and reward functions to encourage good interactions and discourage bad ones and then recompute a new plan. D-TREMOR showed that this approach can allow distributed planning for hundreds of agents. However, the quality and speed of the planning process depends on the prioritization scheme used. Lower priority agents shape their models with respect to the models of higher priority agents. In this paper, we introduce a new prioritization scheme that is guaranteed to converge and is empirically better, in terms of solution quality and planning time, than the existing prioritization scheme for some problems.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed AI

General Terms

Algorithms, Experimentation

Keywords

DEC-POMDP, Uncertainty, Multi-Agent Systems

1. INTRODUCTION

Cooperative multi-agent and multi-robot teams in domains such as sensor networks and disaster rescue [1, 2] require that agents plan courses of action that achieve their joint objectives. In complex domains, where agents are faced with many options, uncertainty and risk, finding good plans can be computationally extremely difficult. An interesting class of multi-agent POMDP planning problems can be solved by having agents iteratively solve individual POMDPs, find interactions with other individual plans, shape their transition and reward functions to encourage good interactions and discourage bad ones then recompute a new plan. One such algorithm, Distributed-Team's Reshaping of Models for Rapid Execution (D-TREMOR), has been shown to efficiently compute POMDP plans for hundreds of agents, an order of magnitude scale up over most centralized, joint POMDP planners [2].

However, the speed and quality of the D-TREMOR planning process depends on how the models are shaped with respect to possible interactions. In this paper, we look at the priority ordering of agents when they shape their models to improve interactions. The intuitive idea is to give order to the agents and make lower priority agents plan around the plans of the higher priority agents. In decentralized prioritized planning, the agents can plan simultaneously with conflicts in the plans resolved in favor of the higher priority agents. One prioritization scheme that is shown to work well is prioritize agents that are more valuable to the team. We have applied this same concept to D-TREMOR, creating an algorithm called PD-TREMOR. Specifically, priorities are dynamically set based on the expected utility the agent computes for its local plan. Although these values change at each iteration, at least one additional agent's priority is fixed to ensure convergence.

2. BACKGROUND

We employ the DPCL model [2] to represent the problems of interest in this paper. DPCL is similar to the DEC-POMDP model in that they are both represented by the tuple of $\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R}, \Omega, \mathbb{O} \rangle$, where $\mathbb{S}, \mathbb{A}, \Omega$ are the joint states, actions and observations, respectively, and $\mathbb{P}, \mathbb{R}, \mathbb{O}$ are the joint transition, reward and observation functions, respectively. The primary difference between DPCLs and DEC-POMDPs is that the interactions between agents in DPCL are limited to *coordination locales* (CLs). CLs represent situations where the actions of one agent affect the local transition and reward functions of other agents. There are two kinds of CLs: positive and negative CLs. Intuitively, *positive CLs* are CLs where the effects result in a positive gain in joint rewards. Conversely, *negative CLs* are CLs where the effects result in a negative gain in joint rewards. Formally, a CL is defined as the tuple of $\langle t, \{(s_i, a_i)\}_1^n \rangle$, where t is the decision epoch, s_i is the local state of agent i and a_i is the action taken by agent i. The set of CLs is computed from the joint transition and reward functions.

D-TREMOR [3] is a distributed DPCL algorithm, where each agent iteratively solves its individual POMDP, broadcasts its individual plan to all other agents, computes its own CLs, shapes its own POMDP model taking into account its *active* CLs, that is, CLs with a high probability of occur-

*This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Algorithm 1 PD-TREMOR(Agent i)

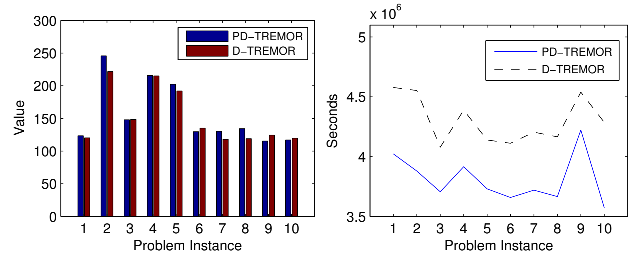
1: $\pi_i \leftarrow$ SolveIndividualPOMDP($\mathcal{M}_i$)
2: $iter \leftarrow 0$
3: **for all** $cl \in allCLs$ **do**
4: $\quad R^0_{i,cl} \leftarrow$ SetInitialPriority($\mathcal{M}_i, cl, \pi_i$)
5: **while** $iter < MaxIterations$ **do**
6: $\quad \alpha CLs \leftarrow$ ComputeActiveCLs($\mathcal{M}_i, allCLs, \pi_i$)
7: $\quad$ **for all** $cl \in \alpha CLs$ **do**
8: $\quad\quad val_{i,cl} \leftarrow$ EvaluateCL($\mathcal{M}_i, cl, \pi_i$)
9: $\quad\quad$ CommunicateCL($i, cl, pr_{i,cl}, val_{i,cl}, R^{iter}_{i,cl}$)
10: $\quad recCLs \leftarrow$ ReceiveCLs()
11: $\quad \mathcal{M}_i \leftarrow$ ShapeModel($\mathcal{M}_i, recCLs, \{R^{iter}_{i,cl}\}$)
12: $\quad \{\pi_i, val_i\} \leftarrow$ SolveIndividualPOMDP($\mathcal{M}_i$)
13: $\quad iter \leftarrow iter + 1$
14: $\quad$ **for all** $cl \in allCLs$ **do**
15: $\quad\quad R^{iter}_{i,cl} \leftarrow$ UpdatePriority($val_i, val_{i,cl}$)

rence, and repeats the above steps until convergence or for a maximum number of iterations. The agents shape their POMDP models in two steps: (a) the individual transition and reward functions are modified in such a way that the joint plan evaluation is equal (or nearly equal) to the sum of individual plan evaluations; and (b) incentives or hindrances are introduced in the individual agent models based on whether a CL accrues extra reward or is a cost to the team members. This incentive/hindrance is the difference in the value of the plan for the team with the CL. To ensure convergence, D-TREMOR employs two mechanisms – probabilistic shaping of agent models to resolve positive CLs and a prioritization scheme that determines which agent model to shape to resolve negative CLs.

3. PD-TREMOR

Unfortunately, the prioritization scheme used by D-TREMOR is rather ad-hoc. It is based on agent IDs, which are arbitrary. As a result, one can construct simple examples where the scheme can lead to arbitrarily bad results. We thus introduce a prioritization scheme that associates the priority of an agent with the expected value of its individual plan. The larger the expected value of an agent, the higher the priority of that agent. The intuition is that agents with small expected rewards should shape their models so that they can find individual plans with higher expected rewards. This scheme is dynamic across iterations since the individual plans can change across iterations. However, this scheme ensures that the priority of at least one agent is fixed at each iteration to ensure convergence. Thus, it takes at most n iterations to fix the agent models of all agents.

We implement this scheme over D-TREMOR and refer to the new extension as *Prioritized D-TREMOR* (PD-TREMOR). Algorithm 1 shows the pseudocode. The overall algorithm has the same distributed structure as the D-TREMOR, with each "self" agent i running in parallel to other agents in the system. Each agent i now maintains a priority $R^{iter}_{i,cl}$ for each CL cl and iteration $iter$. Each agent starts by computing its individual plan π_i assuming no other agents exist in the environment (line 1) and sets its initial priorities with the SetInitialPriority() function (lines 3-4). It then computes its active CLs (line 6) and for each active CL, it evaluates the effect of that CL (line 8) and broadcasts that information together with its priorities to the other agents (line 9). Upon receiving the



(a) Solution Quality (b) Runtime

Figure 1: Experimental Results

CL and priority information of all other agents (line 10), each agent shapes its model according to those priorities with the ShapeModel() function (line 11). Intuitively, for each CL, low priority agents shape their models in favor of higher priority agents. Finally, each agent solves its individual POMDPs with its newly shaped model (line 12) and repeats these steps for a maximum number of iterations (line 5).

4. EXPERIMENTS

We run experiments using the disaster rescue problem described in [2]. Each problem instance was solved once by PD-TREMOR and 5 times with (non-prioritized) D-TREMOR. As D-TREMOR contained probabilistic shaping heuristics, these multiple runs were necessary to measure characteristic performance, which was unnecessary for PD-TREMOR's deterministic prioritization heuristics. However, we show results for problems with *only* negative interactions only due to space constraints. Our algorithm failed to perform statistically better for problems containing positive interactions.

Figure 1(a) shows the expected value of the solutions (joint plans) generated by D-TREMOR and PD-TREMOR. As D-TREMOR is stochastic, its performance data is displayed as a boxplot over each of the 10 problem instances. While overall value was highly specific to individual map instances, PD-TREMOR's plans consistently matched and exceeded the value of average D-TREMOR plans on most maps. This suggests that dynamic prioritization offers competitive performance when resolving negative interactions. An additional benefit of the dynamic prioritization can be seen in Figure 1(b), a plot of the total time taken for 10 iterations of each algorithm. Here, the solid line represents PD-TREMOR, and the dotted line represents the time taken by D-TREMOR. In every case, PD-TREMOR is able to complete the same number of iterations faster, implying that it is performing a more efficient search.

5. REFERENCES

[1] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked Distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proceedings of AAAI*, pages 133–139, 2005.

[2] P. Varakantham, J. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *Proceedings of ICAPS*, pages 313–320, 2009.

[3] P. Velagapudi, P. Varakantham, P. Scerri, and K. Sycara. Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In *Proceedings of AAMAS*, pages 955–962, 2011.

Coordinated Look-Ahead Scheduling for Real-Time Traffic Signal Control

(Extended Abstract)

Xiao-Feng Xie, Stephen F. Smith, Gregory J. Barlow
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue,Pittsburgh, PA 15213
{xfxie,sfs,gjb}@cs.cmu.edu

ABSTRACT

We take an agent-based approach to real-time traffic signal control based on coordinated look-ahead scheduling. At each decision point, each agent constructs a schedule that optimizes movement of the currently approaching traffic through its intersection. For strengthening its local view, each agent queries the scheduled outflows from its direct upstream neighbors to obtain an optimistic observation, which is capable of incorporating non-local impacts from indirect neighbors. We summarize results on a road network of tightly-coupled intersections that demonstrate the ability of our approach.[1]

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Coherence and coordination*

General Terms

Algorithms

Keywords

Distributed Scheduling, Multi-Agent Coordination, Intelligent Transportation Systems, Real-Time Systems

1. INTRODUCTION

Intelligent traffic signal control presents the potential to substantially reduce congestion in road networks. However, how to achieve effective real-time control remains challenging [2]. Not only are the number of joint signal control sequences and local observations huge for just one intersection, but efficient flow of traffic through a road network also requires coordination among neighboring intersections.

Given the complexity and inherently distributed nature of real-time traffic signal control, we take an agent-based approach to solving this problem. We assume that each intersection is controlled by an agent using a schedule-driven intersection control strategy (SchIC) [4]. To strengthen the local views of individual agents and avoid myopic decisions, each agent asynchronously requests a projection of output flows from its direct upstream neighbors at each decision point to obtain an optimistic observation, which is capable of incorporating non-local impacts from indirect neighbors.

[1]This research was supported in part by the Traffic21 Initiative at Carnegie Mellon University, with support from the Hillman Foundation and the Heinz Endowments.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. PROBLEM DEFINITION

To keep the following description of our coordinated look-ahead scheduling as simple as possible, we focus on an one-way road network of signalized intersections. At each intersection, the traffic light cycles through a fixed sequence of phases I, where each phase $i \in I$ governs the right of way for a set of non-conflicting movements from entry to exit roads.

Each intersection is controlled by an agent that proceeds according to a rolling horizon [2–4], by holding a finite signal sequence SS_{TL}, and continually appending it with a short sequence (SS_{ext}) at each successive decision point. Each *signal sequence* contains a sequence of green phases and associated durations. Furthermore, SS_{TL} always satisfies the timing constraints for fairness and safety: each phase i has a variable duration (g_i) that can range between a minimum (G_i^{min}) and maximum (G_i^{max}), while the yellow light after each phase i runs for a fixed duration (Y_i).

For traffic signal control, the objective is to minimize the average delay of vehicles traveling through the road network.

3. INTERSECTION CONTROL

We adopt a schedule-driven intersection control (SchIC) strategy [4]. The basic idea is to form a *scheduling problem* using the current *observation* (o), particularly the *inflows* (IF) in the *prediction horizon* (H), and to generate a *schedule* that obtains a near optimal *control flow* (CF^*).

To achieve efficiency, we exploit an aggregate flow representation. Vehicles in a given non-uniform flow are organized using an ordered *cluster sequence* $C = (c_1, \cdots, c_{|C|})$, where $|C|$ is the number of clusters in C. Each *cluster* c is defined as $(|c|, arr, dep)$, where $|c|$ is the number of vehicles in c, and arr (dep) gives the expected arrival (departure) time at the intersection respectively for the first (last) vehicle in c.

An observation o contains the current decision time cdt, the current phase index cpi and duration cpd of SS_{TL}, and the inflows IF containing the currently sensed vehicles.

Formally, $IF = (C_{IF,1}, \cdots, C_{IF,|I|})$, where $C_{IF,i}$ is a cluster sequence containing the vehicles with the right of way during phase i. Clusters in each $C_{IF,i}$ are further aggregated into an anticipated queue and arriving clusters.

A control flow CF contains the results of applying a signal sequence that clears all clusters in an observation o. Formally, $CF = (S, C_{CF})$, where S is a sequence of phase indices, i.e., $(s_1, \cdots, s_{|S|})$, and C_{CF} contains a sequence of clusters $(c_{CF,1}, \cdots, c_{CF,|S|})$ that are reorganized from IF.

Algorithm 1 Obtain an optimistic non-local observation

1: $m = GetEntryRoadByPhase(i)$ {For each phase i}
2: $UpAgent = GetUpstreamAgent(m)$
3: Request C_{OF} from $UpAgent$ using (cdt, m, H_{ext})
4: $Shift(C_{OF}$, the travel time on $m)$
5: Append C_{OF} into $C_{IF,i}$

Algorithm 2 Return C_{OF} for a message (cdt, n, H_{ext})

1: $(C_{OF}, S_{OF}) = (C^*_{CF}, S^*) \cap [cdt, cdt + H_{ext}]$
2: **for** $k = |C_{OF}|$ to 1 **do**
3: $|c_{OF,k}| = |c_{OF,k}| \cdot tp(s_{OF,k}, n)$ {turning proportion}
4: **end for**

For any k, all vehicles in $c_{CF,k}$ belong to C_{IF,s_k}.

The *scheduling search space* is formed by viewing each cluster as a non-divisible job. The jobs in $C_{IF,i}$ can only leave the intersection when the phase index is i, and the jth job can only leave after the $(j-1)$th one has left. Each S is a *schedule* with $|S| = \sum_{i=1}^{|I|} |C_{IF,i}|$. For a partial schedule S_k (the first k elements of S), its *schedule status* is defined as $X{=}(x_1, \cdots, x_{|I|})$, where $x_i \in [0, |C_{IF,i}|]$. In the state update that adds s_k to S_{k-1}, we have $x_{s_k} = x_{s_k} + 1$, $c_{CF,k}$ comes from the x_{s_k}th cluster in C_{IF,s_k}, and the actual arrival time and cumulative delay of $c_{CF,k}$ are determined according to a greedy construction of the corresponding signal sequence [4].

The cumulative delay of CF^* is minimized by a dynamic programming process [4], which has $|I|^2 \cdot \prod_{i=1}^{|I|}(|C_{IF,i}| + 1)$ state updates in the worse case, where $|C_{IF,i}| \leq H$, and each state update can be executed in constant time. It is polynomial in H since $|I|$ is limited in the real world.

The first job in CF^*, if available, is used to determine SS_{ext}. There are two possible extension choices: 1) terminate the current green phase and move to the next (if $|S^*| \equiv 0$, or $s^*_1 \neq cpi$, or $arr(c^*_{CF,1}) \geq SwitchBack(cpi)$); or otherwise 2) extend the current phase, in which case $ext = \min(dep(c^*_{CF,1}) - cdt, th_{ext})$, where th_{ext} is the upper limit. A *repair rule* is applied lastly to ensure that SS_{TL} does not violate any time constraints after appending SS_{ext}.

4. BASIC COORDINATION MECHANISM

In a road network, an agent is susceptible to myopic decisions if its local prediction horizon is not sufficiently long. To counteract this possibility we extend each agent's local view with an optimistic non-local observation from its upstream agents, as shown in Algorithm 1. For each phase index i, the corresponding entry road m is identified, and the corresponding upstream agent $UpAgent$ is obtained. The agent then sends $UpAgent$ a request message (cdt, m, H_{ext}), where H_{ext} is the maximum *horizon extension*, for the planned output flow C_{OF} of $UpAgent$. Upon receipt of C_{OF}, the downstream agent adds an offset time — the average travel time between the two agents (intersections) — to all the clusters in C_{OF} and appends the clusters to the end of $C_{IF,i}$.

$UpAgent$ executes Algorithm 2 to obtain the output flow C_{OF} at the current time cdt, based the previously planned control flow (S^*, C^*_{CF}). The entry road m of the requesting agent is the exit road n of $UpAgent$. In Line 1, (C_{OF}, S_{OF}) is obtained as the subsequence of (C^*_{CF}, S^*) that belongs to the time period $[cdt, cdt + H_{ext}]$. In Line 3, $tp(i, n)$ is the portion of traffic turning onto exit road n during phase i.

An essential property of this protocol is that non-local impacts from indirect neighbors can be included if H_{ext} is sufficiently long, since the control flow of direct neighbors contains flow information from their upstream neighbors.

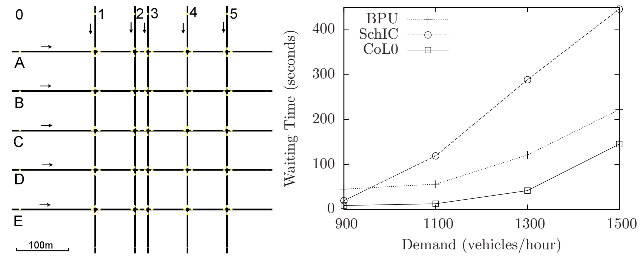


Figure 1: (a) 5X5 grid network; (b) Average Results.

The optimistic assumption that is made is that direct and indirect neighbors are trying to follow their schedules. The optimization capability of SchIC makes schedules quite stable. Minor schedule changes in neighbors can be absorbed by exploiting the temporal flexibility in their control flows.

5. RESULTS

We simulate performance using SUMO[2] on a 5X5 one-way grid network as shown in Figure 1 (a). In this network, all road lengths are 75 meters, except for the horizontal roads **2** → **3** and **0** → **1**, which are respectively 25 and 150 meters.

On each road, the free-flow speed is 10 meters per second. For each intersection, Y, G^{min} and G^{max} are respectively 5, 5, and 55 seconds. Because the minimal switchback time ($Y + G^{min} + Y = 15$ seconds) is longer than the travel time on one road (2.5 or 7.5 seconds), non-local impacts from indirect neighbors might be nontrivial and cannot be ignored.

Only through traffic movements are considered. For background traffic, each minor route generates a flow of 1/20 of the total traffic. There are two major flows on **C** and **3** that generate 3/5 of the total traffic. The total simulation time is one hour, and for each twenty minute period, the demand ratios between **C** and **3** are 35:25, 40:20, and 45:15.

Figure 1 (b) shows the average results of three control strategies, i.e., BPU, SchIC, and CoL0, for different demands. BPU (balanced phase utilization) [1] is an adaptive coordination strategy using offset calculation, SchIC is the isolated control strategy [4], and CoL0 applies the optimistic non-local observation ($H_{ext} = 15$ seconds) to SchIC.

CoL0 produced lower waiting times than both other strategies. Comparison to SchIC demonstrates the added benefit of optimistic non-local observation. Furthermore, CoL0 outperforms BPU without requiring explicit offset calculation; coordination between neighbors is instead accomplished implicitly by looking ahead to upstream output flows. Future work will explore the use of additional coordination mechanisms to address specific situations (e.g., queue spillbacks).

6. REFERENCES

[1] G. J. Barlow. *Improving Memory for Optimization and Learning in Dynamic Environments*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2011.

[2] M. Papageorgiou, C. Diakaki, V. Dinopoulou, and et al. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[3] S. Sen and K. Head. Controlled optimization of phases at an intersection. *Transport. Sci.*, 31(1):5–17, 1997.

[4] X.-F. Xie, S. F. Smith, L. Lu, and G. J. Barlow. Schedule-driven intersection control. Technical Report CMU-RI-TR-11-34, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2011.

[2] Simulation of urban mobility: http://sumo.sourceforge.net

Global Optimization for Multiple Agents (Extended Abstract)

Brammert Ottens, Boi faltings
Artificial Intelligence Laboratory, EPFL
Lausanne, Switzerland
firstname.lastname@epfl.ch

ABSTRACT

In many situations, agents optimize their own operations locally but their local problems are interdependent. We consider the problem of coordinating these local problems to find a globally optimal solution. We model the coordination problem as a Distributed Constraint Optimization Problem that can take advantage of the locality of interactions, and then show how incremental elicitation and solving techniques can minimize the effort required. We illustrate the approach on an example of coordinating logistics service providers, for example couriers in a delivery company.

1. INTRODUCTION

In any cooperative multi agent setting, agents need to coordinate their actions. Take as an example a logistics setting, where individual couriers pick up and deliver packets. The decision on who is *assigned* which packet must be coordinated, i.e no two couriers should try to pick up the same packet, because this leads to a waste of resources. Furthermore, no two *coordinated* assignments are the same, i.e. some assignments use the available resources more efficiently then others. Also, individual couriers could have individual preferences over different routes. They might prefer one restaurant over another for lunch, know which roads are more likely to be congested and so on and so forth. This latter information is difficult to formalize, and hence it would be best if each courier would be able to plan its own route. We thus have a *global* coordination problem with *local* preferences.

More generally, one can say that coordination problems consist of two highly interrelated parts: the local problems of each particular agent, that determine the preferences of individual packet assignments, and the coordination problem that determines which assignments are compatible. In this paper we argue that the DCOP paradigm [2] is a natural model for coordination problems. DCOP algorithms, however, have been designed without regard of the possible difficulty of solving the local problems of the agents. This paper evaluates a range of DCOP algorithms on problems with nontrivial local problems.

2. MULTI AGENT COORDINATION PROBLEMS

Many coordination problems are resource or task allocation problems: given a finite set of resources or tasks, distribute them over the agents. Often, however, multiple outcomes are feasible, and the agents have preferences over these outcomes. A solution to a coordination problem is thus a feasible outcome that maximizes the local preferences of the different agents.

An example of a coordination problem can be found in logistics. Being able to efficiently distribute goods using couriers has large practical value. The types of problems we are looking at in this paper are inspired by a project with a courier company in a large european city, where couriers make independent decisions about their routes. The particular model used in this paper is an adaptation of the Truck Task Coordination (TTC) problem as given in [4].

We take the perspective of a single company, consisting of a group of couriers, dispersed over a geographical area. Each courier has its own *garage*, from which it operates. Customers offer packets for pickup and delivery to the company, but there are restrictions on which couriers are allowed to service them: packets will only be offered to couriers whose garages are within a certain range of the pickup and delivery locations. Furthermore, the range of the couriers will also be limited. These two restrictions together make that not all possible packet assignments are feasible. The main difference with standard VRP problems [6] is that not all couriers are able to service all packets. Furthermore, the goal is to maximize utility and not to minimize driving distance, where the utility is defined as the payment obtained from delivering a set of packets, minus the cost incurred by driving.

The DCOP paradigm is well suited to model the TTC problem. Let $P_i \subseteq \mathcal{P}$ be the set of packets that has been offered to courier t_i, and let $T_j \subseteq \mathcal{T}$ be the set of couriers that have been requested to deliver packet p_j. Then for each courier t_i and every packet p_j such that $p_j \in P_i$ and $|T_j| > 1$, t_i owns a binary variable x^i_j. Packets p_j for which $|T_j| = 1$ are assumed to be delivered by this courier, if within *courierRange*. If $x^i_j = 1$, then t_i will service packet p_j, and if $x^i_j = 0$ it will not service it. If a packet is not serviced, a penalty γ is incurred. For each packet, this is modeled through a $|T_j|$-ary constraint, running over all variables that represent the particular packet. The coordination constraint must enforce that no two couriers will deliver the same packet. This is captured by the following hard constraint.

$$x^i_k + x^j_k \quad \leq \quad 1 \tag{1}$$

The local utility of an agent depends on the set of packets it is assigned, and on the route that it will take. Only packets that are within a certain distance are to be serviced. Let $distance(t_i, p_j) = true$ when both the pickup and delivery city of packet p_j are within *courierRange*, and $false$ otherwise. Then $ownPackets_i = \{p_j | distance(t_i, p_j) = true \wedge |T_j| = 1\}$ is the set of packets that only courier t_i can service, and $coordinationPackets_i = P_i \setminus ownPackets_i$ is the set of packets courier t_i needs to co-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

ordinate over. For every delivered packet p_j, a courier gets paid αw_{p_j}, where w_{p_j} is the weight of packet p_j and α is the payment per unit weight. Hence it is guaranteed a payment of $pr = \sum_{p \in ownPackets_i} \alpha w_p$. The cost of the route is provided by the local solver. If the assignment contains a packet whose pickup or delivery city is outside of the courierRange, or when the capacity constraint cannot be met, it is infeasible and this is indicating by setting the cost to ∞.

$$vrp_i(ownPackets_i, x^i_{j_1}, \ldots, x^i_{j_m}) \;=\; pr + \sum_{k=1}^{m} x^i_{j_k} \alpha w_{p_{j_k}} - \text{cost of the route} \qquad (2)$$

3. EXPERIMENTAL RESULTS

The main goal of the experiments presented in this section is to investigate the influence of the presence of non trivial local problems on the complexity of the coordination problem. Performance is measured using both simulated time [5] and Non Concurrent Constraint Checks (NCCC) [1].

3.1 Experimental Setup

We evaluated several DCOP algorithms on the TTC problem. We created a map of size 1000x1000, with 50 cities, 6 couriers and 16 packets randomly dispersed over the cities. The customer range and courier range are taken from $\{200, 250, 300, 350, 400\}$ and for each combination of parameters we generated a 101 instances. We evaluated DPOP, O-DPOP, DSA, MGM and MGM2 on all these problems, setting the penalty for querying the local problem at 60 seconds per query. The local solver used for O-DPOP is not guaranteed to generate packet assignments in a best first order, and any violations are recognized and the utilities capped. The local search algorithms are run until they converge.

All algorithms have been written in Java, and have been implemented in the Frodo [3] framework. All experiments have been run on a four core Intel xeon 3 Ghz machine running linux, and each run was allocated 2 Gb of internal memory.

3.2 Results

Due to space constraints we show only a fraction of the experimental results. Figure 1 shows the solution quality obtained by the algorithms, and Figure 2 shows the simulated runtime. First, note that the local search algorithms are only able to find good solutions for the simplest problems. Furthermore, the lack of results for local search algorithms on problems with a customer range of more than 350 is caused by the fact that they are not able to find feasible solutions. The more complex the problems become, the more suboptimal the found solutions are. As for runtime, it is clear that O-DPOP outperforms DPOP. For the smaller problems it is even faster than the local search algorithms, but never worse.

4. CONCLUSIONS

There is a wide range of methods for solving coordination problems. In this paper we show that the DCOP paradigm is a natural choice for modeling such problems. Experimental results show that DCOP methods allow one to take advantage of the problem structure to obtain optimal solutions with reasonable complexity.

Furthermore, the results show that if the problems do not ask for a great amount of coordination, i.e locally good solutions are part of globally good solutions, the incremental elicitation algorithm O-DPOP needs to perform little to no work to find the optimal allocation of goods. It even outperforms local search algorithms.

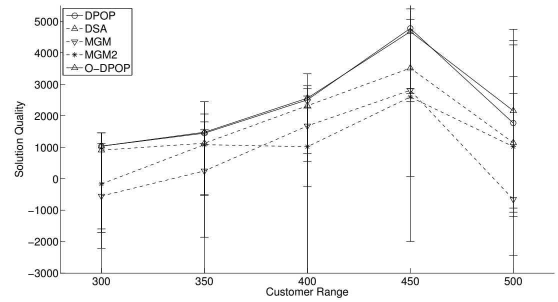


Figure 1: Utility for courier range of 400

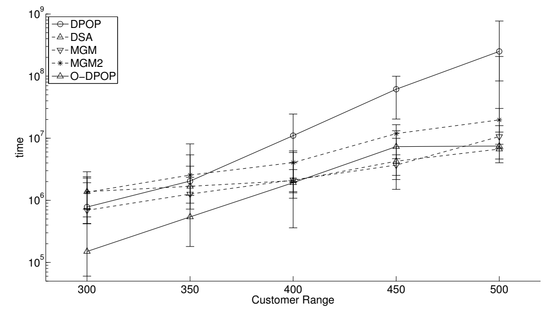


Figure 2: Simulated Runtime for courier range of 400

In future work, we plan to improve the preference elicitation scheme used here to make it more efficient and also allow for anytime performance. We are also considering more efficient data structures for task allocation that could improve the communication efficiency of the process.

5. REFERENCES

[1] Amir Gershman, Roie Zivan, Tal Grinshpoun, Alon Grubshtein, and Amnon Meisels. Measuring distributed constraint optimization algorithms. In *Proceedings of DCR'08*, pages 17–24, May 13 2008.

[2] Katsutoshi Hirayama and Makoto Yokoo. Distributed partial constraint satisfaction problem. In *CP'97*, pages 222–236, 1997.

[3] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek. FRODO 2.0: An open-source framework for distributed constraint optimization. In Katsutoshi Hirayama, William Yeoh, and Roie Zivan, editors, *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, pages 160–164, July 13 2009.

[4] Brammert Ottens and Boi Faltings. Coordinating agent plans through distributed constraint optimization. In *Proceedings of the ICAPS'08 Multiagent Planning Workshop (MASPLAN'08)*, September 14 2008.

[5] Evan A. Sultanik, Robert N. Lass, and William C. Regli. DCOPolis: A framework for simulating and deploying distributed constraint optimization algorithms. In Jonathan P. Pearce, editor, *Proceedings of the 9th Intl Workshop on Distributed Constraint Reasoning (CP-DCR'07)*, September 23 2007.

[6] Paolo Toth and Daniele Vigo, editors. *The Vehicle Routing Problem*. SIAM, 2001.

Scalable decentralized supply chain formation through binarized belief propagation

(Extended Abstract)

Toni Penya-Alba, Jesus Cerquides, Juan A. Rodriguez-Aguilar
IIIA - CSIC
Campus de la UAB, E-08193 Bellaterra, Spain
{tonipenya, cerquide, jar}@iiia.csic.es

Meritxell Vinyals
Department of Computer Science
University of Verona
Strada le Grazie, 15, Verona, Italy
meritxell.vinalssalgado@univr.it

ABSTRACT

Supply Chain Formation (SCF) is the process of determining the participants in a supply chain, who will exchange what with whom, and the terms of the exchanges. Decentralized SCF appears as a highly intricate task because agents only possess local information, have limited knowledge about the capabilities of other agents, and prefer to preserve privacy. Very recently, the decentralized SCF problem has been cast as an optimization problem that can be efficiently approximated using max-sum loopy belief propagation. Unfortunately, the memory and communication requirements of this approach largely hinder its scalability. This paper presents a novel encoding of the problem into a binary factor graph (containing only binary variables) along with an alternative algorithm. These allow to scale up to form supply chains in markets with higher degrees of competition.

Categories and Subject Descriptors

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

General Terms

Algorithms, Economics, Experimentation

Keywords

Supply chain, belief propagation, scalability

1. INTRODUCTION

According to [4], "Supply Chain Formation (SCF) is the process of determining the participants in a supply chain, who will exchange what with whom, and the terms of the exchanges". Although intractable [3], the SCF problem has been widely tackled by the multi-agent systems (MAS) literature, mainly through centralized auction-based approaches [5, 1]. Furthermore, as argued in [4], even when the computation is tractable, no single entity may have global allocative authority to compute allocations over the entire supply chain (SC). To overcome these limitations, a decentralized manner to solve the problem is proposed in [4]. More recently, Winsper et al. [6] cast the decentralized SCF problem into an optimization problem that can be approximated using max-sum loopy belief propagation (LBP)[1]. Unfortunately, the memory and communication requirements of this approach hinder its scalability.

In this paper we propose a novel approach to the decentralized SCF problem, the so-called Reduced Binary Loopy Belief Propagation (RB-LBP), that significantly outperforms LBP in terms of scalability.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. SCF PROBLEMS AS FACTOR GRAPHS

In LBP the SCF problem is casted into a factor graph composed of variables and factors. A single variable is created for each participant in the SC. The values (states) of each participant's variable encode the individual decisions that the agent needs to make regarding her exchange relationships plus an inactive state. For example, say that an agent needs to purchase a good to produce another one. Consider also that there are three possible producers for the requested good and three possible consumers for the produced good. Therefore, the agent's variable will have 10 states. That is, one for each of the producer-consumer combinations plus an inactive state. Notice that the number of states of an agent's variable grows exponentially with the number of agents and goods.

Agents' buying and selling prices are introduced by means of **activation factors**. Each agent has an activation factor that stores a value of zero whenever the agent is inactive and the agent's buying or selling price otherwise. Furthermore, in the factor graph, variables corresponding to potential partners are connected through a **compatibility factor**. Each of these factors encodes the compatibility between the decisions of the two agents involved. Two agents' decisions are incompatible whenever one of them is willing to trade with the other, but the other does not. Notice that the size of the compatibility factors is the product of the sizes of the variables it connects. Therefore, the memory needed by an agent to store factors grows exponentially to the number of agents and goods. Moreover, the messages exchanged between two agents encode their preferences over each other states. As a consequence, the communication requirements of LBP are also exponential to the number of goods and agents.

[1]We address the reader to [2] for a description of max-sum.

Measure	LBP	RB-LBP
Memory needed per agent to store the preferences over her state	$\mathcal{O}(A^G)$	$\mathcal{O}(G \cdot A)$
Size of largest factor	$\mathcal{O}(A^{2G})$	$\mathcal{O}(1)$
Maximum memory needed per agent (to store both preferences and factors)	$\boldsymbol{\mathcal{O}(G \cdot A^{2G+1})}$	$\boldsymbol{\mathcal{O}(G \cdot A)}$
Maximum message size	$\mathcal{O}(A^G)$	$\mathcal{O}(1)$
Maximum bandwith consumed per agent and iteration	$\boldsymbol{\mathcal{O}(G \cdot A^{G+1})}$	$\boldsymbol{\mathcal{O}(G \cdot A)}$
Overall consumed bandwith	$\mathcal{O}(n \cdot G \cdot A^{G+1})$	$\mathcal{O}(n \cdot G \cdot A)$

Table 1: Required resources: LBP vs. RB-LBP.

2.1 Scaling up supply chain formation

In order to cope with the scalability issues of LBP, we model the SCF problem as a binary factor graph containing only binary variables.

In this new model, each agent is aware of two sets of variables that encode her decisions to collaborate with potential partners. On the one hand, each agent encodes whether she is active (part of the SC) or not by means of an **activation variable**. On the other hand, each agent encodes her decision to trade a particular good with a particular producer/consumer using an **option variable**. Notice that the number of variables an agent needs to encode her decisions is linear to the number of possible exchanges she is involved in.

First, to guarantee that only one of the providers of a given good is selected, we make use of a **selection factor**. A selection factor links the activation variable from the agent with the option variables for that good. Second, we need to guarantee that the decisions from different agents are coherent among them. Thus, we add an **equality factor** constraining the seller's option variable and the buyer's option variable to be either both 1 or both 0. Notice that there is no need to store selection and activation factors in memory since they can be encoded as logical expressions.

Then, we show that, since we only employ binary variables and hard constraints, we can greatly reduce the computation required to assess messages. First, we only consider the configurations of variables that satisfy equality and selection factors. Second, instead of sending messages with a value for each of the two states of each variable, in RB-LBP messages contain the difference between these two values. Both changes together severely reduce the computation needed to assess messages. Moreover, since each agent only exchanges a single value with each of her neighbours, bandwidth requirements in RB-LBP scale linearly with the number of goods and agents.

Worst case memory and bandwidth requirements for both RB-LBP and LBP are summarized in table 1. A denotes the maximum number of agents connected to a good, G denotes the maximum number of goods an agent is interested in, and n stands for the total number of agents in the network.

3. EVALUATION

We benchmarked RB-LBP against LBP in the networks described by Walsh et al. in [4] and in larger networks with higher degrees of competition (in terms of number of providers offering each good). In the networks described by Walsh et al., RB-LBP requires from 2 up to 13 times less memory than LBP depending on the network structure. Moreover, the bandwidth consumed by an agent during an LBP iteration is up to 5 times larger than RB-LBP's.

For larger networks (up to 500 agents and 50 goods), LBP memory requirements are up to 5 orders of magnitude greater than for RB-LBP. Bandwidth usage for LBP is up to 787 times larger than for RB-LBP and, regarding computational time, RB-LBP is up to 20 times faster than LBP. Finally, the median SC value obtained by RB-LBP is up to 2 times greater than those obtained by LBP.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have described RB-LBP, a novel approach for decentralized SCF. We have shown both theoretically and experimentally that RB-LBP scales nicely to market scenarios with larger number of participants and increasing competition. Our experimental results show that RB-LBP can significantly reduce the usage of memory and communication several orders of magnitude with respect to LBP. Furthermore, RB-LBP produces up to two times higher value supply chains and has smaller time complexity. Therefore, RB-LBP allows to tackle large-scale decentralized SCF problems.

Up to date approaches for decentralized SCF [4, 6] can only be applied to networks where agents can produce at most a single good. In order to compare with previously existing approaches, all the experimental results in this paper are over this kind of networks. However, RB-LBP can readily be applied to scenarios where producers can deliver more than one good. Experimentally evaluating RB-LBP in these scenarios and over a variety of actual-world network structures is left as future work.

5. ACKNOWLEDGMENTS

This work has been funded by projects EVE (TIN2009-14702-C02-01), AT (CSD2007-0022), CSIC 201050I008, and the Generalitat of Catalunya (2009-SGR-1434).

6. REFERENCES

[1] J. Cerquides, U. Endriss, A. Giovannucci, and J. A. Rodriguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *IJCAI*, pages 1221–1226. Morgan Kaufmann Publishers Inc., 2007.

[2] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS (2)*, pages 639–646, 2008.

[3] V. Fionda and G. Greco. Charting the tractability frontier of mixed multi-unit combinatorial auctions. In C. Boutilier, editor, *IJCAI*, pages 134–139, 2009.

[4] W. E. Walsh and M. P. Wellman. Decentralized supply chain formation: A market protocol and competitive equilibrium analysis. *J. Artif. Intell. Res. (JAIR)*, 19:513–567, 2003.

[5] W. E. Walsh, M. P. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *2nd ACM Conference on Electronic Commerce*, EC '00, pages 260–269, New York, NY, USA, 2000. ACM.

[6] M. Winsper and M. Chli. Decentralised supply chain formation: A belief propagation-based approach. In *ECAI*, pages 1125–1126, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

Planning and Evaluating Multiagent Influences Under Reward Uncertainty

(Extended Abstract)

Stefan Witwicki
GAIPS / INESC-ID
Instituto Superior Técnico
UTL (Porto Salvo, Portugal)
stefan.witwicki@ist.utl.pt

Inn-Tung Chen, Edmund Durfee, and Satinder Singh
Computer Science and Engineering
University of Michigan (Ann Arbor, MI, USA)
{inntung, durfee, baveja}@umich.edu

ABSTRACT

Forming commitments about abstract influences that agents can exert on one another has shown promise in improving the tractability of multiagent coordination under uncertainty. We now extend this approach to domains with meta-level reward-model uncertainty. Intuitively, an agent may actually improve collective performance by forming a weaker commitment that allows more latitude to adapt its policy as it refines its reward model. To account for reward uncertainty as such, we introduce and contrast three new techniques.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory, Performance

Keywords

Multiagent Planning, Transition-Decoupled POMDP, Model Uncertainty, Bayesian Rewards, Influence Abstraction, Commitments

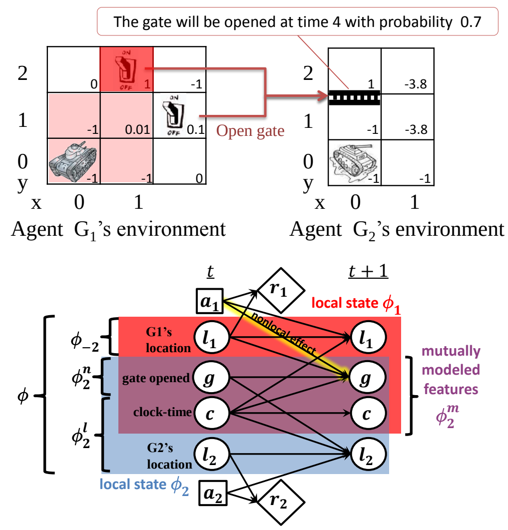


Figure 1: Example Problem and corresponding TD-POMDP

1. INTRODUCTION

Implicit in the problem of optimal multiagent coordination is the need to balance the local value of one's actions with the nonlocal value gained (or lost) from helping (or hindering) others. This problem is complicated by the presence of transition and observation uncertainty, where agents cannot be certain of the effects of their actions on their peers nor be fully aware of the situations their peers are encountering. Influence abstraction has proven useful in reducing the computational burden of optimal coordination by restricting consideration to an abstracted space of possible probabilistic nonlocal effects [1, 5]. In a running example shown in Figure 1 (top), two military field units G_1 and G_2 (where G_1 can use one of two switches to open a gate for G_2) can successfully coordinate by G_1 committing to a desirable *influence* in the form of a time and probability of opening the gate. By abstracting away local policy details that are superfluous to other agents, influences can enable agents to effectively cope with transition and observation uncertainty.

In this paper, we consider a third complicating factor: *dynamic and uncertain rewards*. In the example problem, the rewards G_1 receives in different locations depend upon the presence of an encroaching enemy; as time progresses, the enemy might render some locations more harmful, as manifested by nondeterministically decreasing rewards (with an intensity reflected in the shading in Figure 1). If the agent were alone, it could leverage the reward dynamics to reactively select the best actions depending on how its rewards progress. (For instance, it could navigate away from a switch as it starts to become more harmful.) Committing to a particular influence (e.g., raising the gate at a given time), on the other hand, may constrain the agent's policy in such a way as to preclude taking these actions and saving itself from harm. When planning its influences under reward uncertainty, the agent should account for the latitude that each influence allows in improving its local value. This insight motivates our investigation into the efficacy of influence-based planning under reward uncertainty, which we summarize below.

2. INFLUENCE-BASED PLANNING

There are several decision-theoretic formulations for problems like that portrayed in Figure 1, where agents act largely independently but can sometimes achieve preconditions that affect others [1, 4, 5]. Each of these formulations decomposes the conventional joint

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

decision model [2] into a set of $\mathcal{N}$ *local models*, one per agent i that includes a *local state* feature vector ϕ_i; similarly, they decompose the *joint reward* function into a summation of local reward functions: $R(\phi(t), a(t)) = \sum_{i=1}^{\mathcal{N}} R_i(\phi_i(t), a_i(t))$. The TD-POMDP of Witwicki and Durfee (W&D) [5], an instance of which is depicted in Figure 1 (bottom), further divides a local state ϕ_i into *nonlocally-affected* features ϕ_i^n (that only other agents' actions immediately affect) and *locally-affected* features ϕ_i^l, and explicitly distinguishes those *mutually-modeled* features ϕ_i^m through which i's interactions occur. In our example, agent G_2 models a single nonlocally-affected feature g (gate-opened) that depends on G_1's action.

As W&D have derived, an agent j can plan optimally using a local belief state, $\boldsymbol{b}_i(t) = \langle \phi_i(t), \phi_i^m(1..t-1) \rangle$), and thereby account for the influence of other agents by modeling a probability distribution over changes to its nonlocal features: $\Gamma_{\rightarrow j} = Pr(\phi_j^n(1..T))$. This distribution, which refer to as agent j's *incoming influence*, in our example encodes the probability that agent G_1 will open the gate for agent G_2 (at each time): $\Gamma_{\rightarrow G_2} = Pr(g(1..T))$. Specifying $\Gamma_{\rightarrow j}$ fully decouples agent j from all other agents, allowing j to compute and evaluate its local policy without having to consider the other agents' policies. Moreover, the optimal joint policy can be computed by searching a finite space of *joint influence* points, which W&D have shown can be significantly smaller than the joint policy space. In evaluating a given point in the *influence space*, agent j should also reason about its *outgoing influence* $\Gamma_{j\rightarrow}$, selecting an *influence-constrained* policy $\pi_j^{*|\Gamma_{j\rightarrow}}$ that achieves $\Gamma_{j\rightarrow}$.

3. THREE ALGORITHMS FOR HANDLING REWARD UNCERTAINTY

Given a fixed, known model of the agents' environment, outgoing influence achievement, incoming influence evaluation, and influence-based planning are all well defined [5]. We now extend them to dynamic or unknown environments wherein agents may be uncertain as to which model is the correct model. In particular, let there be K possible local reward functions $\{R_i^k\}_{k=1}^K$ per agent (independently distributed). Prior to execution, each agent i has only a prior distribution over its reward function, but during execution, i's observations can inform a posterior distribution over the true reward function. In the subsections that follow, we introduce and contrast three different influence-based planning extensions that afford different levels of computational efficiency and approximation.

Extended Belief State (EBS).

First, consider that W&D's approach can be directly applied to a TD-POMDP wherein each agent's belief state has been extended to include a distribution over the true reward function. By branching for every realizable posterior reward distribution after every action, the agent can account for the uncertainty precisely as it plans and evaluates influence points. However, the computation of each such evaluation will depend heavily on the size of the reward distribution, over which the extended-belief-state space grows exponentially.

Mean Reward (MR).

A simple approximation to the EBS algorithm is to completely collapse the uncertainty over each agents' rewards into a single expected or mean reward function, i.e., use the reward distribution to induce a mean-reward TD-POMDP where agent i's local reward function is $\bar{R}_i(\phi_i(t), a_i(t)) = \sum_{k=1}^K Pr(R_i^k) R_i^k(\phi_i(t), a_i(t))$. W&D's influence-based planning method then implicitly accounts for reward uncertainty at no additional computational cost. Although generally an approximation, we have proven that the mean-reward (MR) algorithm is optimal in special cases where agents cannot gain information (informing a new posterior distributions) about their true reward functions as they act and observe.

Influence-Constrained Iterative MR (ICIMR).

Finally, we develop a hybrid approach that builds off of the iterative mean-reward algorithm (IMR) for single-agent Bayesian-MDPs [3]. IMR reapplies the mean-reward technique after each belief update, because changes to the posterior distribution over reward functions can yield a different mean reward function $\bar{R}_i^{t+1} = \mathbb{E}_{R_i^k \sim \boldsymbol{b}_i(t+1)}[R_i^k]$, and hence adopting the policy $\tilde{\pi}_i^{t+1}$ optimal with respect to the updated mean reward may outperform the current policy $\tilde{\pi}_i^t$. Effectively, this involves (perhaps pre-)computation and adoption of a new policy at each time step.

Our ICIMR algorithm's novel departure from IMR comes from our multiagent setting and the role of commitments to influences. An agent who has already committed to probabilistically influencing others cannot iteratively shift from policy to policy without taking its committed outgoing influences into account. A stringent constraint that we could place on this agent is that its policy at the current iteration must *from its current state* satisfy all its outgoing commitments. Unfortunately, this is untenable, because stochastic state transitions could have put the agent into a state from which *no* policy can achieve the requisite commitments. Instead, we require that the agent's adopted policy must have satisfied its commitments, in expectation, *from its initial state*. Formally, agent i should adopt policy $\tilde{\pi}_i^{t+1} = \pi_i^{*|\boldsymbol{b}_i(t+1), \Gamma_{i\rightarrow}}$ that achieves outgoing influences $\Gamma_{i\rightarrow}$ *and* is consistent with its previous action choices $\tilde{\pi}_i^t(0..t)$.

With ICIMR, agent i plans and evaluates outgoing influences by iteratively considering each possible next mean-reward MDP that it could encounter. This resembles the lookahead performed with the EBS algorithm, except that whereas EBS considers every possible action at each successive state, ICIMR only considers the action dictated by the mean-reward policy in the state (given the posterior reward distribution). ICIMR branches for transition and reward uncertainty, but not future action, thereby allowing more efficient planning. And, although ICIMR is an approximation of EBS, we have proven that ICIMR yields solutions whose quality is greater than or equal to those of MR. A preliminary empirical analysis indicates that ICIMR can strike a good compromise between solution quality and computational overhead, making it a useful technique for tackling reward uncertainty in an efficient, yet principled, manner.

4. ACKNOWLEDGEMENTS

This work was supported, in part, by the Fundação para a Ciência e a Tecnologia (FCT) and the CMU-Portugal Program under project CMU-PT/SIA/0023/2009, and by the National Science Foundation under grants IIS-0964512, IIS-1148668, and IIS-0905146.

5. REFERENCES

[1] R. Becker, S. Zilberstein, and V. Lesser. Decentralized Markov decision processes with event-driven interactions. In *AAMAS*, pages 302–309, 2004.

[2] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Oper. Res.*, 27(4):819–840, 2002.

[3] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of ICML '06*, pages 697–704, 2006.

[4] P. Varakantham, J. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *ICAPS,* 313–320, 2009.

[5] S. Witwicki and E. Durfee. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *ICAPS,* 185–192, 2010.

A Better Maximization Procedure For Online Distributed Constraint Optimization

(Extended Abstract)

Yoonheui Kim
University of Massachusetts at Amherst, MA 01003, USA
ykim@cs.umass.edu

Victor Lesser
University of Massachusetts at Amherst, MA 01003, USA
lesser@cs.umass.edu

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Coherence and coordination*

General Terms

Performance, Algorithms

Keywords

Distributed Problem Solving, DCOP, Max-Sum

1. INTRODUCTION

Many message passing algorithms on graphical models include maximization operations on sums of local node function and message values from neighbors. In recent work by McAuly et al, faster maximization computation was achieved in a static environment by offline presorting of the values of local functions. However, this efficiency is only guaranteed in special cases when constraint nodes receive messages involving fewer variables than the local function. In this paper, we generalize the approach to be applicable to more general settings where offline presorting of constraint functions is not realistic and messages may involve as many variables as the constraint function. We further improve the approach in two ways, first by creating different value sets with sum values from the previous cycle and the changes in message values from the current cycle, and second by conditionally applying the technique based on a correlation measure. These new approaches with no preprocesing step obtain the expected computational complexity with an exponent of 1.5 of the possible values per node except the initial cycle which requires 2. We demonstrate the effectiveness of this approach in a distributed optimization problem involving the coordination and scheduling of radars.

2. FAST BELIEF PROPAGATION

The Fast Belief Propagation (FBP) [4] scheme was developed for a (undirected) graphical model, where the maximum a posteriori inference is done by finding the values of variables that maximize the sum of the node and edge potentials. In a pairwise graphical model, computing a message $m_{A\to B}$ between two neighboring cliques $A = (\mathbf{i}, \mathbf{j})$ and $B = (\mathbf{i}, \mathbf{k})$ is equivalent in complexity to solving

$$m_{A\to B}(y_i) = \Psi_i(y_i) + \max_{y_j}[\underbrace{\Psi_j(y_j)}_{\mathbf{v_a}} + \underbrace{\Phi_{i,j}(y_i, y_j)}_{\mathbf{v_b}}], \qquad (1)$$

where $\Psi_i(y_i)$ is the sum of $\Phi(y_i|x_i)$ and any first-order messages over y_i, that is, the sum of the values only related to y_i given the observation x_i (similarly for $\Psi_j(y_j)$).

Let the list of values of v_a be L_A and that of v_b be L_B. Assuming the list L_B, which is of length N, contains the values of v_b already sorted offline for each value of y_i, and assuming the length of the list of L_A is much smaller than N where sorting of L_A is much smaller than $O(N)$, we can use the FBP technique to find the maximum sum of L_A and L_B with expected time complexity $O(\sqrt{N})$ given binary constraints and order statistics independence of variables.

3. FAST BELIEF PROPAGATION ON GENERAL GRAPHS

We relax the assumption that the constraint function is given offline by partially sorting the value list online which contains constraint function values for every variables' configuration. Also, the guarantee of FBP on performing in $O(\sqrt{N})$ is restricted to pairwise factor graphs using two lists and we maintain the benefit of $O(\sqrt{N})$ on n-ary factors by introducing the message list that we construct by merging incoming messages. Additionally we relax the assumption that there are fewer variables associated with messages than the number of variables in a constraint function by partially generating the message list, i.e. the same number of sorted items in value lists.

G-FBP: Algorithm for applying FBP with partial lists

We have modified the FBP technique, which we call G-FBP, so that it can be applied to partially sorted lists where the ranks of some items in unsorted part are not known. Once the bounding items, which has higher ranks than the item with maximum value are found, unmatched items are computed directly from constraint function and received messages. Let the number of sorted items $K\sqrt{N}$ with list of length N,

THEOREM 1. *The expected time complexity of $O(\sqrt{N})$ holds with partial lists when $(1 - \frac{K}{\sqrt{N}})^{K\sqrt{N}} < \frac{K}{\sqrt{N}}$.*

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Message-Passing with G-FBP technique

The technique works with any message-passing based optimization algorithm.

1. [Value List Construction] In cycle 1,
   1.1 work as in the original message passing algorithm which computes all possible variable nodes' value configuration $\mathbf{x}$ and store the values $F(\mathbf{x})$.
   1.2 Send messages for all x_i of neighbor i.
   1.3 For the lists created in step 1.1, select the top $K\sqrt{N}$ values from each list and order the items and save this as $v_b(x_i)$.
2. [Message List Construction] Sort the received messages from each neighbor. Combine the $K\sqrt{N}$ items by adding the sorted messages from the largest items.
3. [Finding Maximum] Find maximum using value list v_b and message list v_a for all x_i for all neighbors i using G-FBP technique
4. send the messages using the computed maximum value.
5. repeat step 2-4.

4. INDEPENDENCE ASSUMPTION AND CORRELATION MEASURE

With two negatively correlated lists, it is likely that the G-FBP scheme fails to find the maximum item within limited number of items therefore increasing the time complexity of the algorithm. The correlation of two lists are domain-dependent [4] and, from our observations, it also varies for each constraint function and received messages on each cycle. We extend G-FBP technique to ensure the independence of two lists.

Correlation Measure

We modify the Spearman's rank correlation measure [5] to measure correlation among two partially sorted lists and conditionally apply G-FBP technique. Let x and y be two sorted lists where x_i and y_i are the ranks of the items with index i. m is the median rank. We use correlation measure:

$$\rho' = \frac{\sum_i (r_{x_i})(r_{y_i})}{\sqrt{\sum_i r_{x_i}^2 \sum_i r_{y_i}^2}} \tag{2}$$

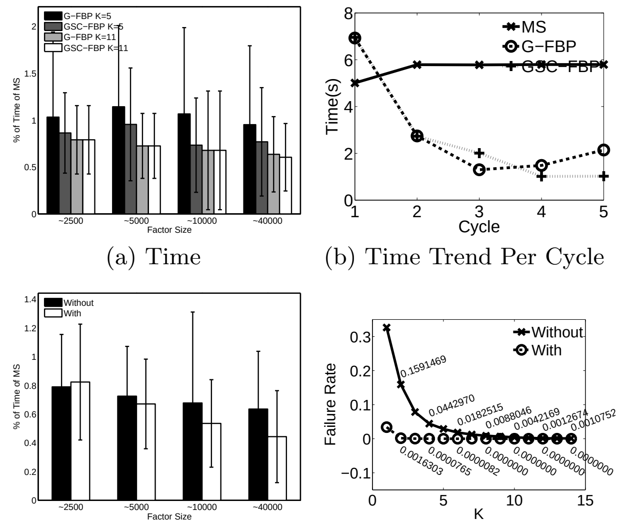
$$\text{where } r_k = \begin{cases} \frac{(N-K\sqrt{N})}{K\sqrt{N}}(k_i - m_k), & \text{if } k_i \text{ in sorted lists} \\ (\frac{(K\sqrt{N}+1+2K\sqrt{N})}{2} - m_k), & \text{if } k_i \text{ is not found} \end{cases}$$

and $m_k = K\sqrt{N} + 1/2$, imaginary median, as we consider the imaginary length of list $2K\sqrt{N}$.

GSC-FBP: Improving the Rank of Items in the list

We create two lists, sum list L_{sum} , the value of sum for variables' configuration in the previous round, and change list $L_{changes}$, the difference between previous and current round messages as the following equation 3 and replace value list and message list. As the algorithm proceeds, the list L_{change} becomes closer to the uniform distribution which makes it independent of L_{sum}. Let $r_{m\to n}$ be the message from function node m to variable node n.

$$\begin{aligned} & r_{m\to n}(x_n) \\ & = \max_{x_m \backslash n}(\Sigma_{n' \in N(m) \backslash n}(\underbrace{q_{n'\to m}(x_{n'})}_{\text{data dependent}} + \underbrace{f_m(x_m)}_{\text{data independent}})) \\ & = \max_{x_m \backslash n}(\Sigma_{n' \in N(m) \backslash n}(\underbrace{q'_{n'\to m}(x_{n'}) - q_{n'\to m}(x_{n'})}_{changes} \\ & + \underbrace{f_m^{previoussum}(x_m) + q_{n'\to m}(x_{n'})}_{sum}))) \end{aligned} \tag{3}$$



(a) Time (b) Time Trend Per Cycle

(c) Time (d) Failing Probability

Figure 1: (a) Computation Time Ratio against Max-Sum. (d) The Probability of Failing to Find the Maximum Item in Sorted Parts. With: G-FBP using correlation. Without: G-FBP not using correlation.

Round	4	5	6	7
G-FBP	0.0793	0.2612	0.2607	0.3543
GSC-FBP	0.0500	0.0450	0.0730	0.0208

Table 1: Relative position of Bounding Items

5. EXPERIMENTS

We compared the performance with the Max-Sum approximate distributed constraint algorithm [1] on the domain of real-time adaptive NetRad system [3]. See [2], for more details on the formulation of the distributed constraint optimization problem for this problem. We use an abstract simulator that involved 48 radars with a scenario of 96 phenomena with random locations, size, and type(density 2 constraint graphs).

6. REFERENCES

[1] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 639–646, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[2] Y. Kim, M. Krainin, and V. Lesser. Effective Variants of the Max-Sum Algorithm for Radar Coordination and Scheduling. In *Proceedings of 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 357–364, Lyon, France, October 2011.

[3] J. F. Kurose et al. An End-User-Responsive Sensor Network Architecture for Hazardous Weather Detection, Prediction and Response. In *Proceedings of the Second Asian Internet Engineering Conference, AINTEC*, pages 1–15, 2006.

[4] J. J. McAuley and T. S. Caetano. Exploiting data-independence for fast belief-propagation. In J. Fürnkranz and T. Joachims, editors, *ICML*, pages 767–774. Omnipress, 2010.

[5] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3/4):441–471, 1987.

Agent-human Coordination with Communication Costs under Uncertainty*

(Extended Abstract)

Asaf Frieder[1], Raz Lin[1] and Sarit Kraus[1,2]
[1] Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel 52900
{linraz,sarit}@cs.biu.ac.il
[2] Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742 USA

ABSTRACT

As agents' technology becomes increasing more prevalent, coordination in mixed agent-human environments becomes a key issue. Agent-human coordination is becoming even more important in real life situations, where uncertainty and incomplete information exists and communication is costly. While abundant research has focused on aspects of computerized teamwork, little attention has been given to the issues raised in teams that consist of both computerized agents and people. In this paper we focus on teamwork between an agent and a human counterpart and present a novel agent designed to interact proficiently with people. In extensive simulations we matched our agent with people and compared it with another state-of-the-art agent. Our results demonstrate the significant improvement in coordination when our agent is involved.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Experimentation

Keywords

human-robot/agent interaction, POMDPS, uncertainty, teamwork

1. INTRODUCTION

More and more agents are deployed in mixed agent-human environments and are expected to interact efficiently with people. Such settings may include uncertainty and incomplete information. Communication, which can be costly, might be available for the parties to assist in obtaining more information in order to build a good model of the world. Efficient coordination in teams between agents and people is the key component for turning their interaction into a successful one. The importance of coordination between agents and people only increases in real life situations, in which uncertainty and incomplete information exist.

Teamwork has been the focus of abundant research in the multi-agent community. However, while research has focused on decision theoretic framework, communication strategies and multi-agent policies (e.g., [2]), only some focus has been on the issues raised when people are involved as part of the team [3]. Our work focuses on efficient coordination between agents and people with communication costs and uncertainty. We model the problem using DEC-POMDPs (Decentralized Partially Observable Markov Decision Process) [1]. The problem involves a team having a joint reward (goals), while each team member has only partial observations of the state of the world. Thus, even if information exists, it only provides partial support as to the state of the world, making it difficult to construct a reliable view of the world without coordinating with other teammates. To validate the efficacy of our agent, we chose the Colorado/Wyoming domain, which was first introduced by Roth *et al.* [2] and offered as a benchmark for evaluation of communication heuristics in multi-agent POMDPs.

While there are studies that focus on DEC-POMDPs, most of them pursue the theoretical aspects of the multi-agents aspect, and do not deal with the fact that people can be part of the team [2]. Zuckerman *et al.* [4] improved coordination with humans using focal points. We, however, focus on the problem of improving coordination between an agent and people by means of shared observations. The addition of communication only increases the challenge, making the adaptation of their model far from straightforward. Our novelty also lies in introducing an agent capable of successfully interacting with a human counterpart in such settings. The agent is adaptable to the environment and people's behavior, and is able to sophisticatedly decide which information to communicate to the other team member based on the communication cost and the possible effects of this information on its counterpart's behavior.

2. COORDINATION WITH COMMUNICATION COSTS IN DEC-POMDPS

A DEC-POMDP [1] model separates the resolution of the problem into time steps in which the agents choose actions simultaneously. These actions can have deterministic or non-deterministic effects on the state. Following these actions, each team member receives an additional observation of the world state. The state transition and the joint reward function are dependent on the joint actions of all agents.

*This research is based upon work supported in part by the U.S. Army Research Lab and Research Office grant number W911NF-08-1-0144, ARO grants W911NF-09-1-0206 and W911-NF-11-1-0344 and by ERC grant #267523.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

We focus on POMDPs in which the team consists of two agents and the team members are able to communicate with each other (e.g., [2]). As communication is costly we limit the communication messages to include only self observations. This can also be supported in real settings where limitations occur to prevent lengthy communications that can breach the integrity of the team members (e.g., surrendering their locations). By sharing their observations, the team members can avoid uncoordinated actions caused by contradictory private knowledge, allowing them to build a coherent and a concise view of the world states faster.

A naïve approach for team communication is sharing all information among themselves. Once all the information is shared, finding the optimal joint action becomes a simple POMDP problem that each team member can solve in parallel. Then, each member can perform the action assigned to it in the joint action plan, described by the POMDP policy, for their joint belief. However this solution is only optimal if two assumptions hold. First, that there is no cost associated with communication. Second, that all team members consider the same joint actions to be optimal (by using the same POMDP policy). As this is hardly the case in real settings, existing agents might fail when matched with people. Our agent's design takes these considerations into account to achieve proficient interaction with people.

3. AGENT DESIGN

When coordinating with someone else, it is hard to predict with full certainty what the other team member (especially if it is a human partner) will do. The task is even harder if the agent interacts with someone only once and not repeatedly. Thus, an efficient agent working with people needs, amongst other things, to approximate what percentage of the population will perform each action based on the existing partial observations. Our agent interacts with the same counterpart only once and thus its design tries to tackle the challenge by generating a good model of the population based on 445 people who played the game. Our agent uses a neural network which outputs the probabilities of the other team member taking an action based on features that encod the agent beliefs, past actions and communication- and position-related information. We coin our agent *TMDC* (standing for team modeling with decentralized communication).

4. DESIGNING THE AGENT'S STRATEGY

The general design of the agent's strategy consists of building a POMDP using the prediction of the human behavior described beforehand. This is done as when interacting with people we cannot ensure mutual predictability. Thus, *TMDC* uses its model, and not the shared belief, to predict what will be its counterpart's behavior. In addition, *TMDC* chooses its action based on all its knowledge (which also includes private knowledge), and only communicates in order to influence the actions of the other teammate. Given all previously shared observations, the agent evaluates an action by considering all possible results, calculating immediate rewards and using offline estimation of future rewards. This evaluation is then used by a hill climbing heuristic that finds which observations (taken from the set of all observations, including shared observations) can maximize the score of the team and hence should be shared. We continue to describe the agent's strategy in detail.

5. EXPERIMENTS

The experiments were conducted on the Colorado/Wyoming domain and were conducted using the Amazon Mechanical Turk service (AMT). This framework allows publishing of tasks designated for people all around the world. We prohibited multiple participation by the same people. The players were provided with a manual of the game before their participation. Although the manual is very detailed, we took great care not to give strategic advice. We then required that each worker pass a short multiple choice test to verify that they read the manual and understood the game. The player received a bonus based on the score of the team, if it was positive. We ensured that the costs and penalties of the game would have a meaningful effect on the player even if the team did not gain the reward for a successful signal.

We matched 64 people with our *TMDC* agent, with a state-of-the-art agent *PDCS* [2] and with 64 other people (*PDCS* was designed to coordinate well with multi-agent teams). The results demonstrate that our agent significantly outperforms the *PDCS* agent ($p < 0.001$) when matched with people (52.84 as compared to 17.5). Interesting also that the human-human team achieved a score of only 27.18. While this score generated no significance difference compared to the *PDCS* results, the *TMDC*-human teams achieved significantly higher scores ($p < 0.003$) than the human-human teams.

6. CONCLUSIONS

Settings in which hybrid teams of people and automated agents need to achieve a common goal are becoming more common in today's reality. Communication in such situations is a key issue for coordinating actions. As communications is costly and sometimes even limited (e.g., due to security issues or range limitations) it becomes of great essence to devise an efficient strategy to utilize communication. This paper presented a novel agent design that can proficiently coordinate with people under uncertainty while taking into account the cost of communication.

Our agent was specifically designed taking into account the fact that it interacts with people and was also evaluated with people. Experiments with more than 300 people demonstrated how it outperforms the state-of-the-art agent. One of the main factors accounting for the success of our agent is the understanding that it requires a good model of the counterpart to generate an efficient strategy.

This paper is only part of a new and exciting journey. Future work warrants careful investigation on improving the prediction model of people's behavior. In addition we will investigate settings in which even more limited information is available to the team members. In such situations the challenge is on the understanding of the abstract model that is available and how to utilize communication's strategies for efficient coordination that will allow increasing the accuracy of the model.

7. REFERENCES

[1] D. 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.

[2] M. Roth, R. Simmons, and M. Veloso. What to communicate? execution-time decision in multi-agent POMDPs. *Distributed Autonomous Robotic Systems*, 2006.

[3] A. van Wissen, Y. Gal, B. Kamphorst, and V. Dignum. Human–agent teamwork in dynamic environments. *Computers in Human Behavior*, 28:23–33, 2012.

[4] I. Zuckerman, S. Kraus, and J. S. Rosenschein. Using focal points learning to improve human-machine tactic coordination. *JAAMAS*, 22(2):289–316, 2011.

Token Economy for Online Exchange Systems

(Extended Abstract)

Jie Xu
University of California
Los Angeles
Electrical Engineering
Department
jiexu@ucla.edu

William Zame
University of California
Los Angeles
Economics Department
zame@econ.ucla.edu

Mihaela van der Schaar
University of California
Los Angeles
Electrical Engineering
Department
mihaela@ee.ucla.edu

ABSTRACT

This paper studies the design of online exchange systems that are operated based on the exchange of tokens, a simple internal currency which provides indirect reciprocity among agents for cooperation. The emphasis is on how the protocol designer should choose a protocol - a supply of tokens and suggested strategies - to maximize service provision, taking into account that impatient agents will comply with the protocol if and only if it is in their interests to do so. The protocol is designed in such a way that it is robust to (small) errors in the designer's knowledge of the system parameters. We prove that robust protocols have a simple pure threshold structure and there is a unique optimal supply of tokens that balances the token distribution in the population and achieves the optimal efficiency. In the meanwhile, we also emphasize that choosing the wrong token supply can result in an enormous efficiency loss.[1]

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*intelligent agents, multiagent system, cooperation*

General Terms

Design, Economics, Theory

Keywords

Game theory, Agent cooperation, Formal model

1. INTRODUCTION

Content, knowledge and resource sharing services are currently proliferating in many online systems, e.g. BitTorrent, Yahoo Answers and crowdsourcing markets such as Amazon Mechanical Turk. The expansion of such sharing and exchange services will depend on their participating members (herein referred to as agents) to contribute and share resources with each other. However, these systems are vulnerable to "free-riding" problems since the participating agents are self-interested. To compel the self-interested agents to cooperate, incentive schemes can be designed which rely on the information that individual agents have. Such incentives schemes can be classified into two categories: personal reciprocation (direct reciprocation) [1] and social reciprocation (indirect reciprocation)[4]. Reputation is often used as a way to achieve cooperation among self-interested users but has limitations in fully anonymous and decentralized systems. Moreover, they are also vulnerable to collusion attacks.

In this paper, we design a new framework for providing incentives in social communities, using tokens. Agents exchange tokens for services: the client who receives service from a server pays for that service with a token which the server will later use to obtain service when it becomes a client. Here we ask what the designer can achieve by imposing a system that relies solely on the exchange of intrinsically worthless *tokens* or *fiat money*. Our emphasis is on the *design* of such a system; in particular, how the designer should choose a protocol - a supply of tokens and suggested strategies - to maximize the system efficiency. Because it seems impossible for the designer to have exact knowledge of the system parameters, we insist that the chosen protocol must be consistent with (small) perturbations in these parameters. Thus, the chosen protocol must induce a *robust equilibrium.* Among all such choices/recommendations, the designer should select one that maximizes the social welfare/system efficiency - or at least approaches this maximum. We characterize the robust equilibria (in terms of the system parameters), show that they have a particularly simple form, and determine the achievable system efficiency. When agents are patient, it is possible to design robust equilibria to nearly optimal efficiency; however, the correct design is important: the "wrong" design do not achieve nearly the optimum, even when agents are arbitrarily patient.

This work connects to a number of literatures. The most related ones include macro-economic literature on money as a medium of exchange [3][6] and computer science and engineering literature on token-like system design [2][5].

2. PROBLEM FORMULATION

We consider a continuum (mass 1) of agents each possess a unique resource that can be duplicated and provided to others. The benefit of receiving this resource is b and the cost of producing it is c; we assume $b > c > 0$. Agents discount future benefits/costs at the constant rate $\beta \in (0, 1)$. Agents are risk neutral so seek to maximize the discounted present value of a stream of benefits and costs. Time is discrete. In

[1] Full version of this paper can be found online at http://www.ee.ucla.edu/~jiexu/documents/aamas12_long.pdf

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

each time period, a fraction $\rho \leq 1/2$ of the population is randomly chosen to be a client and matched with randomly chosen server; the fraction $1-2\rho$ is unmatched. When a client and a server are matched, the client chooses whether or not to request service, the server chooses whether or not provide service if requested. The parameters b, c, β, ρ completely describe the environment. Write the benefit/cost ratio $r = b/c$. Each agent can hold an arbitrary non-negative finite number of tokens, but cannot hold a negative number of tokens and cannot borrow. The protocol designer creates incentives for the agents to provide or share resources by providing a supply of tokens and recommending strategies for agents when they are clients and servers. The recommended strategy is a pair $(\sigma, \tau) : \mathbb{N}_+ \to (0,1)$; τ is the client strategy and σ is the server strategy. For each token holding k, $\sigma(k)$ is the recommended probability to provide service when the agent becomes a server; $\tau(k)$ is the recommended probability to request service when it is a client. In other words, the protocol designer recommends a mixed strategy for the agents.

The system designer chooses a protocol $\Pi = (\alpha, \sigma, \tau)$ where α is the supply of tokens (average number of tokens per capita). Define the system efficiency as the probability that the service provision is successfully carried out when two agents are paired given the system parameters b, c, β. Write μ the fraction of agents who do not request service when they are clients and ν the fraction of agents who do not provide service when they are servers. by the Law of Large Numbers, the efficiency is computed in the straightforward manner,$\texttt{Eff}\,(\Pi|b,c,\beta) = (1-\mu)(1-\nu)$. Taking into account that impatient agents will comply with the protocol if and only if it is in their interests to do so, the protocol need be an equilibrium given the system parameters. Write $\Phi(\Pi)$ the set of $\{(\beta,\gamma)\}$ for which Π is an equilibrium. The design problem are thus to choose the protocol

$$\Pi = \underset{\Pi:(\beta,r)\in\Phi(\Pi)}{\arg\max}\ \texttt{Eff}\,(\Pi|\beta, r)$$

3. MAIN RESULTS

3.1 Structural Property

The knowledge of the protocol designer of the system parameters (b, c, β) may not be accurate. Hence, the strategy must be robust to the small perturbations in the parameters.

THEOREM 1. *If $\Pi = (\alpha, \sigma)$ is a robust equilibrium then σ is a pure threshold strategy.*

Existence of equilibrium is not trivial. It is not obvious that there will be any discount factor β that makes agents be willing to use a certain threshold. The following theorem claims that such β can always be found.

THEOREM 2. *For each pure threshold strategy protocol $\Pi = (\alpha, \sigma_K)$ and benefit/cost ratio $r > 1$, the set $\beta : \Pi_K \in EQ(r, \beta)$ is a non-degenerate interval $[\beta^L, \beta^H]$.*

3.2 Optimal Token Supply

In general it seems hard to determine the efficiency of a given protocol or to compare the efficiency of different protocols. However, for a given threshold strategy, we can find the most efficient protocol and compute its efficiency. Write $\Pi_K = (K/2, \sigma_K)$.

THEOREM 3. *For a given threshold strategy σ_K , Π_K is the most efficient protocol; i.e., $\mathtt{Eff}(\alpha, \sigma_K) \leq \mathtt{Eff}(\Pi_K)$ for every per capita supply of tokens α. Moreover,*

$$\mathtt{Eff}\,(\Pi_K) = 1 - \frac{1}{(K+1)^2}$$

Theorem 3 identifies a sense in which there is an optimal quantity of tokens. This optimal token supply balances the token distribution in the population in the sense that there are not too many agents who do not serve or too many agent who cannot request service. However, these most efficient protocols (for a given threshold) need not be equilibrium protocols; i.e. such combinations of token supply and threshold need not be feasible for all system parameters. For example, given the benefit/cost ratio r, it does not exclude the possibility that for some discount factor β, we cannot find any threshold protocol with the corresponding optimal token supply that is an equilibrium. However, we disclaim this conjecture by showing that the sustainable discount factor intervals overlap between consecutive threshold protocols with optimal token supply. Based on this overlap property, the following theorem describes the equilibrium threshold in the limiting case.

THEOREM 4. *For each fixed benefit-cost ratio $r > 1$*

$$\lim_{\beta\to 1} \inf \{K : (\beta, r) \in \Phi(\Pi_K)\} = \infty$$

Characterizing the equilibrium threshold is important because only with the correct knowledge of sustainable thresholds can the protocol designer choose the right token supply. Otherwise, there may be an enormous efficiency loss. We provide an bound to make the point that choosing the wrong protocol can result in strict efficiency loss.

THEOREM 5. *For each $\alpha \in (0, \infty)$ and each threshold K*

$$\mathtt{Eff}\,(\alpha, \sigma_K) \leq 1 - \frac{1}{2\lceil\alpha\rceil + 1}$$

(independently of the parameters of the population)

4. REFERENCES

[1] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *the 5th ACM conference on Electronic commerce*, EC '04, pages 102–111, New York, NY, USA, 2004. ACM.

[2] E. J. Friedman, J. Y. Halpern, and I. Kash. Efficiency and nash equilibria in a scrip system for p2p networks. In *7th ACM conference on Electronic commerce*, EC '06, pages 140–149, New York, NY, USA, 2006. ACM.

[3] E. J. Green and R. Zhou. A rudimentary random-matching model with divisible money and prices. GE, Growth, Math methods 9606001, EconWPA, June 1996.

[4] M. Kandori. Social norms and community enforcement. *Review of Economic Studies*, 59(1):63–80, January 1992.

[5] I. A. Kash, E. J. Friedman, and J. Y. Halpern. Optimizing scrip systems: efficiency, crashes, hoarders, and altruists. In *Proceedings of the 8th ACM conference on Electronic commerce*, EC '07, pages 305–315, New York, NY, USA, 2007. ACM.

[6] R. Zhou. Individual and aggregate real balances in a random-matching model. *International Economic Review*, 40(4):pp. 1009–1038, 1999.

Using the Max-Sum Algorithm for Supply Chain Formation in Dynamic Multi-Unit Environments

(Extended Abstract)

Michael Winsper and Maria Chli
Aston University
Aston Triangle
Birmingham, United Kingdom B4 7ET
{winsperm, m.chli}@aston.ac.uk

ABSTRACT

The max-sum loopy belief propagation (LBP) algorithm was shown in [4] to produce strong results in a simple decentralised supply chain formation (SCF) scenario where goods are traded in single units. In this paper, we demonstrate the performance of LBP in a multi-unit SCF scenario with additional constraints. We also provide experimental analysis of LBP's performance in dynamic scenarios where the properties and composition of participants are altered while the algorithm is running. Our results suggest that LBP continues to produce strong solutions in multi-unit scenarios, and that performance remains solid in a dynamic setting.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Economics, Algorithms, Experimentation

Keywords

Supply Chain Formation, Loopy Belief Propagation

1. INTRODUCTION

Computational approaches to SCF model potential supply chain participants as individual computational agents which express their capabilities and costs through a mechanism which determines the subset of agents capable of forming the most efficient supply chain. Although centralised SCF techniques [1] have allowed for multi-unit exchanges for some time, the existing state of the art in decentralised [4, 3] SCF only model simple scenarios where goods are exchanged in single units. Additionally, [4] does not model the effect of changes to the properties or composition of participants once the process has begun. In this paper, we propose a framework for the representation of multi-unit supply chains and extend the LBP-based technique for decentralised SCF presented in [4] to the multi-unit case. We also present a set of experiments analysing the performance of LBP in a dynamic environment, where changes to the properties and composition of participants occur while the algorithm is running. Our results demonstrate that LBP is capable of producing efficient allocations over a range of network topologies in both static and dynamic environments.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. MODEL

The use of task dependency networks (TDNs) for the representation of supply chains was originally proposed in [3]. For the first time, we extend this TDN representation to the multi-unit case by modelling input to output good ratios, production capacities and consumer desired good quantities. An example of the extended representation is shown in Figure 1. Values below producers and consumers represent reserve prices and production capacities, and consumption values and desired consumable good quantities. Edges from goods to producers are labelled with the producer's input to output ratio for that good. A producer with a single input and an input ratio of 2 for that good requires two units of that good in order to produce one unit of its output.

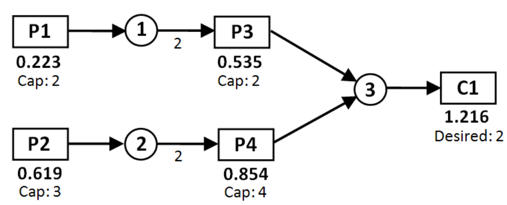


Figure 1: The Simple supply chain TDN from [3] extended to the multi-unit case. Producers (P[x]) and a consumer (C1) are represented by rectangles, while goods are represented by circles. Edges indicate potential flows of goods.

Producers and Consumers At initialisation, each producer is assigned a production capacity which specifies the maximum number of units each producer is able to produce of its output good, and an input to output ratio for each of their inputs. In order to produce one unit of their output good, producers are required to acquire a number of units of each of their input goods equal to their ratio for that good. A producer cannot produce its output good unless it acquires the necessary quantities of all of its input goods. Producers attach a reserve price R_p to their output good, which is linear with the number of units of its output good that it produces. Consumers seek to acquire a number of units of their consumable good no greater than their desired consumable good quantity. In each network, each consumer is assigned a static consumption value V_c representing the valuation the consumer holds for obtaining a single unit of its consumable good. The total value a consumer receives is linear with the number of goods it obtains.

3. APPLYING THE MAX-SUM ALGORITHM

The max-sum algorithm is a variant of loopy belief propagation (LBP), a decentralised and distributed approximate inference scheme involving applying Pearl's belief propagation algorithm [2] to graphs containing cycles. It uses iterative stages of message passing as a means for estimating the maximum a posteriori assignment; in our case, this corresponds to the network-wide state configuration that maximises Eq. 3. Each state encodes a combination of purchases and sales which may be made by an agent. States are associated with costs - the *unary cost*, representing the cost to the allocation of the agent being in that state, and the *pairwise cost*, which encodes the compatibility of two states of neighbouring agents. At each iteration of the algorithm, every node in the graph sends a message to each of its neighbours, representing the sender's beliefs about the potential cost to the total efficiency of the network of each of the recipient's states. This is calculated using Eq. 1, where x_v is a state of recipient j, $bel_i(x_u)$ is sender i's belief in its own state x_u, $m_{j\to i}(x_u)$ is the message passed from j to i in the previous step about i's state x_u, and $g_{ij}(x_u, x_v)$ is the pairwise cost of states x_u and x_v. Once values have been calculated for all of j's states, i passes the message to j.

$$m_{i\to j}(x_v) = min_{x_u}\Big(bel_i(x_u) - m_{j\to i}(x_u) + g_{ij}(x_u, x_v)\Big) \quad (1)$$

Once all nodes have sent a message to each of their neighbours, nodes then update their beliefs about their own states based upon the content of the messages they received using Eq. 2, where $f_i(x_u)$ is the unary cost of i's state x_u, and $m_{j\to i}(x_u)$ are the messages received from i's set of neighbours N_u about state x_u in the previous step. The process of message passing and belief update continues until the beliefs of each node stabilise. For more information on applying LBP to SCF, we refer the reader to [4].

$$bel_i(x_u) = f_i(x_u) + \sum_{j\in N_u} m_{j\to i}(x_u) \quad (2)$$

Allocation Before allocation is performed, each agent determines their *final state* - the state, when beliefs stabilise, which the agent believes holds the lowest cost. Once the final states of each of the agents have been determined, we classify producers which successfully sell their output good and consumers which acquire their consumable good as *active*. We calculate allocation values using Eq. 3, where C is the set of active consumers C, A_c is the number of goods acquired by consumer c, P is the set of active producers and M_p is the number of goods manufactured by producer p.

$$Val = \sum_{c\in C} V_c A_c - \sum_{p\in P} R_p M_p \quad (3)$$

4. RESULTS

We perform two sets of experiments, examining the performance of LBP in both static and dynamic environments. In the static environment, we compare LBP with a multi-unit implementation of the SAMP-SB auction protocol from [3], extended by using multiple copies of each agent to represent capacities and desired good quantities. This representation does not allow for the use of input to output good ratios, so for fair comparison we also test LBP with all ratios set to 1, referred to as ratioless LBP. We test each technique over 100 runs on each of the networks from [3]. We vary input ratios (drawn from $[1\ldots2]$), consumer desired goods (from $[2\ldots3]$), reserve prices (from $U(0,1)$), and production capacities ($[4\ldots5]$) between each run, discarding runs in which the optimal allocation value, determined using mixed integer programming, is non-positive. Consumption values are fixed at the per-network values given in [3] over every run. In the dynamic environment, at a number (drawn from $[6\ldots10]$) of randomly chosen steps during each run we randomly change one of the aforementioned properties of a single agent, introduce a new producer or consumer, or remove a producer. We present our results in terms of average efficiency, calculated by dividing the total allocation values produced by each method over 100 runs on each network by the maximum available value over the same 100 runs.

Table 1: Average efficiency in each network produced by LBP and the SAMP-SB auction protocol from [3] on the networks from [3]. A result of 1.000 equates to the capture of 100% of available efficiency.

Network	Static LBP *ratioless*	Static LBP *with ratios*	Dynamic LBP *with ratios*	Static SAMP-SB *ratioless*
Simple	1.000	1.000	0.911	0.999
Unbalanced	0.962	0.872	0.713	0.964
Two-Cons	0.986	0.983	0.801	0.963
Bigger	0.969	0.813	0.520	0.995
Many-Cons	1.000	1.000	0.989	0.425
Greedy-Bad	0.91	0.839	0.793	0.666

Table 1 shows the average efficiency produced by each method. We see that both static LBP-based methods are able to match or outperform SAMP-SB on the majority of networks. As expected, LBP finds the optimal allocation 100% of the time in static scenarios on acyclic networks, while still being able to produce highly efficient allocations on more loopy networks. We also see that LBP tended to perform better when input to output good ratios are not present; this is to be expected since the presence of ratios serves to constrain the number of solutions available. In the dynamic setting, we see that for most networks, average efficiency is roughly comparable to the results produced in a static environment.

5. REFERENCES

[1] J. Cerquides, U. Endriss, A. Giovannucci, and J. Rodríguez-Aguilar. Bidding Languages and Winner Determination for Mixed Multi-Unit Combinatorial Auctions. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 469–476, 2007.

[2] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, volume 1. Morgan Kaufmann, 1st edition, 1988.

[3] W. Walsh and M. Wellman. Decentralized Supply Chain Formation: A Market Protocol and Competitive Equilibrium Analysis. *Journal of Artificial Intelligence Research*, 19:513–567, 2003.

[4] M. Winsper and M. Chli. Decentralized Supply Chain Formation using Max-Sum Loopy Belief Propagation. *Computational Intelligence*, In Press, 2012.

Complexity and Approximability of Social Welfare Optimization in Multiagent Resource Allocation

(Extended Abstract)

Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe
Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany
Nhan-Tam.Nguyen@uni-duesseldorf.de, {thanh, roos, rothe}@cs.uni-duesseldorf.de

ABSTRACT

An important task in multiagent resource allocation, which provides mechanisms to allocate bundles of (indivisible and nonshareable) resources to agents, is to maximize social welfare. We study the computational complexity of exact social welfare optimization by the Nash product, which can be seen as a sensible compromise between the well-known notions of utilitarian and egalitarian social welfare. When utilitiy functions are represented in the bundle or the k-additive form, for $k \geq 3$, we prove that the corresponding computational problems are DP-complete (where DP denotes the second level of the boolean hierarchy over NP), thus confirming two conjectures raised by Roos and Rothe [10]. We also study the approximability of social welfare optimization problems.

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

Multiagent resource allocation, social welfare optimization, computational complexity, auction and mechanism design

1. INTRODUCTION

In multiagent resource allocation (see, e.g., the survey by Chevaleyre et al. [2]) agents have preferences over bundles of resources. We consider preference representation by utility functions and assume that resources are indivisible and nonshareable. Taking the preferences of agents into account, the task is to allocate bundles of resources to agents. By aggregating the agents' utilities we arrive at the notion of social welfare with which we can assess the quality of allocations from the viewpoint of a global system designer.

One approach is the prominent *utilitarian social welfare*, which is the sum of the agents' utilities and which measures the average benefit every agent achieves. Utilitarian social welfare, however, lacks "fairness" because the utilities that agents realize in a given allocation can differ greatly. Interpreting the utilities as bids or valuations in a combinatorial auction, utilitarian social welfare corresponds to an auctioneer's revenue.

Egalitarian social welfare, in contrast, looks at the agent that is worst off and seeks to improves this agent's utility. While this concept provides some measure of fairness when the minimum needs of all agents are to be satisfied, it does have some disadvantages; for example, it is not strictly monotonic: Raising the utility of an agent who is not worst off does not increase egalitarian social welfare.

The *Nash product*, the product of the agents' utilities, can be seen as a compromise between these two approaches. On the one hand, it has the monotonicity property of utilitarian social welfare because an increase in any agent's utility leads to an increase of the Nash product (provided all agents have positive utility). On the other hand, the Nash product increases as well when reducing inequitableness among agents by redistributing utilities, thereby providing a measure of fairness. Looking at the ordering that is induced by the allocations, the *social welfare ordering*, Moulin [5] presents further beneficial properties of the Nash product. For example, the Nash product is uniquely characterized by independence of individual scale of utilities, i.e., even if different "currencies" are used to measure the agents' utilities, the social welfare ordering remains unaffected.

Having a measure for the quality of allocations, it is a natural task to optimize social welfare, and to ask for the computational complexity of this task.

2. PRELIMINARIES

Multiagent Resource Allocation

Let $A = \{a_1, \ldots, a_n\}$ be the set of *agents*, $R = \{r_1, \ldots, r_m\}$ the set of *resources* (which each are assumed to be indivisible and nonshareable), and let $U = \{u_1, \ldots, u_n\}$ be the set of the agents' *utility functions*. The mapping $u_i : 2^R \to \mathbb{F}$ is agent a_i's utility function, where 2^R denotes the power set of R and $\mathbb{F}$ is a numerical set (such as the set $\mathbb{N}$ of nonnegative integers, the set $\mathbb{Q}$ of rational numbers, and the set $\mathbb{Q}^+$ of nonnegative rational numbers). Such a triple (A, R, U) is called a *multiagent resource allocation setting* (a *MARA setting*, for short). An *allocation* for a given MARA setting (A, R, U) is a mapping $X : A \to 2^R$ with $\bigcup_{a_i \in A} X(a_i) = R$ and $X(a_i) \cap X(a_j) = \emptyset$ for any two distinct agents a_i and a_j. The set of all allocations for a MARA setting (A, R, U) is denoted by $\Pi_{A,R}$ and has cardinality n^m. We use the shorthand $u_i(X)$ to denote the utility $u_i(X(a_i))$ agent a_i can realize in allocation X. We consider the following representation forms for utility functions:

1. The *bundle form*: A utility function u is represented by a list of pairs $(R', u(R'))$ for any bundle $R' \subseteq R$, where pairs with $u(R') = 0$ are dropped.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. The *k-additive form* (Chevaleyre et al. [3] and Conitzer et al. [4]), for some fixed positive integer k: A utility function $u : 2^R \to \mathbb{F}$ is in k-additive form if there are coefficients $\alpha_T \in \mathbb{F}$ for each bundle $T \subseteq R$ with $\|T\| \leq k$ such that for any bundle $R' \subseteq R$ the following holds:

$$u(R') = \sum_{T \subseteq R', \|T\| \leq k} \alpha_T$$

DEFINITION 1. *For a MARA setting (A,R,U) and an allocation $X \in \Pi_{A,R}$, define*

1. *the* egalitarian social welfare of X *as* $sw_e(X) = \min_{a_i \in A}\{u_i(X)\}$*;*
2. *the* Nash product of X *as* $sw_N(X) = \prod_{a_i \in A} u_i(X)$.
3. *As an additional notation, for $S \in \{e,N\}$, denote the* maximum egalitarian/Nash product social welfare of a MARA setting $M = (A,R,U)$ *(or of a problem instance that contains a MARA setting M) by*

$$\max_S(M) = \max\{sw_S(X) \mid X \in \Pi_{A,R}\}.$$

For $\mathbb{F} \in \{\mathbb{N}, \mathbb{Q}^+, \mathbb{Q}\}$ and $\texttt{form} \in \{\texttt{bundle}, k\texttt{-additive}\}$, define:

$\mathbb{F}$-NASH PRODUCT SOCIAL WELFARE OPTIMIZATION$_{\texttt{form}}$	
Given:	A MARA setting $M = (A,R,U)$, where form indicates how every $u_i : 2^R \to \mathbb{F}$ in U is represented, and $t \in \mathbb{F}$.
Question:	Is there an allocation $X \in \Pi_{A,R}$ such that $sw_N(X) \geq t$?

which we abbreviate by $\mathbb{F}$-NPSWO$_{\texttt{form}}$. The exact version of this problem is denoted by $\mathbb{F}$-EXACT NASH PRODUCT SOCIAL WELFARE OPTIMIZATION$_{\texttt{form}}$ (or $\mathbb{F}$-XNPSWO$_{\texttt{form}}$, for short) and asks, given a MARA setting $M = (A,R,U)$ and $t \in \mathbb{F}$, whether $\max_N(M) = t$. The corresponding problems for utilitarian and egalitarian social welfare can be defined analogously and have been studied, e.g., by Chevaleyre et al. [2] and Roos and Rothe [10].

As the goal is to find a feasible allocation that maximizes social welfare, we also consider the corresponding maximization problems.

$\mathbb{F}$-MAXIMUM EGALITARIAN SOCIAL WELFARE$_{\texttt{form}}$	
Input:	A MARA setting $M = (A,R,U)$, where form indicates how every $u_i : 2^R \to \mathbb{F}$ in U is represented.
Output:	$\max_e(M)$.

As a shorthand, write $\mathbb{F}$-MAX-ESW$_{\texttt{form}}$. Based on sw_N, define $\mathbb{F}$-MAXIMUM NASH PRODUCT SOCIAL WELFARE$_{\texttt{form}}$ (or $\mathbb{F}$-MAX-NPSW$_{\texttt{form}}$) accordingly.

Complexity Theory and Theory of Approximation

We assume that the reader is familiar with the basic notions of computational complexity theory (see, e.g., the textbooks by Papadimitriou [8] and Rothe [11]).

Papadimitriou and Yannakakis [9] introduced the complexity class $\mathrm{DP} = \{L_1 - L_2 \mid L_1, L_2 \in \mathrm{NP}\}$, which contains the differences of any two NP-problems.

DEFINITION 2. *An α-*approximation algorithm *for an optimization problem is a polynomial-time algorithm that for all instances of the problem produces a solution whose value is within a factor of α of the value of an optimal solution.*

DEFINITION 3. *A maximization problem Π has a* polynomial-time approximation scheme *(*PTAS*) if for every ε, $0 < \varepsilon < 1$, there exists an ε-approximation algorithm for Π.*

3. RESULTS

We use a sufficient condition for DP-hardness by Chang and Kadin [1] to obtain the following complexity results:

THEOREM 4. *$\mathbb{Q}^+$-XNPSWO$_{\texttt{bundle}}$ is* DP*-complete.*

THEOREM 5. *For each $k \geq 3$, $\mathbb{Q}^+$-XNPSWO$_{k\texttt{-additive}}$ is* DP*-complete.*

Turning to approximability, we note that a reduction mentioned in [10] and attributed to a reviewer of that paper provides these kinds of inapproximability results.

PROPOSITION 6. *The problems $\mathbb{Q}$-MAX-ESW$_{\texttt{bundle}}$ and $\mathbb{Q}$-MAX-NPSW$_{\texttt{bundle}}$ cannot be approximated within any factor in polynomial time, unless* P = NP. *This result holds even when the utilities are restricted to the domain $\{0,1\}$.*

THEOREM 7. *$\mathbb{Q}^+$-MAX-NPSW$_{\texttt{1-additive}}$ can be solved exactly in polynomial time when the number of agents and resources are the same and the empty bundle has always utility zero.*

THEOREM 8. *There is a PTAS for $\mathbb{Q}^+$-MAX-NPSW$_{\texttt{1-additive}}$ when restricted to only two agents having the same utility function u with $u(\emptyset) = 0$.*

More details can be found in [6, 7].

4. ACKNOWLEDGMENTS

This work was supported in part by DFG grant RO-1202/14-1, a DAAD grant for a PPP project in the PROCOPE programme, and a fellowship from the Vietnamese government. We gratefully acknowledge interesting discussions with Ulle Endriss, and we thank the anonymous reviewers for their helpful comments.

5. REFERENCES

[1] R. Chang and J. Kadin. On computing boolean connectives of characteristic functions. *Theory of Computing Systems*, 28(3):173–198, 1995.

[2] Y. Chevaleyre, P. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[3] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163:49–62, 2008.

[4] V. Conitzer, T. Sandholm, and P. Santi. Combinatorial auctions with k-wise dependent valuations. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 248–254. AAAI Press, 2005.

[5] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2004.

[6] N. Nguyen, M. Roos, and J. Rothe. Exact optimization of social welfare by the Nash product is DP-complete. In *Website Proceedings of the 12th International Symposium on Artificial Intelligence and Mathematics*, Jan. 2012.

[7] T. Nguyen, M. Roos, and J. Rothe. A survey of approximability and inapproximability results for social welfare optimization in multiagent resource allocation. In *Website Proceedings of the Special Session on Computational Social Choice at the 12th International Symposium on Artificial Intelligence and Mathematics*, Jan. 2012.

[8] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.

[9] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[10] M. Roos and J. Rothe. Complexity of social welfare optimization in multiagent resource allocation. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 641–648. IFAAMAS, May 2010.

[11] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

When speed matters in learning against adversarial opponents

(Extended Abstract)

Mohamed Elidrisi
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
elidrisi@cs.umn.edu

Maria Gini
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
gini@cs.umn.edu

ABSTRACT

We propose a novel algorithm that is able to learn and adapt to an opponent even within a limited number of interactions and against a rapidly adapting opponent. The context we use is two player normal form games. We compare the performance of an agent using our algorithm against agents using existing multiagent learning algorithms.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Algorithms, Economics, Experimentation

Keywords

Multiagent Learning, Game Theory, Adaptive Learning

1. INTRODUCTION

A challenging issue in the design of intelligent agents is how to endow them with the ability to interact with other intelligent agents. Multiagent learning is primarily concerned with the problem of learning and acting in the presence of opponents. Multiagent learning has received considerable attention in the past decade from the research community, which has produced a wide range of learning agents and a set of criteria for developing them. Within the AI community, the problem has been addressed in multiple ways, either by adapting single agent reinforcement learning algorithms for multiagent settings [3], or combining policy search with knowledge of the adversarial nature of the opponent [1], or from a game theoretic perspective [4].

One of the major constraint typically assumed is that the opponent is either stationary or will converge to a stationary policy [1]. The stationarity assumption has been relaxed to some degree (e.g. [4]), but there are still critical assumptions that limit the use of learning agents in real world domains. We investigate two of those. The first relates to the need for extremely long sequences of interactions between the agents, often in the order of hundreds of thousand, before the agent learns a policy to use against the opponent. The second relates to the fact that abrupt changes in the opponent's play often require to restart the learning process.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. A NEW ALGORITHM: FAL

We propose a novel algorithm, *Fast Adaptive Learner (FAL)*, to learn a strategy to use when playing a sequence of games against an opponent. A strategy in a repeated game is a mapping from the history of actions to a probability distribution over the actions. The key feature of our algorithm is the ability to learn in a limited number of interactions and to detect and adapt to potentially fast changes in the opponent's strategy.

The algorithm, at a high level, uses two models:

1. a *Predictive Model* which makes a prediction about the opponent's next action. The predictive model has to be online in nature with some decay function over the history of interactions. It also has to view the interactions as a sequential prediction problem not as independent predictions and detect abrupt changes in the interactions.
2. a *Reasoning Model* which chooses a suitable best response accordingly. The reasoning model needs a belief model of whether the opponent is cooperative or competitive and the ability to explore if the opponent is teachable. It should also be able to measure the success of the predictive model in addition to maintaining a target average reward as a safety value.

There is a large class of models and methods that can be used for both parts of the algorithm. We made specific choices for the models used in our experiments, but we are not limited to the models we used. It is important to note that the memory of the predictive model limits the target class of opponents FAL can adapt to.

3. EXPERIMENTAL RESULTS

We have instantiated FAL's predictive model with ELPH and its reasoning model with Godfather-Future.

ELPH [2] is an an online predictive algorithm that learns to predict from short sequences. ELPH keeps a hypotheses space with the patterns observed and predictions sets that are updated constantly and pruned using entropy.

3.0,3.0	0.0,5.0
5.0,0.0	1.0,1.0

Table 1: Prisoner's Dilemma game matrix

	Q1	WOLF-PHC	FAL	God Father	Bully
Q1	1.7,1.7	1.7,1.7	2.2,2.2	2.4,2.4	0.9,1.4
WOLF-PHC		1.9,1.9	2.2,2.2	2.4,2.4	0.9,1.4
FAL			3.0,3.0	3.0,3.0	1.0,1.0
GodFather				3.0,3.0	1.0,1.0
Bully					1.0,1.0

Table 2: Average pairwise payoffs after playing 100 rounds of Prisoner's Dilemma. Results are from 1000 runs.

To explain Godfather-Future, we need a few concepts from game theory. A *security value* is the strategy that maximizes the player's own minimum payoff. A *targetable pair* is any pair of deterministic strategies in the game with the property that it yields a reward for the player higher than its security value. The Godfather-Future strategy computes a targetable pair of actions that leads to higher reward than its security value. The original Godfather [3] plays its part of the targetable pair if the opponent played its half of the targetable pair in the last interaction, Godfather-Future plays its part if the opponent is predicted to play its part in the next interaction.

Experiment 1. We compared experimentally the performance of different learning algorithms, using two-player repeated normal form games. The results in Table 2 show the outcomes of playing Prisoners Dilemma for 100 iterations. We repeated each of the 100 iterations 1000 times to reduce noise. The performance of FAL is compared against a set of algorithms and strategies from the literature, specifically Q-Learning, WOLF-PHC [1], Bully, and Godfather [3].

In Prisoner's Dilemma, shown in Table 1, the dominant strategy is to defect (D) and receive a reward of 1.0. Cooperating (C) would lead to a higher outcome of 3.0 but with the added risk of getting 0 if the opponent decided to betray.

From Table 2 it is clear that FAL and Godfather are the best performing methods across the board. Q-Learning and WOLF-PHC were among the worst especially against a stationary policy like Bully. It is important to note that Q-Learning and WOLF-PHC will perform as well as the other agents in longer sequences of games but our goal in this research is to analyze short term performance.

Experiment 2. In Experiment 1 we have shown that FAL is able to learn faster and achieve better results than Q-Learning, WOLF-PHC, and Bully. However, the performance of Godfather and FAL were almost identical in many scenarios. In order to show the ability of FAL to adapt rapidly we present now results against an opponent that changes its strategy after some period of time. Detecting the change and adapting to it is the real advantage that we are aiming at achieving in this work.

We use as opponent an agent we call *Switch*. The agent starts by following the classical Godfather strategy until it reaches stage 40 of the game. After that, Switch follows a deterministic repeated sequence of actions C, D, C, C, D, C indefinitely. This agent is designed to be deterministic and predictable with a bounded memory.

In this experiment, Switch played a sequence of 100 games against FAL, Godfather, and WOLF-PHC. Figure 1 shows the average reward over time for the 3 agents against Switch. Positive values imply that Switch is getting more reward, 0 are ties, and negative values are the others. It is evident in the graph that FAL is the best performing agent. FAL is able to detect and adapt in less than 20 games to the opponent's policy changes while the rest of the agents were not able to detect it until end of sequence at game 100.

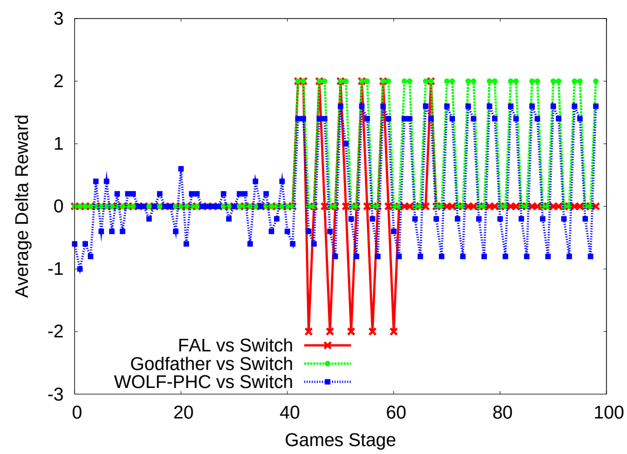


Figure 1: Average delta reward for the 3 Agents vs. Switch agent.

4. CONCLUSIONS AND FUTURE WORK

The goal of this work is to motivate and introduce the need for new requirements on multiagent learning algorithms, specifically to create agents that learn after playing a limited number of games against an opponent and that are capable of adapting to sudden and frequent changes in the opponent strategy. We proposed a new algorithm, FAL, and demonstrated experimentally that FAL outperforms agents using other learning methods in the Prisoner's Dilemma and against an abrupt policy changing opponent. Future work will be directed at examining theoretical properties of FAL, applying it to a larger class of games, and expanding the algorithm to play against more than one opponents.

5. REFERENCES

[1] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[2] S. Jensen, D. Boley, M. Gini, and P. Schrater. Non-stationary policy learning in 2-player zero sum games. In *Proc. Nat'l Conf. on Artificial Intelligence*, pages 789–794. AAAI Press, 2005.

[3] M. Littman and P. Stone. Leading best-response strategies in repeated games. In *Int'l Joint Conf. on Artificial Intelligence Workshop on Economic Agents, Models, and Mechanisms*, 2001.

[4] R. Powers, Y. Shoham, and T. Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1):45–76, 2007.

Do Experts Help in Two-Sided Search? (Extended Abstract)

Yinon Nahum
Weizmann Institute of Science
yinonn@gmail.com

David Sarne
Bar-Ilan University
david.sarne@gmail.com

Sanmay Das
Rensselaer Polytechnic Inst.
sanmay@cs.rpi.edu

Onn Shehory
IBM Research
onn@il.ibm.com

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics

Keywords

Sequential Search, Two-Sided Search, Social Welfare

1. INTRODUCTION

We study agents matching to form teams in a distributed multi-agent environment. Each agent receives information about the potential value of teaming with others. This information signal may be noisy. If all candidate agents agree to the matching the team is formed and each agent receives the true unknown utility of the matching, and leaves the market. We consider the effect of the presence of information brokers, or experts, on the outcomes of such matching processes. Experts can, upon payment of a fee, perform the service of finding and revealing the true value of a match to any agent. We analyze the equilibrium formed in the two-sided search setting, given the fee set by a monopolist expert. We then derive the revenue maximizing strategy for the expert as the first mover in a Stackelberg game. We find that better information can hurt: the presence of the expert, even if the use of its services is optional, can degrade individual agents' utilities and overall social welfare. While in one-sided search the presence of the expert can only help, in two-sided search the externality imposed by the fact that others are consulting the expert can lead to a situation where the equilibrium outcome is that everyone consults the expert, even though all agents would be better off if the expert were not present. As an antidote, we show how market designers can enhance welfare by subsidizing the expert to make her services more expensive, instead of providing conventional subsidies which reduce consumer costs.

2. MODEL

Our model is based on a standard two-sided distributed search model [1, 2], augmented to include uncertain signals.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The model assumes fully rational self-interested agents, searching for appropriate partners to form mutually acceptable pair-wise partnerships.

The number of agents may be either infinite or finite and all agents are ex ante identical, in that there are no individuals who are "naturally" better than others. However, when a potential match is formed, each agent gets some idiosyncratic utility from the particular qualities of that partnership. This utility is drawn anew each time a partnership with the same agent is evaluated in later stages of the search (as the number of agents in the population grows large, this becomes increasingly unlikely, since potential partnerships are drawn at random from the population; however, even with a relatively small number of agents, it models cases where the utility of a partnership is dependent on the circumstances in which it is formed).

At any period, the matching technology arranges a meeting between two agents, each of whom pays a search cost c_s and receives a different, independent noisy signal, denoted s, indicating the estimated value of the match to it. We assume that agents are acquainted with the distribution of signals $f_s(s)$ and the conditional probability density of values given signals, $f_v(v|s)$. Upon receiving a signal, an agent can either accept the partnership, decline it, or pay a cost c_e to consult an expert who then reveals to that agent the (noiseless) true value of the partnership to that agent. If the agent does consult the expert, it must decide whether to accept or decline the partnership once it receives the true value. If both agents decide to accept the partnership, a match takes place and the agents leave the market. If either one of the agents declines the partnership, the agents go back into the searching population and continue their search by sampling another partnering opportunity at search cost c_s, and so on.

Agents are rational and self-interested; they maximize expected utility (the value they receive from the partnership they eventually form minus the accumulated costs of querying the expert and interacting with other agents along the search path). In addition, the expert is a rational, self-interested monopolist; her goal is to maximize her own expected utility: the accumulated payment she receives from the agents minus her expenses, denoted d_e, which are a function of the cost of producing the information required to inform agents of the exact values of matches.

3. ANALYSIS

Any searcher can query an expert at cost c_e to find out the true value (to her) of a potential partner. The searcher has 3 alternatives. She can (1) reject the current potential partnership without querying the expert, paying search cost c_s to reveal the signal for the next potential partnership; (2) query the expert to obtain the true value v, paying a cost c_e, and then decide whether to accept the partnership; or (3) accept the current partnership without querying the expert. If both potential partners accept then the search terminates. Case (2) termination provides the searcher with the true value v. Case (3) termination provides the searcher with the (unknown) true value of the partnership. With no mutual acceptance, the search resumes.

A solution for a general density function $f_v(v|s)$ dictates an optimal strategy with a complex structure of the form of (S', S'', V), where: (a) S' is a set of signal intervals for which the searcher should resume her search without querying the expert; (b) S'' is a set of signal intervals for which the searcher should accept the partnership without querying the expert; and (c) for any signal that is not in S' or S'' the searcher should query the expert, and accept the partnership if the value obtained is above a threshold V, and resume otherwise. The value V is the expected utility from resuming the search given that the other agents use strategy $(S'_{\text{others}}, S''_{\text{others}}, V_{\text{others}})$and is given by:

$$V(S', S'', V) = -c_s - c_e \int_{s \notin \{S', S''\}} f_s(s)\, ds + (1 - A \cdot B) \cdot V(S', S'', V) + B \cdot C \qquad (1)$$

where A is the probability that the searcher accepts the partnership eventually (either directly or after querying the expert), B is the probability that the potential partner accepts the match, and C is the searcher's expected utility if both sides accept the partnership; these are given by:

$$A = \int_{s \in S''} f_s(s)\, ds + \int_{s \notin \{S', S''\}} f_s(s)\Big(1 - F_v(V|s)\Big)\, ds$$

$$B = \int_{s \in S''_{\text{others}}} f_s(s)\, ds + \int_{s \notin \{S'_{\text{others}}, S''_{\text{others}}\}} f_s(s)\Big(1 - F_v(V_{\text{others}}|s)\Big)\, ds$$

$$C = \int_{s \in S''} f_s(s) E[v|s]\, ds + \int_{s \notin \{S', S''\}} \left(f_s(s) \int_{y=V}^{\infty} y f_v(y|s)\, dy \right) ds$$

The value of $V(S', S'', V)$ in Equation 1 is derived recursively, considering the next search iteration. The searcher pays c_s for receiving the noisy signal. The next element is the expected expert query cost, incurred whenever receiving a signal $s \notin \{S', S''\}$. The third element applies to the case of resuming search, when at least one of the sides rejects the partnership, in which case the searcher continues with an expected utility $V(S', S'', V)$. The last element applies to the case where the search is terminated, since both sides accepted the opportunity. Similarly, the first element in A and B applies to a case where the searcher accepted the match without querying the expert and the second applies to a case where the searcher accepted the match after querying the expert. The first element in C applies to a case where the searcher accepted the match without querying the expert, in which case the expected revenue is $E[v|s]$. The second element applies to the case where the searcher accepted the match after querying the expert.

Expected profit of the expert: The expected profit of the expert is: $\pi_e = \mathbb{E}(\text{Profit}) = (c_e - d_e)\eta_{c_e}$, where η_{c_e} is the expected number of expert queries a searcher performs. The expert can maximize the above expression with respect to c_e (η_{c_e} decreases as c_e increases) to find the profit maximizing price to charge searchers.

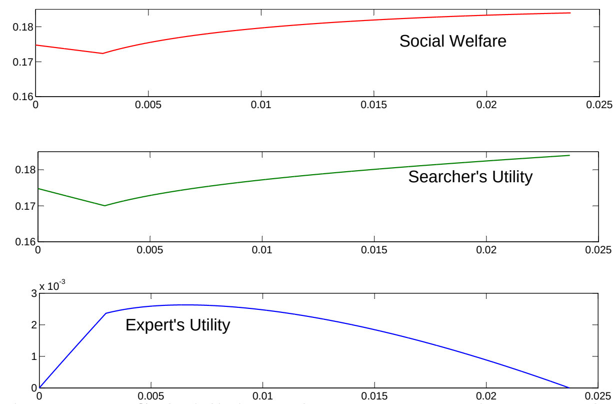


Figure 1: Subsidizing the expert to increase her query price from 0.0065 to 0.0237, thus maximizing social welfare. In this example $c_s = 0.1$.

4. ILLUSTRATIVE EVALUATION

As an example, we consider a synthetic environment where agents form pairwise partnerships. The signal is an upper bound on the true value (e.g., people tend to get a good first impression of others). Specifically, we assume signals s are uniformly distributed on $[0, 1]$ ($f_s(s) = 1$ if $0 < s < 1$ and zero otherwise) and the conditional density of true values is a monotonic increasing function in the interval $[0, s]$: $f_v(y|s) = \frac{3\sqrt{y}}{2\sqrt{s^3}}$.

A market designer can motivate the expert to modify her query price by changing the expert's incentives, in order to increase social welfare. The expert computes the profit-maximizing cost c_e to charge, given that individual agents play their optimal search strategies subject to c_s. For instance, for $c_s = 0.1$, the optimal expert query cost is $c_e = 0.0065$ (for example, see Figure 1, where the lower curve, which demonstrates the expert's profit as a function of query cost, peaks at 0.0065; note, however, that social welfare is not maximized at $c_e = 0.0065$).

In the case of one-sided search, social welfare maximization typically involves reducing the expert's query cost. However, in keeping with our finding that more information can hurt social welfare in two-sided search, in many settings a *reverse subsidy* can be optimal. That is, for increasing social welfare, it is necessary that the expert increase her query price. Figure 1 presents one example. In this case, social welfare (taking into account the subsidy) is maximized when the query price is 0.0237 (seen at the upper curve in the figure). The optimal subsidy is so high that the expert never gets used – in this case the mechanism is essentially paying the expert to leave the market.

Acknowledgments

This work was supported in part by ISF/BSF grants 1401/09 and 2008-404 and in part by NSF award IIS-0952918.

5. REFERENCES

[1] K. Burdett and R. Wright. Two-sided search with nontransferable utility. *Review of Economic Dynamics*, 1:220–245, 1998.

[2] H. Chade and G. Ventura. On income taxation and marital decisions. *Review of Economic Dynamics*, 8(3):565–599, July 2005.

The Benefits of Search Costs in Multiagent Exploration (Extended Abstract)

David Sarne
Bar-Ilan University
david.sarne@gmail.com

Yonatan Aumann
Bar-Ilan University
yaumann@gmail.com

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics

Keywords

Sequential Search, Two-Sided Search, Social Welfare

1. INTRODUCTION

Humans and software agents alike spend considerable time and effort in searching. Search enables finding the things that better fit and agent's goals. But search can also be a costly process. Search costs can either come in the form of direct monetary payments, or in the form of time and resources spent. In general, the searcher must balance between the benefits provided by longer and broader search, on the one hand, and the associated increased cost, on the other.

In economic literature search costs are often referred to as "environment friction" or "market *inefficiency*" and associated with reduced market performance [1]. Indeed, in the presence of search costs a rational player will not aim to find the best option, but rather settle for the "good enough", beyond which the marginal cost of searching exceeds the marginal benefit of continuing the search. Thus, search costs promote sub-optimal results (or so it would seem). As such, the traditional wisdom is that when designing a MAS environment, search costs should be avoided or reduced to a minimum. Taking eCommerce as an example, most researchers see a great benefit in the ability of eMarketplaces to lower the buyers' cost to obtain information (e.g. about the price features) from multiple sellers, as well as the sellers' reduced costs to communicate their information [1]. The lowered search cost is associated in this case with increased economic efficiency and enable new markets to emerge. Similarly, many systems have been introduced in which central mechanisms or mediators are used in order to supply the agents complete information concerning market opportunities, eliminating the need to engage in costly search.

In this paper we show that, not withstanding the above, search costs – "friction", if and when applied appropriately, can also be beneficial, and actually *increase* both the expected aggregate welfare, and the expected utility of each individual agent. This holds even if the proceeds from the search costs are discarded and do not directly benefit anyone in the system, as we assume throughout. Thus, artificially applied search costs can be used as a mechanism to improve market efficiency. We show this for one-sided search settings using standard models from *search theory*. Similar results for two-sided search settings are available however omitted for space considerations.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ONE-SIDED SEARCH

The Model.

We employ the fixed-sample-size search model [2], wherein each searcher executes a single search round in which it obtains a large set of opportunities simultaneously, and chooses the one associated with the highest utility. Consider an environment with m homogeneous servers, and N homogeneous agents requesting service from these servers. The agents are assigned a random order. Each agent, in turn, can request to query any number of servers. The queries are assigned to available servers. Each server can address one query in each time step. Since the queries may be executed in parallel, agents need to determine the number of queries they request in advance. Once the queries are executed, the agent obtains the results and leaves the system.

Each reply provides the agent with some non-negative utility. The utility, x, obtained by agent A_j from the reply of server i is randomly drawn from a distribution D_U characterized by a probability distribution function (p.d.f.) $f(x)$, and cumulative distribution function (c.d.f.) $F(x)$. For simplicity we assume that all servers and all agents are homogeneous, and thus share the same functions $f(x)$ and $F(x)$. The overall utility obtained by the agent from the set of all replies is the maximum among the utilities provided by the individual replies from the different servers. We assume that the future is discounted by a factor of δ (which is common to all agents).

Agents are assumed to be self-interested, and thus aim to maximize their own expected utility. Thus, if there is no cost for querying a server, all agents will request to query all servers. This, however, means that serving each agent takes more resources, and thus delays serving other agents. Since the future is discounted, agents further down the line actually end up losing from this delay more than they gain from accessing more serves. We show that by introducing a cost for each query, we can drive agents to perform less queries, and increase the expected utility.

No Search Costs.

For a search cost of c, let S_c be the expected aggregate utility with a search cost of c. For any $k = 1, \ldots, m$, let $E_k = E(\max\{x_1, \ldots, x_k : x \in D_U\})$ be the expected maximum of k independent draws from the utility distribution D_U. Then, if there are no search costs, ignoring discounting, each agent would obtain expected utility of E_m. However, each agent occupies all the m available servers. Thus, the i-th agent is only serviced at time i. Thus, taking discounting into account, the overall expected utility of the i-th agent is $E_m\delta^i$. The aggregate expected utility, summed up over all agents, is thus:

$$S_0 = \sum_{i=1}^{N} E_m\delta^i = \delta E_m \frac{1-\delta^N}{1-\delta} \underset{N\to\infty}{\longrightarrow} \frac{\delta E_m}{1-\delta} \tag{1}$$

With Search Costs.

Now, assume that we introduce a cost c for each query. Then, the rational choice for an agent is to query $k \leq m$ servers such that the expected marginal utility of querying the k-th server (rather than $k-1$ servers) is at least c, but the marginal utility of querying the $k+1$ server is less than c. Thus, each agent will choose to query k servers such that $k = \arg\max_k\{E_k - E_{k-1} \geq c\}$. Conversely, the minimum cost that will guarantee querying exactly k servers is: $c_k = E_{k+1} - E_k$.

With a search cost of c_k, ignoring discounting, the expected utility of each agent is: $U_k = E_k - k \cdot c_k = (k+1)E_k - kE_{k+1}$. At any one time step, m/k agents can be served in parallel (assuming k divides m). Thus, the i-th agent is served at time $\left\lfloor \frac{i}{m/k} \right\rfloor$. Thus, taking into account discounting, the expected utility of the i-th agent is $U_k\delta^{\left\lfloor \frac{i}{m/k} \right\rfloor}$. Thus, the total expected utility, summed up over all agents, is:

$$S_{c_k} = \sum_{i=1}^{\left\lfloor \frac{N}{m/k} \right\rfloor} \frac{m}{k} U_k\delta^i = \frac{m}{k}\delta U_k \frac{1-\delta^{\left\lfloor \frac{N}{m/k} \right\rfloor}}{1-\delta} \underset{N\to\infty}{\longrightarrow} \frac{m}{k}\frac{\delta U_k}{1-\delta} \tag{2}$$

Advantageous Search Costs - Aggregate Utility.

Theorem 1. *For any non-degenerate distribution D_U on non-negative utilities and any discounting factor $\delta < 1$, there exist m_0 and N_0 such that for any $m \geq m_0$ and $N \geq N_0$, there exists a c such that introducing a search cost of c for each query increases the expected aggregate utility. This holds even if the proceeds of the search costs are discarded and do not benefit anyone.*

Proof. Set $\vec{S_0} = \frac{\delta E_m}{1-\delta}$ and $\vec{S_{c_k}} = \frac{m}{k}\frac{\delta U_k}{1-\delta}$. By (1) and (2) we have that $S_0 \underset{N\to\infty}{\longrightarrow} \vec{S_0}$ and $S_{c_k} \underset{N\to\infty}{\longrightarrow} \vec{S_{c_k}}$. Suppose that $\vec{S_{c_k}} > \vec{S_0}$. Set $\epsilon = \vec{S_{c_k}} - \vec{S_0}$. Then, there exists an N_0 such that for any $N \geq N_0$, $S_0 < \vec{S_0} + \epsilon/2$ and $S_{c_k} \geq \vec{S_{c_k}} - \epsilon/2$. Thus, for $N \geq N_0$ we have that $S_{c_k} > S_0$, i.e. introducing a search cost of c_k increases aggregate utility.

It thus remains to prove that $\vec{S_{c_k}} > \vec{S_0}$. We show that this holds for any k, provided that m is sufficiently large. Indeed, $\vec{S_{c_k}} = \frac{m}{k}\frac{\delta U_k}{1-\delta} > \frac{\delta E_m}{1-\delta} = \vec{S_0}$ iff

$$\frac{U_k}{k} > \frac{E_m}{m} \tag{3}$$

The left hand side of (3) is independent of m, while the right hand side approaches 0 as m grows [2]. Thus, provided that U_k is positive, (3) necessarily holds for m sufficiently large.

We show that U_k is positive for any k. Denote $f_k(x)$ the p.d.f. of the maximum of k independent samples from D_U, and let $F_k(x)$ be the associated c.d.f. Then, $F_k(x) = (F(x))^k$ and $f_k(x) = (F_k(x))' = k(F(x))^{k-1}f(x)$. By definition, $E_k = \int_0^\infty y f_k(y)dy$. Thus,

$$U_k = (k+1)E_k - kE_{k+1} =$$
$$(k+1)\int_0^\infty f_k(y)ydy - k\int_0^\infty f_{k+1}(y)ydy =$$
$$k(k+1)\int_0^\infty (F(y))^{k-1}(1-F(y))f(y)ydy > 0$$

The last inequality is due to the fact that all elements of the integral are non-negative, and assuming that the distribution is non-degenerate (i.e. is not concentrated all in one value) at least one element is strictly positive. □

Individual Utility.

Corollary 1. *For any non-degenerate distribution D_U on non-negative utilities and any discounting factor $\delta < 1$, if agents are assigned a random order then there exist m_0 and N_0 such that for any $m \geq m_0$ and $N \geq N_0$, there exists a c such that introducing a search cost of c for each query increases the expected utility of each player. This holds even if the proceeds of the search costs are discarded and do not benefit anyone.*

Proof. Considering a specific player, for any position i, the probability that the player is i-th in the order is $1/N$. Thus, if there are no search costs then the expected utility of any player is:

$$\sum_{i=1}^{N} \frac{1}{N} E_m\delta^i = \frac{1}{N} S_0 \tag{4}$$

Similarly, the expected utility of the player with a search cost of c_k, is

$$\sum_{i=1}^{\left\lfloor \frac{N}{m/k} \right\rfloor} \frac{1}{N}\frac{m}{k} U_k\delta^i = \frac{1}{N} S_{c_k} \tag{5}$$

Thus, the theorem follows by the exact same reasoning as that in the proof of Theorem 1. □

3. CONCLUSIONS

The implication of these results to market designers is that the effects of search costs should be carefully analyzed in each case, and not assumed to be universally detrimental. Rather, there are cases when it may be beneficial to deliberately introduce artificial search costs. When search costs are already part of the system, there is no general answer for whether or not decreasing these costs will improve the system's performance. In some settings, an increase rather than a decrease can actually contribute to improving expected utility. In other cases, a decrease in search costs can contribute to improving expected utility, but decreasing the costs beyond a certain point can result with the opposite effect. The analysis methodology given in this paper can facilitate the calculation of the right search cost to which the market designer should strive.

Acknowledgments

This work was supported in part by ISF/BSF grants 1401/09 and 2008-404.

4. REFERENCES

[1] Y. Bakos. Reducing buyer search costs: Implications for electronic marketplaces. *Mgmt. Sci.*, 42:1676–92, 1997.

[2] G. Stigler. The economics of information. *Journal of Political Economy*, 69(3):213–225, 1961.

Adaptive Negotiating Agents in Dynamic Games: Outperforming Human Behavior in Diverse Societies

(Extended Abstract)

Eunkyung Kim, Luyan Chi, Yu Ning, Yu-Han Chang, Rajiv Maheswaran
Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
{eunkyung, lchi, yuning}@usc.edu, {ychang, maheswar}@isi.edu

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms, Economics, Experimentation

Keywords

Multi-Agent Systems, Game Theory, Ultimatum Game, Mathematical Models of Human Behavior, Learning, Adaptation

1. INTRODUCTION

Creating software agents that can negotiate effectively is an important problem that has been studied by agent researchers in contexts such as the trading agent competition and the virtual agents community. In the former, the goal is typically to find optimal policies in settings with uncertain and incomplete information, and where policies are typically evaluated in societies of entirely artificial agents [6]. In the latter, a goal is to create agents that can interact with humans — in many cases, to train them in negotiation with individuals from particular cultures or different value settings [5].

In this paper, we examine a problem that combines the complexities of these goals. We want to create negotiating agents that can perform effectively in multiple environments, specifically in a multitude of societies where values and styles of negotiation might be significantly different. Since agent performance is highly dependent on the interaction environment, the design of such an agent is not a straightforward optimization problem.

As context for this investigation, we use the Social Ultimatum Game [2], a multi-agent multi-round extension of the Ultimatum Game, a classical game-theoretic problem which has been studied for decades due to the behavioral variance it elicits. It has been shown through many investigations that humans exhibit a wide range of behaviors that deviate from a "rational" payoff-maximizing strategy based on factors such as cultural background, occupation and emotional factors among others in the classical Ultimatum Game.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. SOCIAL ULTIMATUM GAME

The Ultimatum Game, is a two-player game where a player, P_1 proposes a split of an endowment $e \in \mathbb{N}$ to another player P_2 where P_2 would receive $q \in \{0, \delta, 2\delta, \ldots, e-\delta, e\}$ for some value $\delta \in \mathbb{N}$. If P_2 accepts the offer, they receive q and P_1 receives $e-q$. If P_2 rejects, neither player receives anything. The subgame-perfect Nash or Stackelberg equilibrium states that P_1 offer $q = \delta$, and P_2 accept. This is because a "rational" P_2 should accept any offer of $q > 0$, and P_1 knows this. Yet, humans make offers that exceed δ, even making "fair" offers of $e/2$, and reject offers less than the minimum.

To represent the characteristics that people operate in societies of multiple agents and repeated interactions, we introduce the Social Ultimatum Game. There are N players, denoted $\{P_1, P_2, \ldots, P_N\}$, playing K rounds, where $N \geq 3$. The requirement of having at least three players in necessary to give each player a choice of whom to interact with.

In each round k, every player P_m chooses a single potential partner P_n and makes an offer $q^k_{m,n}$. Each player P_n then considers the offers they have received and makes a decision $d^k_{m,n} \in \{0, 1\}$ with respect to each offer $q^k_{m,n}$ to either accept (1) or reject (0) it. If the offer is accepted by P_m, P_m receives $e - q^k_{m,n}$ and P_n receives $q^k_{m,n}$, where e is the endowment to be shared. If an offer is rejected by P_n, then both players receive 0 for that particular offer in round k. Thus, P_m's reward in round k is the sum of the offers they accept from other players (if any are made to them) and their portion of the proposal they make to another player, if accepted, $r^k_m = (e - q^k_{m,n})d^k_{m,n} + \sum_{j=1\ldots N, j\neq m} q^k_{j,m} d^k_{j,m}$. The total rewards for P_m over the game is the sum of per-round winnings, $r_m \sum_{k=1}^{K} r^k_m$.

3. AUTONOMOUS AGENTS

We summarize the types of agents that we implemented.

- **Tit-for-Tat** : This is a fully reciprocal agent that chooses responders who previously made them offers, and offers an amount that reciprocates that previous offer,
- **Regret Minimization** : This agent minimizes worst-case regret by hedging [1] among a set of available actions. It hedges by increasing the weights associated with high payoff actions during gameplay, and probabilistically chooses actions based on these weights, which are initialized using human data,
- **Expected Reward QRE** : This agent learns the ex-

pected rewards of various actions based on human play data and acts using a quantal response equilibrium [3] strategy based on these rewards.

- **SIGAL QRE** : This agent also uses a quantal response equilibrium strategy but the utility is based on the sigmoid acceptance learning [4] approach which incorporates a model of social utility into the rewards.
- **Adaptive Fairness** : This agent is characterized by a fairness threshold which is dynamically updated based on an adaptability parameter and an exploration parameter [2]. It accurately replicates human dynamic reciprocity behavior and is used as a stand-in for various human-like behaviors that are learned from data.
- **Marginal Value Optimization** : This agent chooses an action based on the marginal value of being seen as the preferred partner of each agent in the society. The value is a product of the expected value of the offer received from a particular agent and the marginal increase of the likelihood of receiving an offer.

4. EXPERIMENTS

In order to investigate adaptiveness of the agents and of the humans, we created 10 different societies. We first ran two sets of human experiments, one with undergraduates and staff at a U.S. university, and a second at an international conference with primarily computer science doctoral students and faculty. From this data we estimated parameters for the Adaptive Fairness (AF) agents, using different subsets of humans.

This includes the top 25% scorers at the conference, the top 25% scorers at the university, two clusters of the human population based on offer recipient entropy (people who spread their offers out the most and the least), and four humans drawn randomly from the populations. In addition, SIGAL-QRE, ER-QRE and Regret Minimization agents were created with data from the first two experiments. We then created the following 10 societies for 5-player games where one *test player* plays against four players:

- **AF-Conf-Top25** : 4 Conference Top 25% AF-agents
- **AF-Univ-Top25** : 4 University Top 25% AF-agents
- **AF-Cluster1** : 4 low recipient entropy AF-agents
- **AF-Cluster2** : 4 high recipient entropy AF-agents
- **AF-Alpha-7** : 4 AF-agents for Human #2
- **AF-4Types** : AF-agents for 4 types of humans
- **SIGAL-QRE** : 4 SIGAL-QRE agents
- **ER-QRE** : 4 ER-QRE agents
- **Regret** : 4 Regret Minimization agents
- **TFT-2** : 4 Tit-for-Tat agents with baseline $2 offers

We ran a third set of human experiments using Amazon Mechanical Turk where a human player could play against the societies above in 20-round games with a $10 endowment per round. We created on HIT (Human Intelligence Task) for each instance of a game in a society with 20 assignments, i.e., we had 20 human game play traces for each society type.

We then tested the following agents in each of the societies, running 1000 iterations of games for each: Regret, SIGAL-QRE, ER-QRE, TFT-2 and Marginal Value Optimization (MVO).

Mean of Payoffs	Society									
Test Player	AF-Conf-Top25	AF-Univ-Top25	AF-Cluster 1	AF-Cluster 2	AF-Alpha-7	AF-4Types	SIGAL-QRE	ER-QRE	Regret	TFT-2
Human	199.2	193.5	149.7	173.1	86.9	211.6	167.4	186.4	138.7	192.0
MVO	203.6	229.3	215.4	215.1	111.4	221.3	184.1	174.7	180.2	209.9
SIGAL-QRE	185.7	109.6	147.8	140.1	64.3	168.4	200.0	210.2	178.3	205.9
ER-QRE	146.2	138.5	141.5	131.5	88.0	167.4	190.8	200.0	170.5	202.7
Regret	172.3	98.4	143.9	140.9	62.7	162.1	224.1	231.3	200.0	199.0
TFT-2	93.4	162.9	95.7	113.8	104.8	167.3	209.0	207.5	199.3	200.0

Figure 1: Mean of Payoffs for Test Players

5. RESULTS

Figure 1 shows the mean of payoffs for the test players in the 10 different agent societies. The main result of the paper is that the marginal value optimization (MVO) agent outperforms human players in 9 out of 10 societies. In 7 out of 10 societies the gaps in mean payoff were very high (MVO advantages were 16.6, 17.9, 24.5, 35.9, 41.6, 42.1, 65.7) The only society where it does not outperform humans is the ER-QRE society (-11.7) which is made up of agents which follow a static policy. We see that MVO's assumptions about generating payoffs from others by being the top target is validated, as MVO is able to generate more payoffs from offers made to it by others, when compared to human players.

Furthermore, MVO is also able to generate more payoffs from its own offers when compared to humans in 7 out of 10 human societies. This is because the generous offer reduces the probability of rejection in several of the societies. However, it pays a price for this in societies where the probability of rejection is low (or zero). In two of the three cases, it is able to overcome this loss from improvement in the number and quality of offers made to it by others.

Acknowledgements. This material is based upon work supported by the AFOSR Award No. FA9550-10-1-0569 and ARO MURI award No. W911NF-11-1-0332.

6. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th Symposium on Foundations of Computer Science*, 1995.

[2] Y.-H. Chang, T. Levinboim, and R. Maheswaran. The social ultimatum game. In *Decision Making with Imperfect Decision Makers*, 2011.

[3] R. McKelvey and T. Palfrey. Quantal response equilibria for extensive form games. *Experimental Economics*, 1:9–41, 1998.

[4] N. Peled, Y. Gal, and S. Kraus. A study of computational and human strategies in revelation games. In *Ninth International Conference on Autonomous Agents and Multi-Agent Systems*, 2011.

[5] D. Traum, S. C. Marsella, J. Gratch, J. Lee, and A. Hartholt. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Proceedings of the 8th international conference on Intelligent Virtual Agents*, IVA '08, pages 117–130, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] M. P. Wellman, A. Greenwald, and P. Stone. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*. MIT Press, 2007.

A Robust Approach to Addressing Human Adversaries in Security Games

(Extended Abstract)

James Pita, Richard John, Rajiv Maheswaran, Milind Tambe, Rong Yang
University of Southern California, Los Angeles, CA 90089

Sarit Kraus
Bar-Ilan University, Ramat-Gan 52900, Israel and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742

ABSTRACT

While game-theoretic approaches have been proposed for addressing complex security resource allocation problems, many of the standard game-theoretic assumptions fail to address human adversaries who security forces will likely face. To that end, approaches have been proposed that attempt to incorporate better models of human decision-making in these security settings. We take a new approach where instead of trying to create a model of human decision-making, we leverage ideas from robust optimization techniques. In addition, we extend our approach and the previous best performing approach to also address human anchoring biases under limited observation conditions. To evaluate our approach, we perform a comprehensive examination comparing the performance of our new approach against the current leading approaches to addressing human adversaries. Finally, in our experiments we take the first ever analysis of some demographic information and personality measures that may influence decision making in security games.

Categories and Subject Descriptors

H.4 [**Computing Methodology**]: Game Theory

General Terms

Security, Algorithms, Performance

Keywords

Human Behavior, Stackelberg Games, Decision-making, Security

1. INTRODUCTION

Game-theory has gained traction in security resource allocation decisions in important settings [4]. Security games refer to a special class of Stackelberg games where there are two agents - the defender (security force) and an attacker - who act as the leader and the follower respectively [9]. Traditionally, Stackelberg games have been used to model these problems because they encapsulate the commitment a defender must make in allocating her security resources before an attacker chooses an attack method.

There exists a number of game-theoretic optimal algorithms for solving security games such as DOBSS [5]. However, one of the key assumptions underlying these approaches is that the attacker is a perfectly-rational player and that the attacker breaks ties in the defender's favor. Thus, these systems optimize their strategy against an expected-value-maximizing opponent and are not robust to deviations from this strategy. It is well known that standard game-theoretic assumptions of expected-value-maximizing rationality are not ideal for addressing human behavior in game-theoretic settings [2]. To that end, a number of approaches have attempted to address these potential deviations by incorporating more realistic models of human decision-making.

COBRA is one such approach that assumes a boundedly-rational opponent and attempts to maximize the defender's utility for the worst-case outcome of any ϵ-optimal response strategy, avoiding the issue of tie breaking by the attacker [6]. One critical issue with COBRA is that if the attacker deviates to any strategy beyond the ϵ-optimal response set then the result can once again be arbitrarily bad for the defender. To address this dilemma, Yang et al. [7] introduced BRQR, which assumes that instead of strictly maximizing expected value, the attacker responds stochastically: the chance of selecting non-optimal strategies increases as the cost of such an error decreases. BRQR thus allows for a more gradual approach to defending against deviations as opposed to the hard-cutoff point. Two issues with BRQR are that it critically depends on the appropriate estimation of λ, which represents the amount of error in the attacker's response function; and that its runtime is slow.

To attempt to address the issues of BRQR and COBRA, we introduce a new approach, MATCH, based on robust optimization [1] where the defender strategy is robust to certain worst-case deviations from the attacker, but modify the traditional worst-case assumption to a new type of graduated optimization. Furthermore, we extend both MATCH and BRQR to address human anchoring biases as it has been shown that this extension is advantageous under limited observation [6]. In order to evaluate our new approach and these extensions we performed a comprehensive experimental study involving 253 human subjects playing 5956 games under three observation conditions (perfect, limited, and no observation). Since we alter the standard assumptions of robust optimization we also include an alternative algorithm, RECON [8], which employs the traditional worst-case robust optimization. In addition, we examine the influence of two personality measures, psychopathy and numeracy, and demographic information, age and gender, on decision-making in security settings. Psychopathy is especially of interest because research has shown that psychopathy is a strong predictor of both criminal behavior and in particular violent crimes [3]. Gaining insight into the influence of such personality measures and demographic information could potentially motivate future algorithmic developments.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. METHODOLOGY

Methods for computing MATCH: MATCH is a mixed integer linear program (MILP) that utilizes a new idea of graduated robust optimization. Whereas standard robust optimization robustly guards against a worst-case outcome within some error bound, MATCH assumes a utility maximizing outcome on behalf of the attacker, but constrains the impact of deviations depending on the magnitude of the deviation. That is, MATCH has an adjustable parameter, β, which constrains the defender's loss for a deviation by the attacker to be no worse than a proportion (β) of the loss the attacker incurs for that deviation. For example, if the attacker deviates from the expected-value-maximizing target and loses 2 utility, then the defender should not lose more than $\beta * 2$ for this deviation.

Extending BRQR: In order to extend BRQR to handle an anchoring bias we need to alter the way the adversary perceives his reward. Specifically, if the defender has chosen a strategy x for defending her targets, the attacker will now base his decisions on a strategy x' that accounts for his anchoring biases. Thus, in the new model for BRQR, the adversary will respond stochastically according to x' where the chance of selecting non-optimal strategies increases as the *perceived* cost of such an error decreases. We refer to this new strategy as BRQRA.

Extending MATCH: MATCH originally assumes a perfectly-rational adversary so we chose to extend MATCH to address both an anchoring bias and a boundedly-rational attacker as in COBRA. We refer to this new formulation as COBRA-MATCH. Since MATCH and COBRA are both MILPs we are able to extend MATCH utilizing the same types of constraints originally presented in COBRA [6]. Specifically, as in BRQR, the attacker now makes his decision based on x' rather than x. Furthermore, given his perception of the defender strategy (i.e., x') he is willing to choose any strategy within ϵ of what he perceives to be the expected-utility-maximizing strategy. One important consideration in the COBRA-MATCH formulation is that now we must model the attacker's losses for a deviation according to his perception of his loss (i.e., according to x'), while the defender's loss is still based on the real defender strategy (i.e., according to x). It follows that the defender should only lose a proportion (β) of what the attacker *perceives* he has lost.

3. EVALUATION

We conducted empirical tests with human subjects playing a web-based game to evaluate the performance of defender strategies generated using six candidate algorithms: DOBSS, MAXIMIN, COBRA, BRQR/BRQRA, MATCH/COBRA-MATCH, and RECON. In our experiments, we utilize the same eight-target scenario used by Yang et al. [7].Before beginning, subjects were given a tutorial and a test to ensure that they understood the general game play.

Our experiments were run in Amazon Mechanical Turk and participants were paid a base amount of US $1.50 for participating. In order to motivate the subjects, they were informed that a small sample of their games would be chosen at random and they would be paid an additional US $0.15 for the total points earned in that sample. Also, two obvious games were introduced to ensure subjects were paying attention.If subjects failed to respond correctly in the obvious games then their data was removed from the set.

We tested nine different payoff structures (five new, four from yang et al. [7]) in the unlimited observation condition and four in the limited and unobserved conditions (from yang et al. [7]). For each payoff structure, we generated the mixed strategies for the defender using the six algorithms with a variety of parameter settings. We ran experiments for the unlimited observation condition separately from experiments in the limited and unobserved observation conditions. This was to avoid confusion in the subjects and to keep the experimental conditions controlled. Additionally, the order of game instances played by each subject was randomized to mitigate ordering effects on their response. We also examined runtime performance for MATCH versus BRQR.

4. CONCLUSIONS

To address human adversaries, a number of approaches, including COBRA and BRQR, have been introduced which attempt to include more realistic models of human-decision making. Our work provides five fundamental contributions to this line of research: (i) we develop an approach to addressing human adversaries based on robust optimization rather than relying on finding more appropriate models of human decision-making; (ii) we extend both BRQR and MATCH to address human anchoring biases under limited observation; (iii) we do a comprehensive experimental analysis of the performance of MATCH against previous approaches and runtime analysis showing the efficiency of MATCH; and (iv) we make the first ever evaluation of the influence of some demographic and personality measures on decision-making in security games.

5. ACKNOWLEDGEMENT

This research was supported by the United States Department of Homeland Security through the National Center for Risk and Economic Analysis of Terrorism Events (CREATE) under award number 2010-ST-061-RE0001. However, any opinions, findings, and conclusions or recommendations in this document are those of the authors and do not necessarily reflect views of the United States Department of Homeland Security, or the University of Southern California, or CREATE.

6. REFERENCES

[1] M. Aghassi and D. Bertsimas. Robust game theory. *Math. Program.*, 107(1-2):231–273, 2006.

[2] C. Camerer. In *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.

[3] R. D. Hare and C. S. Neumann. Psychopathy as a clinical and empirical construct. *Annual Review of Clinical Psychology*, 4:217–246, 2008.

[4] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, F. Ordóñez, and M. Tambe. Software assistants for randomized patrol planning for the LAX airport police and the Federal Air Marshals Service. *Interfaces*, 40(4):267–290, 2010.

[5] P. Paruchuri, J. Marecki, J. Pearce, M. Tambe, F. Ordóñez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *AAMAS*, 2008.

[6] J. Pita, M. Jain, M. Tambe, F. Ordóñez, and S. Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence Journal*, 174(15):1142–1171, 2010.

[7] R. Yang, C. Kiekintveld, F. Ordóñez, M. Tambe, and R. John. Improving resource allocation strategy against human adversaries in security games. In *IJCAI*, 2011.

[8] Z. Yin, M. Jain, M. Tambe, and F. Ordóñez. Risk-averse strategies for security games with execution and observational uncertainty. In *AAAI*, 2011.

[9] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*, 2010.

Designing Better Strategies against Human Adversaries in Network Security Games

(Extended Abstract)

Rong Yang, Fei Fang, Albert Xin Jiang,
Karthik Rajagopal*, Milind Tambe, Rajiv Maheswaran
University of Southern California, Los Angeles, CA, 90089
{yangrong,feifang,jiangx,tambe,maheswar}@usc.edu, *{nkartr}@gmail.com

ABSTRACT

In a Network Security Game (NSG), security agencies must allocate limited resources to protect targets embedded in a network, such as important buildings in a city road network. A recent line of work relaxed the perfect-rationality assumption of human adversary and showed significant advantages of incorporating the bounded rationality adversary models in non-networked security domains. Given that real-world NSG are often extremely complex and hence very difficult for humans to solve, it is critical that we address human bounded rationality when designing defender strategies. To that end, the key contributions of this paper include: (i) comprehensive experiments with human subjects using a web-based game that we designed to simulate NSGs; (ii) new behavioral models of human adversary in NSGs, which we train with the data collected from human experiments; (iii) new algorithms for computing the defender optimal strategy against the new models.

Categories and Subject Descriptors

H.4 [**Computing Methodology**]: Game Theory

General Terms

Security, Algorithm

Keywords

Bounded Rationality, Network Stackelberg Games, Decision-making, Quantal Response

1. INTRODUCTION

Stackelberg Security Games (SSGs) have received great attention recently in solving real-world security problems, in which security forces (the leader) must allocate resources to protect one or more potential targets from being damaged by the attackers (the followers). Since the attackers can usually observe the defender's strategy before deciding on a plan of attack, the defender commits to a randomized strategy before the attacker chooses a strategy. Such attacker-defender Stackelberg game models have been used as the basis of many real-world deployed systems, including ARMOR, IRIS and GUARDS [6].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

In this paper we focus on security games whose domains have structure that is naturally modeled as graphs. For example, in response to the devastating terrorist attacks in 2008 [2], Mumbai police deployed randomized checkpoints as one countermeasure to prevent future attacks [7]. This can be modeled as a Stackelberg game on a graph with intersections as nodes and roads as edges, where certain nodes are targets for attacks. The attacker chooses a path on the graph ending at one of the targets. The defender can schedule checkpoints on edges to try to catch the attacker before a target is reached. Previous studies [7, 8] model these games as Network Security Games.

A common assumption of these previous studies is that the attacker is perfectly rational (i.e. chooses a strategy that maximizes their expected utility). However, extensive experimental studies have shown that standard game-theoretic assumptions of perfect rationality are not ideal for predicting the behavior of humans in multi-agent decision problems, and various alternative models have been proposed [1, 5]. Recently, Yang *et al* [9] studied human behavior models of attackers in the setting of (non-networked) Stackelberg security games. They showed that defender strategies based on a quantal response model (an adaptation of Quantal Response Equilibrium (QRE) concept [5] to the Stackelberg setting) achieved promising performance when tested against human subjects, outperforming previous methods for security games as well as a behavior model based on Prospect Theory [4].

In this work, we initiate the study of human behavior models of adversaries in network security games. Compared to the non-networked domains, the network structure of this domain further complicates the decision process of the human adversaries, hence further motivating the need to relax assumptions of perfect rationality. Specifically, the attacker must choose a path in the graph where each edge is covered by the defender with some observed probability, and thus must reason about sequences of random events. Our goal is to explore any bias and/or heuristic behavior exhibited by human adversaries when facing such decision problems, and to design defender strategies that exploit such behavior. While it is generally accepted that humans tend to rely on heuristics when faced with complex problems (e.g., [3]), to the best of our knowledge, there are no existing studies that specifically addressed heuristic human behavior in the security domain.

We propose two behavior models for attackers in network security games. First, we adapted the quantal response model [9] to network security games. For the second model (which we call quantal response with heuristics), the attacker's behavior now depends on the values of several easy-to-compute *features* of the attacker's decision problem. Furthermore, we developed a web-based game that simulates the decision tasks faced by the attacker. We recruited human subjects to play the game, in order to collect data for training the model as well as evaluate the defender strategies that are computed based on the trained model.

2. METHODOLOGY

We consider a network security game the same as what is defined in [7], except that we now allow general-sum payoff structures.

Adversary Models: We propose two models of the adversary. In the first model, the adversary's mixed strategy is a quantal response (QR) to the defender's strategy: the probability that the adversary chooses path A_i is

$$\texttt{QR}: \quad q_i(\lambda \mid \boldsymbol{x}; \Gamma) = \frac{e^{\lambda U_i^a(\boldsymbol{x};\Gamma)}}{\sum_{A_k \in \mathcal{A}} e^{\lambda U_k^a(\boldsymbol{x};\Gamma)}} \tag{1}$$

where Γ denotes a given game sample, $\boldsymbol{x}$ is the defender's strategy, U_i^a is the adversary's expected utility of choosing path A_i, and $\lambda > 0$ is the parameter of the quantal response model [5] which represents the error level of the adversary's quantal response. In the second model, which we call Quantal Response with Heuristics (QRH), the probability that the adversary chooses path A_i is

$$\texttt{QRH}: \quad q_i(\mu \mid \boldsymbol{x}; \Gamma) = \frac{e^{\mu \cdot f_i(\boldsymbol{x})}}{\sum_{A_k \in \mathcal{A}} e^{\mu \cdot f_k(\boldsymbol{x})}} \tag{2}$$

where $\mu = \langle \mu_1, ..., \mu_m \rangle$ is a vector of coefficients of the model and given $\boldsymbol{x}$, $f_i(\boldsymbol{x}) = \langle f_{i1}(\boldsymbol{x}), .., f_{im}(\boldsymbol{x}) \rangle$ is a vector of m *features* for path A_i that influences the attacker's decision making. Since our focus for the QRH model is on simple heuristics, we use a set of five features for each path that are easy to compute for humans and thus could be used as a basis for heuristics: 1. number of edges; 2. minimum coverage on a single edge; 3. maximum coverage on a single edge; 4. sum of edge coverage; 5. average of edge coverage.

Model Training: We developed a web-based game which simulates the decision tasks faced by the attacker in network security games. Figure 1 displays the interface of the game. Players are

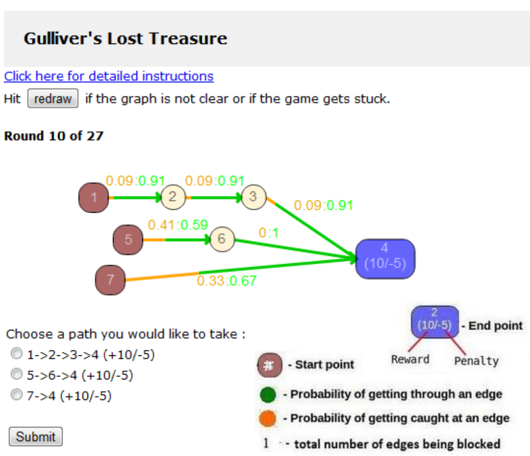


Figure 1: Game Interface (colored)

introduced to the game through a series of explanatory screens describing how the game is played. In the game, the web interface presents a graph to the subjects and specifies the source(starting) nodes and the target nodes in the graph. The subjects are asked to select a path from one of the source nodes to one of the target nodes. They are also told that the defender is trying to catch them by setting up checkpoints on the edges. The probability that there will be a check point on each edge is given to the subjects, as well as the reward for successfully getting through the path and the penalty for being caught by the defender. We posted the game as a Human Intelligent Task on Amazon Mechanical Turk (https://www.mturk.com) to collect data on how human subjects play the game and learned the parameters of both the QR model and the QRH model with the data using Maximum Likelihood Estimation.

Computing Defender Strategy: Given a QR/QRH model of the adversary, we have the following optimization problem to compute the corresponding defender's optimal strategy:

$$\max_{\boldsymbol{x},p} \sum_{A_i \in \mathcal{A}} q_i(\lambda \mid \boldsymbol{x}; \Gamma)((R_i^d - P_i^d)p_i + P_i^d) \tag{3}$$

$$\text{s.t.} \sum_{e \in E} x_e \leq M, \quad 0 \leq x_e \leq 1, \quad \forall e \in E \tag{4}$$

$$p_i = \sum_{e \in A_i} x_e, \quad \forall A_i \in \mathcal{A} \tag{5}$$

where $q_i(\lambda \mid \boldsymbol{x}; \Gamma)$ in Equation (3) is specified in Equation (1) and (2). The problem is a nonlinear and nonconcave. We use a heuristic algorithm based on local optimization with random restarts, similar to that used in [9] to solve the problem.

3. CONCLUSION

We presented an initial study of human behavior models of adversaries in network security games. In particular, we first proposed two behavior models, quantal response (QR) and quantal response with heuristics (QRH). In order to train our models and to evaluate their performances, we developed a web-based game that simulates the decision tasks faced by the attacker. We then trained the model with the data that we collected by posting the game on Amazon Mechanical Turk. Finally, we provided new algorithms to compute the defender optimal strategy against the new models.

4. ACKNOWLEDGMENTS

This research was supported by Army Research Office under the grant # W911NF-10-1-0185.

5. REFERENCES

[1] C. F. Camerer, T. Ho, and J. Chong. A cognitive hierarchy model of games. *QJE*, 119(3):861–898, 2004.

[2] R. Chandran and G. Beitchman. Battle for Mumbai ends, death toll rises to 195. *Times of India*, November 2008.

[3] G. Gigerenzer, P. Todd, and the ABC Research Group. *Simple Heuristics that make us smart*. Oxford University Press, 1999.

[4] D. Kahneman and A. Tvesky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

[5] R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 2:6–38, 1995.

[6] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, New York, NY, 2011.

[7] J. Tsai, Z. Yin, J. Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. Urban security: Game-theoretic resource allocation in networked physical domains. *In AAAI*, 2010.

[8] A. Washburn and K. Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251, 1995.

[9] R. Yang, C. Kiekintveld, F. Ordonez, M. Tambe, and R. John. Improving resource allocation strategy against human adversaries in security games. *In IJCAI*, 2011.

Anytime Algorithms for Multi-agent Visibility-based Pursuit-evasion Games

(Extended Abstract)

Viliam Lisý, Branislav Bošanský, Michal Pěchouček
Agent Technology Center, Dept. of Computer Science and Engineering, FEE, Czech Technical University
Technická 2, 16627 Prague 6, Czech Republic
{lisy, bosansky, pechoucek}@agents.felk.cvut.cz

ABSTRACT

We investigate algorithms for playing multi-agent visibility-based pursuit-evasion games. A team of pursuers attempts to maintain visibility contact with an evader who actively avoids tracking. We aim for applicability of the algorithms in real-world scenarios; hence, we impose hard constraints on the run-time of the algorithms and we evaluate them in a simulation model based on a real-world urban area. We compare Monte-Carlo tree search (MCTS) and iterative deepening minimax algorithms running on the information-set tree of the imperfect-information game. The experimental results demonstrate that both methods create comparable good strategies for the pursuer, while the later performs better in creating the evader's strategy.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Experimentation

Keywords

Pursuit-evasion game, Monte-Carlo tree search, Information-set search, Anytime algorithm

1. PROBLEM DEFINITION

The problem of visibility tracking is of particular interest for defense or security domains in which the target actively avoids being seen by the tracking agents. Game theory provides theoretic and algorithmic foundations for such situations and a game modeling these scenarios is defined as a *visibility-based pursuit-evasion game* with simultaneous moves — a two-player zero-sum extensive-form game between the *pursuer* (that controls multiple pursuing agents) and the *evader*. We focus on variants of these games played in a Euclidean environment discretized as a graph. We assume that both players have a full knowledge about the topology of the environment, but do not know the position of the opponent's agents unless one of their agents can see them.

We adopt the definition of the visibility-based pursuit-evasion game from [3], and we assume a single evading agents and multiple centrally-controlled pursuing agents. The main objective of the pursuers is to minimize the *mean size* of the set of possible positions of the evader based on the shared information of the pursuer's agents (we denote this measure *MS*). The objective of the evader is exactly the opposite; however, the evader needs to approximate this value. The exact value depends on trajectories of the pursuers which may be unknown to the evader. Besides the mean size objective we evaluate two other performance measures. The first is the number of times the evader has been spotted by a pursuing agent (denoted *NS*). The second is the size of the set of possible positions of the evader at the end of the game (denoted *ES*), which is the objective used in [3].

2. ANYTIME ALGORITHMS

Both evaluated algorithms search in the same search space. It is the information set tree[3], where plies of agent's decisions are interleaved with plies of possible observations.

Iterative deepening minimax.

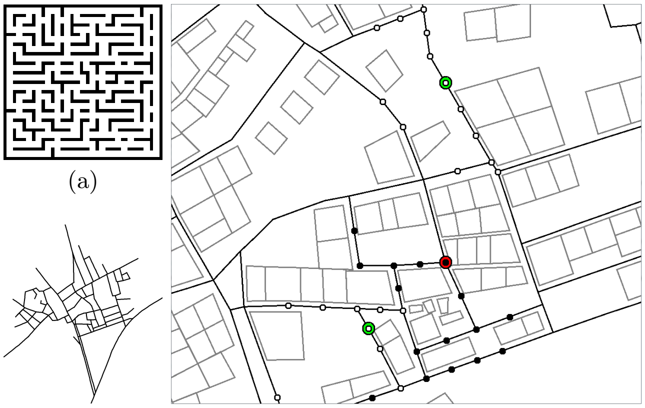
The first algorithm (denoted MM) we use is based on the state of the art technique presented in [3]. It is a depth-limited minimax search with a heuristic evaluation function and the paranoid opponent model. The empirical distribution of computation times of this method with fixed look-ahead depth has a very long tail. In order to meet the anytime requirement, we use iterative deepening and alpha-beta pruning.

Monte-Carlo tree search.

The second algorithm (denoted MC) is MCTS with UCT [2] selection on the same information set tree as in the MM case. The performance of the algorithms was not significantly influenced by the choice of the UCT parameter hence we set it to two in the experiments. We run expansion in each iteration of the algorithm and we select the first child generated for simulation without preference ordering.

We have evaluated several simulation strategies with varying amounts of domain-specific knowledge and cut-off depths. However, consistently with [1], we found that shorter simulations perform better. We achieved the best results when using evaluation functions instead of simulation and back-propagating the returned value in the MCTS tree.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.



(a) (b) (c)

Figure 1: The environment maps used for experimental evaluation. (a) full maze map used in [3]; (b) a detail of the road-network map with agents visualized as the larger circles, the current set of possible positions of the evader as the black circles, and the positions visible to the pursuer as the white circles; (c) the complete road-network map.

Evaluation functions.

The experimental evaluation in [3] identifies the relaxed lookahead heuristic (RLA_p) as the most successful for the pursuer. RLA_p computes the mean number of positions where the evader can be present after d steps of the game ($d = 10$ in our experiments) and cannot be spotted under any movement of the pursuers. The authors, however, do not define any heuristics for the evader. They assume the worst case behavior of the evader that knows the position of the pursuers all the time in their experiments (E. Raboin 2011, pers. comm. 2 February). In this paper, we aim to achieve realistic behavior of the evader as well. Hence we define RLA_e as the same heuristic computed from the perspective of the evader, i.e., with certain evader's position and uncertain pursuer's positions. We also use a modified version of the evaluation function computed as a sum of the objective value MS and RLA. For the case of evader, MS is the mean of sum of possible positions set sizes of the evaders.

If the set of possible positions of a pursuer is too large (e.g., all the currently unseen positions), it renders all the strategies of the evader almost equally bad. The (paranoid) evader always expects the pursuer to appear just in front of it. Therefore in our implementations, the evader ignores actions of any pursuer that can possibly be at more than a certain number of positions (250 in our scenarios).

3. EXPERIMENTAL EVALUATION

In the experiments, two agents of the pursuer are tracking one evader. The implementation of each player uses only one thread and its computation time is limited to one second on Intel(R) i7 CPU @ 2.80GHz. Each scenario runs for 100 time steps and the results are mean of 100 runs. For initial positions of the game, we follow [3]. We use randomized settings with the evader visible to at least one of the pursuing agents, but far enough form the pursuers to make the tracking difficult.

We use two maps in the experiments. The first is the map from [3] for a fair comparison with the state-of-the-art algorithm. The topology of the map in form of 50x49 pixels bitmap is presented in Figure 1a. White pixels represent possible position of the agents, black pixels are obstacles and agent can move to the up to four adjacent pixels in one time step. Line-of-sight visibility with Euclidean distance limitation of 10 pixels is assumed.

The second map is based on the topology of a small real-world urban area. Figure 1b presents the overview of the complete road network and Figure 1c is a detail from the center of the map. The road network was discretized as a graph with a node placed every 25 meters, creating 465 nodes. We assume symmetric visibility and the agents can see each other if they are not further than 200 meters form each other and there is no building in their line of sight. An anytime solution is clearly needed with this map. The information set search with fixed lookahead of 8 finishes in less than one second in more than 50% of positions from our experiments, but still takes more than 10 seconds in approximately 3% of cases.

pursuers → / evader ↓	MM(MS+RLA) NS↑	MS↓	ES↓	MC(RLA) NS↑	MS↓	ES↓
Route-network Map						
MM(MS+RLA)	56.5	89.1	146.3	58.1	88.2	132.8
MC(MS)	63.0	70.6	107.6	71.5	39.6	52.0
Maze Map						
MM(MS+RLA)	58.5	60.3	111.0	56.3	67.0	120.0
MC(MS)	80.4	11.0	17.9	80.7	11.3	17.5

Figure 2: The best Monte-Carlo tree search and iterative deepening minimax approaches. The pursuer maximizes and the evader minimizes the measures marked by ↑.

The results in Figure 2 demonstrate that both iterative deepening minimax and MCTS can be used to create good anytime algorithms for the pursuer. Each of them slightly outperforms the other on one of the domains. This is not true for the evader. The minimax-based player is much stronger on the evader's side in both domains. The main difference between the two players in the game is in the amount of uncertainty about the world state and in the branching factor. The decision nodes of the evader represent moves of one player and the decision nodes of the evader represent joint moves of two agents. Furthermore, the number of new nodes that can be observed after a move is also larger for the pursuer. This indicates that, as in perfect information games, minimax-based approaches perform better on games with smaller branching factors and MCTS on games with larger branching factors.

4. ACKNOWLEDGEMENTS

The work presented is supported by the U.S. Army grant no. W15P7T-05-R-P209 (contract BAA 8020902.A), the Grant Agency of the Czech Technical University in Prague, grant No. SGS10/188/OHK3/2T/13 and the Czech Science Foundation, grant No. P202/12/2054.

5. REFERENCES

[1] P. Ciancarini and G. P. Favini. Monte Carlo tree search in Kriegspiel. *Artif. Intell.*, 174:670–684, July 2010.
[2] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *ECML*, pages 282–293, 2006.
[3] E. Raboin, D. Nau, U. Kuter, and S. Gupta. Strategy generation in multi-agent imperfect-information pursuit games. In *AAMAS*, pages 947–954, 2010.

Computing Optimal Security Strategies in Networked Domains: A Cost-Benefit Approach

(Extended Abstract)

Joshua Letchford
Duke University
Durham, NC
jcl@cs.duke.edu

Yevgeniy Vorobeychik*
Sandia National Laboratories
Livermore, CA
yvorobe@sandia.gov

ABSTRACT

We introduce a novel framework for computing optimal randomized security policies in networked domains which extends previous approaches in several ways. First, we extend previous linear programming techniques for Stackelberg security games to incorporate benefits and costs of arbitrary security configurations on individual assets. Second, we offer a principled model of failure cascades that allows us to capture both the direct and indirect value of assets, and extend this model to capture uncertainty about the structure of the interdependency network. Third, we extend the linear programming formulation to account for exogenous (random) failures in addition to targeted attacks. Fourth, we allow the attacker to choose among several capabilities in attacking a target, and, in a limited way, allow the attacker to attack multiple targets simultaneously. The goal of our work is two-fold. First, we offer techniques to compute optimal security strategies in realistic settings involving interdependent security. Second, our computational framework enables us to attain theoretical insights about security on networks.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed artificial intelligence—*Intelligent agents*

General Terms

Algorithms, Performance, Economics, Security

Keywords

Game theory, Security, Stackelberg Games, Networks

*Sandia National Labs, Livermore, CA. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1. INTRODUCTION

Game theoretic approaches to security have received much attention in recent years. There have been numerous attempts to distill various aspects of the problem into a model that could be solved in closed form, particularly accounting for interdependencies of security decisions (e.g., [5, 2]). Numerous others offer techniques based on mathematical programming to solve actual instances of security problems. One important such class of problems is network interdiction [1], which models zero-sum encounters between an interdictor, who attempts to destroy a portion of a network, and a smuggler, whose goal typically involves some variant of a network flow problem (for example, maximizing flow or computing a shortest path).

Our point of departure is another class of optimization-based approaches in security settings: Stackelberg security games [6]. These are two-player games in which a *defender* aims to protect a set of targets using a fixed set of limited defense resources, while the attacker aims to assail a target that maximizes his expected utility. A central assumption in the literature on Stackelberg security games is that the defender can commit to a probabilistic defense (equivalently, the attacker observes the probabilities with which each target is covered by the defender, but not the actual defense configuration).

Much of the work on Stackelberg security games focuses on building fast, scalable algorithms, often in restricted settings [4, 3]. One important such restriction is to assume that targets exhibit *independence*: that is, the defender's utility only depends on which target is attacked and the security configuration at that target. Short of that restriction, one must, in principle, consider all possible combinations of security decisions jointly for all targets, making scalable computation elusive. Many important settings, however, exhibit interdependencies between potential targets of attack. These may be explicit, as in IT and supply chain network security, or implicit, as in defending critical infrastructure (where, for example, successful delivery of transportation services depends on a highly functional energy sector, and vice versa), or in securing complex software systems (with failures at some modules having potential to adversely affect other modules). While in such settings the assumption of independence seems superficially violated, we demonstrate below that under realistic assumptions about the nature of interdependencies, we can nevertheless leverage the highly scalable optimization techniques which assume independence.

2. STACKELBERG SECURITY GAMES

A Stackelberg security game consists of two players, the leader (defender) and the follower (attacker), and a set of possible targets. The leader can decide upon a randomized policy of defending the targets, possibly with limited defense resources. The follower (attacker) is assumed to observe the randomized policy of the leader, but not the realized defense actions. Upon observing the leader's strategy, the follower chooses a target so as to maximize its expected utility.

In past work, Stackelberg security game formulations focused on defense policies that were costless, but resource bounded. Specifically, it had been assumed that the defender has K fixed resources available with which to cover targets. Additionally, security decisions amounted to covering a set of targets, or not. While in numerous settings to which such work has been applied (e.g., airport security, federal air marshall scheduling) this formulation is very reasonable, in other settings one may choose among many *security configurations* for each valued asset, and, additionally, security resources are only available at some cost. For example, in cybersecurity, protecting computing nodes could involve configuring anti-virus and/or firewall settings, with stronger settings carrying a benefit of better protection, but at a cost of added inconvenience, lost productivity, as well as possible licensing costs. Indeed, costs on resources may usefully take place of resource constraints, since such constraints are often not hard, but rather channel an implicit cost of adding further resources.

3. A GENERAL MODEL OF INTERDEPENDENCIES

Thus far, a key assumption has been that the utility of the defender and the attacker for each target depends only on the defense configuration for that target, as well as whether it is attacked or not. In many domains, such as cybersecurity and supply chain security, assets are fundamentally interdependent, with an attack on one target having potential consequences for others. In this section, we show how to transform certain important classes of problems with interdependent assets into a formulation in which targets become effectively independent, for the purposes of our solution techniques.

Below we focus on the defender's utilities; attacker is treated identically. Let w_t be an *intrinsic worth* of a target to the defender, that is, how much loss the defender would suffer if this target were to be compromised with no other target affected (i.e., not accounting for indirect effects). In doing so, we assume that these worths are independent for different targets. Let $s = \{o_1, \ldots, o_n\}$ be the security configuration on all nodes. Assuming that the utility function is additive in target-specific worths and the attacker can only attack a single target, we can write it as

$$U_t(s) = E\left[\sum_{t'} w_{t'} 1(\text{t}' \text{ affected} \mid \text{s}, \text{t})\right] = \sum_{t'} w_{t'} z_{s,t'}(t),$$

where $1(\cdot)$ is an indicator function and $z_{s,t'}(t)$ is the marginal probability that target t' is affected when the attacker attacks target t. From this expression, it is apparent that in general, $U_t(s)$ depends on defense configurations at all targets, creating an intractable large space of configurations over which the defender has to reason. We now make the crucial assumption that enables fast computation of defender policies by recovering inter-target independence.

ASSUMPTION 1. *For all t and t', $z_{s,t'}(t) = z_{o_t,t'}(t)$.*

In words, the probability that a target t' is affected when t is attacked only depends on the security configuration at the attacked target t. Below, we use a shorthand o instead of o_t where t is clear from context.

A way to interpret our assumption is that once some target is compromised, the fault may spread to other assets in spite of good protection policies. This assumption was operational in other work on interdependent security [5], where a justification is through a story about airline baggage screening: baggage that is transferred between airlines is rarely thoroughly screened, perhaps due to the expense. Thus, even while an airline may have very strong screening policies, it is poorly protected from luggage entering its planes via transfers. Cybersecurity has similar shortcomings: defense is often focused on external threats, with little attention paid to threats coming from computers internal to the network. Thus, once a computer on a network is compromised, the attacker may find it much easier to compromise others on the same network. The problem is exacerbated by the use of common operating environments, since once an exploit is found, it can often be reused to compromise other computing resources on a common network.

Under the above assumption, we can write the defender utility when t is attacked under security configuration o as,

$$U_{o,t} = z_{o,t}(t) w_t + \sum_{t' \neq t} z_{o,t'}(t) w_{t'}.$$

By a similar argument and an analogous assumption for the attacker's utility, we thereby recover target independence required by the Stackelberg linear programming formulations.

4. REFERENCES

[1] Kelly J. Cormican, David P. Morton, and R. Kevin Wood. Stochastic network interdiction. *Operations Research*, 46(2):184–197, 1998.

[2] Jens Grossklags, Nicolas Christin, and John Chuang. Secure or insure? A game-theoretic analysis of information security games. In *Seventeenth International World Wide Web Conference*, pages 209–218, 2008.

[3] Manish Jain, Erim Kardes, Christopher Kiekintveld, Milind Tambe, and Fernando Ordonez. Security games with arbitrary schedules: A branch and price approach. In *Twenty-Fourth National Conference on Artificial Intelligence*, 2010.

[4] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *In AAMAS-09*, 2009.

[5] Howard Kunreuther and Geoffrey Heal. Interdependent security. *Journal of Risk and Uncertainty*, 26(2-3):231–249, 2003.

[6] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *Seventh International Conference on Autonomous Agents and Multiagent Systems*, pages 895–902, 2008.

Automated Equilibrium Analysis of Repeated Games with Private Monitoring: A POMDP Approach

(Extended Abstract)

YongJoon Joe[1], Atsushi Iwasaki[1], Michihiro Kandori[2], Ichiro Obara[3], and Makoto Yokoo[1]
1: Kyushu University, Japan, {yongjoon@agent., iwasaki@, yokoo@}inf.kyushu-u.ac.jp
2: University of Tokyo, Japan, kandori@e.u-tokyo.ac.jp,
3: UCLA, California, ichiro.obara@gmail.com

ABSTRACT

The present paper investigates repeated games with *imperfect private monitoring*, where each player privately receives a noisy observation (signal) of the opponent's action. Such games have been paid considerable attention in the AI and economics literature. Identifying pure strategy equilibria in this class has been known as a hard open problem. Recently, we showed that the theory of partially observable Markov decision processes (POMDP) can be applied to identify a class of equilibria where the equilibrium behavior can be described by a finite state automaton (FSA). However, they did not provide a practical method or a program to apply their general idea to actual problems. We first develop a program that acts as a wrapper of a standard POMDP solver, which takes a description of a repeated game with private monitoring and an FSA as inputs, and automatically checks whether the FSA constitutes a symmetric equilibrium. We apply our program to repeated Prisoner's dilemma and find a novel class of FSA, which we call k-period mutual punishment (k-MP). The k-MP starts with cooperation and defects after observing a defection. It restores cooperation after observing defections k-times in a row. Our program enables us to exhaustively search for all FSAs with at most three states, and we found that 2-MP beats all the other pure strategy equilibria with at most three states for some range of parameter values and it is more efficient in an equilibrium than the grim-trigger.

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

Game theory, repeated games, private monitoring, POMDP

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1. INTRODUCTION

We consider repeated games with *imperfect private monitoring*, where each player privately receives a noisy observation (signal) of the opponent's action. This class of games represents long-term relationships among players and has a wide range of applications, e.g., secret price cutting and agent planning under uncertainty. Therefore, it has been paid considerable attention in the AI and economics literature. In particular, for the AI community, the framework has become increasingly important for handling noisy environments. In fact, Tennenholtz and Zohar consider repeated congestion games where an agent has limited capability in monitoring the actions of her counterparts [5].

Analytical studies on this class of games have not been quite successful. The difficulty comes from the fact that players do not share common information under private monitoring, and finding pure strategy equilibria in such games has been known as a hard open problem [4]. Under private monitoring, each player cannot observe the opponents' private signals, and he or she has to draw statistical inferences about the history of the opponents' private signals. The inferences quickly become very complicated over time, even if players adopt relatively simple strategies [1]. As a result, finding a profile of strategies which are mutual best replies after any history, i.e., finding an equilibrium, is a quite demanding task.

Quite recently, we show that the theory of the partially observable Markov decision process (POMDP) can be used to identify equilibria, when equilibrium behavior is described by a finite state automaton (FSA) [2]. This result is significant since it implies that by utilizing a POMDP solver, we can systematically determine whether a given profile of finite state automata can constitute an equilibrium. Furthermore, this result is interesting since it connects two popular areas in AI and multi-agent systems, namely, POMDP and game theory.

We first develop a program that acts as a wrapper of a standard POMDP solver. Furthermore, as a case study to confirm the usability of this program, we identify equilibria in an infinitely repeated prisoner's dilemma game, where each player privately receives a noisy signal about each other's actions.

2. REPEATED GAMES WITH PRIVATE MONITORING AND FSA

A finite state automaton (FSA) is a popular approach for compactly representing the behavior of a player in repeated games. We focus on a *symmetric pure finite state equilibrium* (SPFSE), which is a pure strategy sequential equilibrium of a repeated game with private monitoring, where each player's behavior on the equilibrium path is given by an FSA. A sequential equilibrium is a refinement of Nash equilibrium for dynamic games of imperfect information.

We apply the POMDP technique to the prisoner's dilemma model analyzed by [2]. The stage game payoff is given as follows.

	$a_2 = C$	$a_2 = D$
$a_1 = C$	$1, 1$	$-y, 1 + x$
$a_1 = D$	$1 + x, -y$	$0, 0$

Each player's private signal is $\omega_i \in \{g, b\}$ (*good* or *bad*), which is a noisy observation of the opponent's action. For example, when the opponent chooses C, player i is more likely to receive the correct signal $\omega_i = g$, but sometimes an observation error provides a wrong signal $\omega_i = b$. Let us introduce the joint distribution of private signals $o(\boldsymbol{\omega} \mid \boldsymbol{a})$ for the prisoner's dilemma model. When the action profile is (C, C), the joint distribution is given as follows (when the action profile is (D, D), p and s are exchanged).

	$w_2 = g$	$w_2 = b$
$w_1 = g$	p	q
$w_1 = b$	r	s

Similarly, when the action profile is (C, D), the joint distribution of private signals is given as follows (when the action profile is (D, C), v and u are exchanged).

	$w_2 = g$	$w_2 = b$
$w_1 = g$	t	u
$w_1 = b$	v	w

These joint distributions of private signals require only the constraints of $p + q + r + s = 1$ and $t + u + v + w = 1$.

We define a monitoring structure that is *nearly-perfect*. We say monitoring is nearly-perfect if each player is always likely to perfectly observe the opponent's action in each period, i.e., $p = v$, $q = r = t = w$, and $s = u = 1 - p - 2q$, where p is much larger than q or s. Although the monitoring structure is quite natural, systematically finding equilibria in such structure has not been possible without utilizing a POMDP solver. Alternatively, we say monitoring is *almost-public* if players are always likely to get the *same* signal (after (C, D), for example, players are likely to get (g, g) or (b, b)), i.e., $p + s = t + w \approx 1$ and $q = r = u = v \approx 0$.

Let us summarize the existing FSAs. First, grim-trigger (GT) is a well-known FSA under which a player first cooperates, but as soon as she observes defection, she defects forever. GT can often constitute an equilibrium. Second, tit-for-tat (TFT) is another well-known FSA in Fig. 1. It is well known that TFT does not prescribe mutual best replies after a deviation (hence it is *not* a subgame perfect Nash equilibrium (SPNE)). This problem does not arise under almost-public monitoring. Finally, 1-period mutual punishment (1-MP) in Fig. 2 is known as *Pavlov* [3]. According to this FSA, a player first cooperates. If her opponent defects, she also defects, but after one period of mutual defection, she returns to cooperation. It is well-known that Pavlov

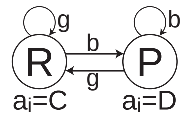


Figure 1: TFT

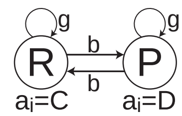


Figure 2: 1-MP

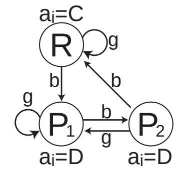


Figure 3: 2-MP

can constitute an SPNE under perfect monitoring. However, this has not been investigated well in the setting of private monitoring.

3. *K*-PERIOD MUTUAL PUNISHMENT

Let us first consider 1-MP. We can see that after one observation error occurs, players can quickly return to the mutual cooperation state *RR*. The expected probability (in the invariant distribution) that players are in state *RR* is about $p - 2q$. Unfortunately, 1-MP does not constitute an SPFSE in our parameterization, since it is too forgiving.

Therefore we generalize the idea of 1-MP to k-period mutual punishment (k-MP). Under this FSA, a player first cooperates. If her opponent defects, she also defects, but after k consecutive periods of mutual defection, she returns to cooperation. Figure 3 shows the FSAs of 2-MP. 2-MP is less forgiving than 1-MP, since it cooperates approximately once in every three periods to the opponent who always defects. By increasing k, we can make this strategy less forgiving. When $k = \infty$, this strategy becomes equivalent to GT.

Although it is somewhat counter-intuitive, requiring such mutual defection periods is beneficial in establishing a robust coordination among players under nearly-perfect monitoring. In contrast, under almost-public monitoring, TFT can better coordinate players' behavior; TFT can be an equilibrium, while k-MP is not. In both cases, GT can be an equilibrium. Accordingly, our program helps us to gain important insights into the way players coordinate their behavior under different private monitoring structures.

Furthermore, we exhaustively search for small-sized FSAs that can constitute an equilibrium under nearly-perfect monitoring. We enumerate all possible FSAs with at most three states, i.e., 5832 FSAs, which is obtained from the numbers of actions, signals, and states, and check whether they constitute an SPFSE. We found that only eleven FSAs (after removing equivalent ones) could be an SPFSE in a reasonably wide range of signal parameters. In addition, among them, 2-MP is the only FSA that is more efficient than GT.

4. REFERENCES

[1] M. Kandori. Repeated games. *Game theory*. pages 286–299. Palgrave macmillan, 2010.

[2] M. Kandori and I. Obara. Towards a Belief-Based Theory of Repeated Games with Private Monitoring: An Application of POMDP. mimeo, 2010.

[3] D. Kraines and V. Kraines. Pavlov and the prisoner's dilemma. *Theory and Decision*, 26:47–79, 1989.

[4] G. Mailath and L. Samuelson. *Repeated Games and Reputation*. Oxford University Press, 2006.

[5] M. Tennenholtz and A. Zohar. Learning equilibria in repeated congestion games. In *AAMAS*, pages 233–240, 2009.

Adversarial Patrolling Games

(Extended Abstract)

Yevgeniy Vorobeychik*
Sandia National Laboratories
Livermore, CA
yvorobe@sandia.gov

Bo An and Milind Tambe
University of Southern California
Los Angeles, CA
{boa,tambe}@usc.edu

ABSTRACT

Defender-Attacker Stackelberg games are the foundations of tools deployed for computing optimal patrolling strategies in adversarial domains such as the United states Federal Air Marshals Service and the United States Coast Guard, among others. In Stackelberg game models of these systems the attacker knows only the probability that each target is covered by the defender, but is oblivious to the detailed timing of the coverage schedule. In many real-world situations, however, the attacker can observe the current location of the defender and can exploit this knowledge to reason about the defender's future moves. We study Stackelberg security games in which the defender sequentially moves between targets, with moves constrained by an exogenously specified graph, while the attacker can observe the defender's current location and his (stochastic) policy concerning future moves.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed artificial intelligence—*Intelligent agents*

General Terms

Algorithms, Performance, Economics, Security

Keywords

Game theory, Security, Stackelberg Games, Patrolling, MDP

1. INTRODUCTION

Game theoretic approaches to security based on Stackelberg game models have received much attention in recent years, with several finding deployment in real-world settings including LAX (Los Angeles International Airport), FAMS (United States Federal Air Marshals Service), TSA (United States Transportation Security Agency), and USCG (United States Coast Guard) [8, 3]. At the backbone of these applications are defender-attacker Stackelberg games in which the defender first commits to a randomized security policy, and the attacker uses surveillance to learn about the policy before attacking. The analysis of Stackelberg security games has focused primarily on computing Strong Stackelberg equilibrium (SSE), i.e., the optimal strategy for the defender [7, 9].

To date, the Stackelberg game models for all real-world security applications assume that attacker knows the probability that each target is covered by the defender, but is oblivious to the actual sequence of defender moves. For example, the defender may in fact visit targets according to some fixed (but randomly generated) patrolling schedule, but the attacker is presumed to be unable to observe the defender's location at any point during the patrol. In many realistic settings, such as USCG [3], it is likely that the attacker can in fact observe the patrol while it is in progress (e.g., the coast guard ships can be quite overt). Thus, one potentially more plausible model in such a setting would allow the attacker to observe both the randomized policy of the defender (i.e., probability distribution over moves) as well as current defender location. We formally model this setting as an *adversarial patrolling game*, or APG.

2. RELATED WORK

Some of the earliest work on adversarial patrolling settings was done in the context of robotic patrols, but involved a comparatively simpler defense decision space (for example, with a set of robots moving around a perimeter, and a single parameter governing the probability that they move forward or back) [1, 2].

More recent work by Basilico *et al.* [5, 4, 6] studied general-sum patrolling games in which they assumed that the attacker is infinitely patient, but the execution of an attack can take an arbitrary number of time steps. However, the resulting formulations rely fundamentally on the assumption that both players are infinitely patient, and cannot be easily generalized to handle an impatient attacker. Moreover, Basilico *et al.* only consider a restricted attacker strategy space, and, additionally, their formulation may involve extraneous constraints which result in suboptimal solutions.

3. ADVERSARIAL PATROLLING

Formally, an *adversarial patrolling game (APG)* can be described by the tuple $\{T, U_d^c(i), U_d^u(i), U_a^c(i), U_a^u(i), \delta, G\}$, where T is the set of n targets patrolled by the defender, $U_d^c(i)$ and $U_d^u(i)$ are the utilities to the defender if an attacker chooses a target $i \in T$ when it is patrolled and not, respectively, while $U_a^c(i)$ and $U_a^u(i)$ are the corresponding attacker utilities, $\delta \in (0,1)$ is the discount factor (in some cases, we also allow $\delta = 1$), and $G = (T, E)$ is a graph with targets as vertices and E the set of directed edges constraining defender patrolling moves between targets. It is useful to consider the representation of this graph as an adjacency matrix A, where

*Sandia National Labs, Livermore, CA. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

$A_{ij} = 1$ if and only if there is an edge from target i to target j. Below we consider a zero-sum game setting, where $U_d^c(i) = -U_a^c(i)$ and $U_d^u(i) = -U_a^u(i)$.

The game proceeds in a (possibly infinite) sequence of steps in which the defender moves between targets (subject to the constraints imposed by G), while the attacker chooses the time and target of attack. The defender's (stochastic) patrolling policy is a schedule π which can in general be an arbitrary function from all observed history (i.e., the sequence of targets patrolled in the past) to a probability distribution over the targets patrolled in the next iteration. The attacker is presumed to know the defender's policy π at the time of decision. At each time step t the attacker observes the defender's current location i and may choose to wait or to attack an arbitrary target $j \in T$. If an attacker waits, he receives no immediate utility, while attacking a target j gains the attacker $U_a^c(i)$ if it is covered by the defender at time $t+1$ and $U_a^u(i)$ if it is not. We denote the attacker's policy by a. We say that a policy (π or a) is *Markovian* if it only depends on the current location of the defender, and we call it *stationary Markovian* if it additionally has no dependence on time.

EXAMPLE 1. ***USCG's Patrolling Problem as an APG****: USCG safeguards important infrastructure at US coasts, ports, and inland waterway. Given a particular port and a variety of critical infrastructure that an adversary may choose to attack, USCG conducts patrols to detect an adversary and protect this infrastructure. However, while the adversary has the opportunity to observe patrol patterns, limited security resources imply that USCG patrols cannot be at every location at all times [3]. In the APG framework, USCG is the defender, while a terrorist group (for example) is an attacker who can conduct surveillance and can both observe the current location of patrols and obtain a good estimate of the stochastic patrolling policy deployed.*

3.1 APG as a Stochastic Game

The adversarial patrolling game can be formulated as a *stochastic game*. A stochastic game is defined by a set of states, a set of players, each taking actions from a finite collection, transition probabilities between states which depend on joint player actions, and, finally, utility (reward) functions of players determined by current state and actions jointly selected by the players.

In our setting, states correspond to the set of targets T, as well as an absorbing state s. Defender actions in each state are the targets j that he can move to in a single time step, while attacker actions are to wait or to attack (for the moment, we will assume that we can compute expected utilities when attacker chooses to attack; we deal with the issue of which targets are attacked below). The state transitions are actually deterministic, conditional on player actions: if the attacker chooses to attack, the system always transitions to the absorbing state s; otherwise, the next target is completely determined by the defender's action. Finally, if the attacker waits, our baseline model involves zero reward accruing to both players. Letting R_i denote the expected utility to attacker of attacking in state i; the defender's utility in the zero-sum model is then $-R_i$. The stochastic game has an infinite horizon, and in our model the attacker's discount factor is δ. Figure 1 offers a schematic illustration of APG as a stochastic game. Since it's a zero-sum game, the defender will aim to minimizes the expected attacker utility (starting from state 0, as we had assumed).

Acknowledgements

This research is supported in part by MURI grant W911NF-11-1-0332.

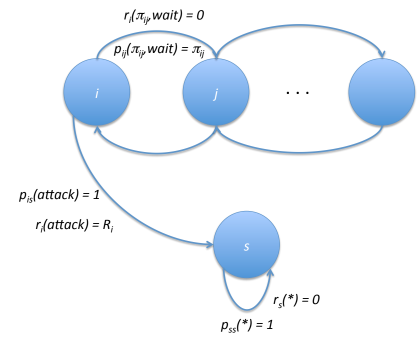


Figure 1: Schematic illustration of APG as a stochastic game, showing example targets-states i and j, as well the absorbing state s. $p_{ij}(\cdot)$ denotes the transition probability, as a function of the probability π_{ij} that the defender moves from i to j and whether or not the attacker chooses "wait" or "attack". Note that if the attacker attacks, the stochastic game transitions to the absorbing state with probability 1, independent of π_{ij}.

4. REFERENCES

[1] Noa Agmon, Sarit Krause, and Gal A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *IEEE International Conference on Robotics and Automation*, pages 2339–2345, 2008.

[2] Noa Agmon, Daniel Urieli, and Peter Stone. Multiagent patrol generalized to complex environmental conditions. In *Twenty-Fifth National Conference on Artificial Intelligence*, 2011.

[3] Bo An, James Pita, Eric Shieh, Milind Tambe, Christopher Kiekintveld, and Janusz Marecki. Guards and protect: Next generation applications of security games. In *SIGECOM*, volume 10, pages 31–34, March 2011.

[4] Nicola Basilico and Nicola Gatti. Automated abstraction for patrolling security games. In *Twenty-Fifth National Conference on Artificial Intelligence*, pages 1096–1099, 2011.

[5] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 57–64, 2009.

[6] Branislav Bosansky, Viliam Lisy, Michal Jakov, and Michal Pechoucek. Computing time-dependent policies for patrolling games with mobile targets. In *Tenth International Conference on Autonomous Agents and Multiagent Systems*, pages 989–996, 2011.

[7] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, EC '06, pages 82–90, New York, NY, USA, 2006. ACM.

[8] Manish Jain, Jason Tsai, James Pita, Christopher Kiekintveld, Shyamsunder Rathi, Milind Tambe, and Fernando Ordóñez. Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40:267–290, July 2010.

[9] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Seventh International Conference on Autonomous Agents and Multiagent Systems*, 2009.

Consensus Games

(Extended Abstract)

Julian Zappala Natasha Alechina Brian Logan

School of Computer Science
University of Nottingham
Nottingham, NG8 1BB, UK
{jxz,nza,bsl}@cs.nott.ac.uk

ABSTRACT

Consensus Games (CGs) are a novel approach to modelling coalition formation in multi-agent systems inspired by threshold models in sociology. In a CG, each agent's degree of commitment to the coalitions in which it may participate is expressed as a quorum function. Agents are willing to form a coalition only if a quorum consensus can be achieved amongst all agents of the coalition.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence Coherence and coordination

General Terms

Algorithms, Theory

Keywords

Multi-agent, Consensus, Threshold, Coalition Formation

1. INTRODUCTION

Coalition formation has traditionally been modelled using game theoretic techniques. Such models often necessitate strong economic assumptions, including that utility is transferable, and that coalitional valuations are known and can be fairly distributed. The multi-agent community in particular have investigated coalition formation in situations where these assumptions cannot easily be applied, for example, [1, 7]. A common assumption in this work is that all member-agents must somehow 'agree'; in other words, for a coalition to form it is necessary that there is a *consensus* among the members of the coalition regarding its formation.

In this extended abstract we propose *consensus games* (CGs), a novel model of consensual coalition formation for multi-agent systems inspired by threshold models in sociology. Threshold models have been used to describe a variety of social phenomena. For example, [3] presents a model in which n individuals face a binary decision, e.g., regarding whether to participate in a riot. Each individual has an idiosyncratic threshold representing the minimum proportion of others which must participate in order that the given individual will also participate. It is shown that the number of agents that ultimately decide to participate, e.g., number of agents that decide to riot, is critically dependent on the distribution of thresholds. Similar models have been used to investigate segregation in urban housing [6], and the adoption of consumer trends [4].

We extend the model proposed in [3] beyond binary choice decisions to the more general problem of coalition formation. For each coalition of which it may be a member, each agent has a threshold representing the proportion of agents that must support the formation of the coalition in order that the agent will also support the formation of the coalition. We focus on the special case of consensus: there is consensus about the formation of a particular coalition only where all agents support the formation of the coalition.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. CONSENSUS GAMES

Definition 1. *A consensus game (CG) is a tuple* $\Gamma = \langle G, q\rangle$ *where:*

G *is a finite set of agents,* $\{1, \dots, n\}, n \geq 2$.

q *is a quorum function.* $q : G \times 2^G \to [0, 1]$ *is a partial function which takes an agent* $i \in G$ *and a coalition* $H \subseteq G$ *where* $i \in H$*, and returns a number in the interval* $[0, 1]$.

For each coalition of which it may be a member, the value of the quorum function indicates the agent's 'degree of support' for the formation of that coalition. For an agent $i \in H \subseteq G$ the quorum function $q(i, H)$ gives the minimum proportion of agents in H that must support the formation of the coalition H in order that i will support the formation of H. Where $q(i, H) = 0$ agent i unconditionally supports the formation of the coalition H, where $0 < q(i, H) \leq \frac{|H|-1}{|H|}$ i conditionally supports the formation of the coalition H; where $\frac{|H|-1}{|H|} < q(i, H) \leq 1$ i does not support the formation of the coalition H. We use the abbreviation $q\#(i, H)$ to denote the number of other agents in H that must support H in order for i to support H. Formally, $q\#(i, H)$ is the minimal natural number k such that $q(i, H) \leq k/|H|$. We will denote by $n_k(H)$ the number of agents $i \in H$ with $q\#(i, H) = k$.

3. STRONG CONSENSUS

A key solution concept for CGs is the strong consensus coalition. A *strong consensus coalition* H is a coalition where for each agent $i \in H$ the quorum threshold $q(i, H)$ is satisfied in the sense that H contains at least $q\#$ other agents with strictly lower $q\#$ values.

Definition 2. *A coalition H is a strong consensus coalition if the following conditions hold:*

- $n_0(H) \neq 0$
- *if $n_k(H) \neq 0$, then $\Sigma_{j<k} n_j(H) \geq k$*

Note that the definition implies that if H is a strong consensus coalition, then $n_{|H|}(H) = 0$.

Consider the following example.

Example 1. *Alice (A) and Bob (B) are considering whether to get married. Bob, tired of bachelorhood, is keen to be married. Alice is not opposed to marrying Bob provided that Bob also wants to marry her, otherwise Alice will happily continue to be single. Alice's and Bob's positions can be formalised as the consensus game $\Gamma = \langle G, q\rangle$ where:*

$$G = \{A, B\}$$

$$q(i, H) = \begin{cases} 0 & \text{if } i = B \text{ and } H = \{A, B\} \\ 0.5 & \text{if } i = A \text{ and } H = \{A, B\} \\ 1 & \text{if } i = B \text{ and } H = \{B\} \\ 0 & \text{if } i = A \text{ and } H = \{A\} \end{cases}$$

In the example, Bob unconditionally supports the formation of the grand coalition (of all agents); Alice conditionally supports formation of this coalition provided that one other agent (Bob) also supports its formation. Alice also unconditionally supports formation of the singleton coalition {A}, whereas Bob does not support formation of the singleton coalition {B}. The grand coalition in this example is a strong consensus coalition.

We now show that there is an alternative definition of a strong consensus coalition as a fixed point of a function that intuitively corresponds to agents indicating their support for a coalition.

Consider the function $f_H : 2^G \to 2^G$ defined relative to $H \subseteq G$:

$$i \in f_H(Q) \text{ iff } i \in H \text{ and } |Q \cap H \setminus \{i\}| \geq q(i, H) \times |H|$$

This function takes as its input a set $Q \subseteq G$ and returns the set of agents in H whose quorum thresholds are satisfied by the membership of $Q \cap H$. If $Q = \emptyset$, f_H will contain only the agents i with $q(i, H) = 0$, if Q is the set of agents which have unconditional support for H, then $f_H(Q)$ will contain the agents i with $q\#(i, H) \leq |Q|$, and so on.

A coalition H is a strong consensus coalition if and only if it is the least fixed point of f_H. First we need the following auxiliary result:

Proposition 1. *The function f_H is guaranteed to possess at least one fixed point.*

We omit the proof due to lack of space.

The least fixed point of f_H can be established by recursive calls to the function starting with the empty set of agents as an argument. We refer to each invocation of f_H as a *round*. If H can achieve strong consensus, then it will be achieved in at most $|H|$ rounds.

We can now show that:

Theorem 1. *H is a strong consensus coalition if and only if it is the least fixed point of f_H.*

We omit the proof due to lack of space.

In characterising the computational complexity of CGs, a natural decision problem is given a CG $\Gamma = \langle G, q\rangle$ and a coalition $H \subseteq G$, can H reach strong consensus? Algorithm 1, which runs in time linear in the number of agents, can be used to determine if H is the least fixed point of f_H.

Algorithm 1 Can H reach strong consensus.

```
function SCC(q, H)
    array support[|H| + 1] ← {0, ..., 0}
    for all i ∈ H do
        k ← ⌈q(i, H) × |H|⌉
        support[k] ← support[k] + 1
    s ← support[0]
    for k from 1 to |H| do
        if k ≤ s then
            s ← s + support[k]
        else
            return false
    return true
```

4. DISCUSSION

The key idea of CGs is that agents' choices are conditioned by the number of other agents also making the same choice. This has some similarities with anonymous games [2], in which the individual utility of participation in a coalition can be dependant on factors including the size of the coalition, and with imitation games [5], in which an agent's behaviour may influence that of other agents.

CGs as presented here treat the problem of coalition formation in an abstract sense. It is often the case that coalition formation in multi-agent systems is directed toward the achievement of the agents' goals. It would therefore be interesting to extend the model of CGs to include representations of collective action and heterogeneous goals.

5. REFERENCES

[1] J. Abdou and H. Keiding. *Effectivity functions in social choice.* Kluwer Academic, 1991.

[2] C. Daskalakis and C. Papadimitriou. Computing equilibria in anonymous games. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 83–93. IEEE, 2007.

[3] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[4] M. Granovetter and R. Soong. Threshold models of interpersonal effects in consumer demand. *Journal of Economic Behavior & Organization*, 7(1):83–99, 1986.

[5] A. McLennan and R. Tourky. Simple complexity from imitation games. *Games and Economic Behavior*, 68(2):683 – 688, 2010.

[6] T. C. Schelling. Models of segregation. *The American Economic Review*, 59(2):488–493, 1969.

[7] M. Wooldridge and P. E. Dunne. On the computational complexity of qualitative coalitional games. *Artificial Intelligence*, 158(1):27–73, 2004.

Individual-based Stability in Hedonic Games depending on the Best or Worst Players

(Extended Abstract)

Haris Aziz Paul Harrenstein Evangelia Pyrga
Institut für Informatik, TU München
{aziz,harrenst,pyrga}@in.tum.de

ABSTRACT

We consider classes of hedonic games in which each player's preferences over coalition structures are induced by the best player ($\mathcal{B}$- and B-hedonic games) or the worst player ($\mathcal{W}$- and W-hedonic games) in his coalition. For these classes, which allow for concise representation, we analyze the computational complexity of deciding the existence of and computing individually stable, Nash stable, and individually rational and contractually individually stable coalition partitions. We identify a key source of intractability in compact coalition formation games in which preferences over players are extended to preferences over coalitions.

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

Economics, Theory and Algorithms

Keywords

Game theory (cooperative and non-cooperative)

1. HEDONIC GAMES

Coalition formation games, as introduced by Drèze and Greenberg [5], provide a simple but versatile formal model for modeling and analyzing how agents join in groups. In many situations it is natural to assume that a player's appreciation of a coalition structure only depends on the coalition he is a member of and not on how the remaining players are grouped. Much of the work on coalition formation concentrates on these so-called *hedonic games*.

Formally, a *hedonic game* is a pair $(N, \succsim)$, where N is a set of players and $\succsim = (\succsim_1, \dots, \succsim_{|N|})$ a profile specifying the preferences of each player i as a transitive and complete relation $\succsim_i$ over the set $\mathcal{N}_i = \{S \subseteq N \mid i \in S\}$ of coalitions i may belong to. If $\succsim_i$ is also anti-symmetric we say that i's preferences are *strict*. A coalition $S \in \mathcal{N}_i$ is *acceptable to i* if i prefers S to being alone, i.e., $S \succsim_i \{i\}$ and *unacceptable*, otherwise.

As the set of coalitions a player may be member of grows exponentially in the number of players, for hedonic games concise representations do not exist in general. However, concise representations are possible if we assume the players to have preferences over the *players* in N and that their appreciation of a coalition S systematically depends on their most or least preferred players in S. We distinguish four such classes of hedonic games: $\mathcal{B}$-hedonic games [2, 4], B-hedonic games, $\mathcal{W}$-hedonic games [3, 4], and W-hedonic games.

As no confusion is likely, we also use $\succsim_i$ to denote player i's preferences over N. For J a subset of players, we denote by $\max_i(J)$ and $\min_i(J)$ the sets of players that are *most*, respectively, *least* preferred by i in J, on the understanding that $\max_i(\emptyset) = \min_i(\emptyset) = \{i\}$. With a slight abuse of notation we write $\max_i(S) \succsim_i \max_i(T)$ ($\min_i(S) \succsim_i \min_i(T)$) if $s \succsim_i t$ for all $s \in \max_i(S)$ and all $t \in \max_i(T)$ ($s \in \min_i(S)$ and all $t \in \min_i(T)$, respectively). Moreover, player j is said to be *acceptable* to i if $j \succsim_i i$, and *unacceptable* otherwise.

In a *B-hedonic* game, the preferences $\succsim_i$ of a player i over players extend to preferences over coalitions in such a way that $S \succsim_i T$ if and only if either (a) some j in T is unacceptable to i or (b) all players in S and T are acceptable to i and $\max_i(S \setminus \{i\}) \succsim_i \max_i(T \setminus \{i\})$. Analogously, in a *W-hedonic* game we have that $S \succsim_i T$ if and only if either (a) some j in T is unacceptable to i or (b) all players in S and T are acceptable to i and $\min_i(S \setminus \{i\}) \succsim_i \min_i(T \setminus \{i\})$. For *hedonic games with $\mathcal{W}$-preferences* (or *$\mathcal{W}$-hedonic games*) are such that $S \succsim_i T$ if and only if $\min_i(S \setminus \{i\}) \succsim_i \min_i(T \setminus \{i\})$. Finally, *hedonic games with $\mathcal{B}$-preferences* (or *$\mathcal{B}$-hedonic games*) are defined such that $S \succ_i T$ if and only if (a) $\max_i(S \setminus \{i\}) \succ_i \max_i(T \setminus \{i\})$ or (b) both $\max_i(S \setminus \{i\}) \sim_i \max_i(T \setminus \{i\})$ and $|S| < |T|$.[1]

A solution of a hedonic game is a partition of the players in coalitions. In this respect, the main focus has been on solutions that capture a notion of stability. Thus, a partition π is said to be *Nash stable* (*NS*) if no player can benefit from moving to another (possibly empty) coalition in π. Partition π is *individually stable* (*IS*) if no player can benefit from moving to another (possibly empty) coalition T in π without making the members of T worse off. Finally, π is *contractually individually stable* (*CIS*) if no player would strictly prefer to move from his coalition S to another existing (pos-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1] $\mathcal{W}$- and $\mathcal{B}$-hedonic games were originally introduced by Cechlárová and Romero-Medina [4]. For $\mathcal{B}$-hedonic games the dependence on coalition size prevents the grand coalition N to be trivially the most preferred one by all players.

sibly empty) coalition T in π without making neither the members of S nor the members of T worse off. It is easily seen that Nash stability implies individual stability and that individual stability implies contractual individual stability. Another, minimal, requirement, automatically satisfied by NS and IS partitions, is that a partition is *individually rational* (*IR*), i.e., that it assigns each player to a coalition that is acceptable to him.

We analyze the computational complexity of deciding the existence of and computing IS, NS, and CIS & IR partitions in B-, $\mathcal{B}$-, W-, and $\mathcal{W}$-hedonic games.

2. RESULTS

We first note that W-hedonic games are equivalent to hedonic games with $\mathcal{W}$-preferences if only individually rational outcomes are considered. For both $\mathcal{W}$- and B-hedonic games, if preferences do not allow unacceptable players, then the partition consisting of the grand coalition is Nash stable and therefore individually stable. However, if unacceptability of players is expressed, we obtain relatively more negative results. Our hardness results are by reductions from SAT and rely on the idea of a so-called *stalker game*. The simplest example is the hedonic game $(N, \succsim)$ where $N = \{1, 2\}$ and $\{1\} \succ_1 \{1, 2\}$ and $\{1, 2\} \succ_2 \{2\}$. Then, player 2 will stalk player 1 and the game has no NS partition.

THEOREM 1. *For $\mathcal{W}$-hedonic and B-hedonic games, deciding whether a NS partition exists is NP-complete.*

PROOF (SKETCH). By a reduction from SAT. Let $\varphi = X_1 \wedge \cdots \wedge X_k$ be a Boolean formula in conjunctive normal form in which all and only the Boolean variables $p_1, \ldots, p_m$ occur. Now define the B-hedonic game $(N, \succsim)$, where $N = \{X_1, \ldots, X_k\} \cup \{p_1, \neg p_1, \ldots, p_m, \neg p_m\} \cup \{0, 1\}$.

Define the preferences $\succsim$ such that for each literal p or $\neg p$, and each clause $X = (x_1 \vee \cdots \vee x_\ell)$,

$$\begin{array}{rl} p\colon & (0, 1, \text{———}, \; p \parallel \neg p, X_1, \ldots, X_k) \\ \neg p\colon & (0, 1, \text{———}, \neg p \parallel \; p, X_1, \ldots, X_k) \\ X\colon & (1, \text{———} \mid X_1, \ldots, X_k \parallel 0, x_1, \ldots, x_\ell) \\ 0\colon & (\text{———}, 0 \parallel 1, X_1, \ldots, X_k) \\ 1\colon & (\text{———}, 1 \parallel 0, X_1, \ldots, X_k), \end{array}$$

where " ——— " stands for the players not explicitly mentioned in the list, " | " for $\succ_i$, commas for $\sim_i$, and the players to the right of " ‖ " are unacceptable.

To prove that φ is satisfiable if and only if an NS partition for $(N, \succsim)$ exists, first assume that there exists a valuation v that satisfies φ. Define the partition $\pi = \{\{1, x'_1, \ldots, x'_{\ell'}\}, \{0, x''_1, \ldots, x''_{\ell''}\}, \{X_1, \ldots, X_k\}\}$ where $x'_1, \ldots, x'_{\ell'}$ are the literals rendered true by v and $x''_1, \ldots, x''_{\ell''}$ are those that are rendered false. It can easily be verified that π is NS-stable.

For the opposite direction, assume that there is a NS partition π. Then, for each clause $X = (x_1 \vee \cdots \vee x_\ell)$ there is some literal $x \in \{x_1 \vee \cdots \vee x_\ell)\}$ that is in π in the same coalition as 1; if not, X would become the stalker of 1. One can show that setting to true all the literals that are in the same coalition as 1 results in an assignment that satisfies φ. $\square$

The reduction in the proof of Theorem 1 is the prototype for more complicated reductions used to establish the results on NP-completeness in Table 1. These results involve an extended concept of a stalker game.

class	preferences	NS	IS	CIS & IR
$\mathcal{B}$	general	?	in P*	in P*
$\mathcal{B}$	strict	in P	in P*	in P*
B	general	NPC	NPC	in P*
B	strict	NPC	NPC	in P*
$\mathcal{W}$/W	general	NPC	NPC	in P*
$\mathcal{W}$/W	strict	NPC	?	in P*

Table 1: Complexity of individual-based stability. The positive results even hold for computing stable partitions whereas the NP-completeness results even hold for checking the existence of a stable partition. An asterisk indicates that a stable partition is guaranteed to exist.

EXAMPLE 1 (EXTENDED STALKER GAME). *Let $N = \{0, \ldots, 4\}$ and, assuming arithmetic modulo 5, the preferences over N of each player i be given by:*

$$i + 1 \succ_i i - 1 \succ_i i \succ_i \cdots$$

Then, in the B-, W- and $\mathcal{W}$-hedonic games induced by these preferences each player i stalks player $i+1$, joining him in any coalition whenever $i+1$ is alone. Consequently, no IS partition exists.

We also obtain some positive results. Firstly, a CIS and IR partition can be computed in polynomial time for all classes of games considered by starting with the individually rational partition of singletons and allowing arbitrary CIS deviations. For $\mathcal{B}$-hedonic games, in which a coalition is unacceptable only if all other players are unacceptable, positive results are even easier to obtain. In particular, we show that for $\mathcal{B}$-hedonic games, an IS partition is guaranteed to exist and can be computed in polynomial time.

Our results are summarized in Table 1 (for details and proofs, please see [1]). We obtain a general insight that in hedonic games based on extensions of preferences over players to preferences over coalitions, the following property can lead to intractability: the presence of an unacceptable player rendering a coalition unacceptable.[2]

REFERENCES

[1] H. Aziz, P. Harrenstein, and E. Pyrga. Individual-based stability in hedonic games depending on the best or worst players. Technical report, http://arxiv.org/abs/1105.1824, 2011.

[2] K. Cechlárová and J. Hajduková. Computational complexity of stable partitions with B-preferences. *International Journal of Game Theory*, 31(3):353–354, 2002.

[3] K. Cechlárová and J. Hajduková. Stable partitions with $\mathcal{W}$-preferences. *Discrete Applied Mathematics*, 138(3): 333–347, 2004.

[4] K. Cechlárová and A. Romero-Medina. Stability in coalition formation games. *International Journal of Game Theory*, 29:487–494, 2001.

[5] J. H. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–1003, 1980.

[2]This work is supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/10-1, BR 2312/3-3, and BR 2312/9-1.

Influence and aggregation of preferences over combinatorial domains

(Extended Abstract)

Nicolas Maudet
LIP6, UPMC
Paris, France
nicolas.maudet@lip6.fr

Maria Silvia Pini
Univ. of Padova, Italy
mpini@math.unipd.it

Francesca Rossi
Univ. of Padova, Italy
frossi@math.unipd.it

Kristen Brent Venable
Univ. of Padova, Italy
kvenable@math.unipd.it

ABSTRACT

In a multi-agent context where a set of agents declares their preferences over a common set of candidates, it is often the case that agents may influence each others. Recent work has modelled the influence phenomenon in the case of voting over a single issue. Here we generalize this model to account for preferences over combinatorially structured domains including several issues. When agents express their preferences as CP-nets, we show how to model influence functions and how to aggregate preferences by interleaving voting and influence convergence.

Categories and Subject Descriptors

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

General Terms

Theory, Algorithms

Keywords

Preference aggregation, influence, combinatorial domains

1. INTRODUCTION

Often a set of agents needs to select a common decision from a set of possible decisions, over which they express their preferences, and such a decision set has a combinatorial structure. That is, it can be seen as the combination of certain issues, where each issue has a set of possible instances. Consider for example a car: usually it is not seen as a single item, but as a combination of features, such as its engine, its shape, its color, and its cost. Each of these features has some possible instances, and a car is the combination of such feature instances. If a family needs to buy a new car, each family member may have his own opinion about each feature of a car, and the task is to choose the car that best fits the preferences of everybody. But suppose the mother knows well the CO2 emissions of the different cars: her preference regarding the engine may affect the one of his son who is concerned by the carbon footprint. In other words, agents may *influence* each other, leading their preferences to be modified accordingly.

The concept of influence has been widely studied in psychology, economics, sociology, and mathematics. Recent work has modelled the influence phenomenon in the case of taking a decision over a single binary issue [2]. Under this iterative model of influence, we may pass from state to state until stability holds, or we may also not converge.

Here we generalize this model to account for preferences over combinatorially structured domains including several issues. Complex influence statements may be represented, *e.g.* influences which depend on the context ("if my daughter prefers the yellow color for the car, I will follow her; otherwise I will stick to my inclinations") or which may involve different features of different agents ("If my wife and my son prefer the small car, then I would prefer the green color").

Usually preferences over combinatorially structured domains are expressed compactly, otherwise too much space would be needed to rank all possible alternatives. CP-nets are a successful framework that allows one to do this [1]. They exploit the independence among some features to give conditional preferences over small subsets of them. CP-nets have already been considered in a multi-agent setting [5, 4]. Here we incorporate influences among agents.

2. MODELLING PREFERENCES AND INFLUENCES

We assume each agent expresses its preferences over the candidates via an acyclic CP-net [1]. CP-nets are sets of *conditional preference statements* (cp-statements) each stating a total order over the values of a variable (say X), possibly depending on each combination of values of a set of other variables (say $X_1, \ldots, X_n$). X is said the dependent variable and $X_1, \ldots, X_n$ are the parents of X. Acyclic CP-nets are CP-nets where the dependency graph (with arcs from parents to dependent variables) does not have cycles.

We also assume that the dependency graphs of such CP-nets must all be compatible with a linear order O over the features: for each voter, the preference over a feature is in-

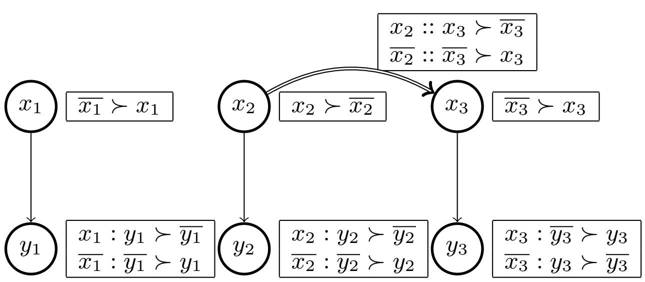
Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

dependent of features following it in O (O-legality in [3]). A *profile* models the *initial inclination* of all agents (their opinions over the candidates before they are influenced) as a collection of n such acyclic CP-nets over the m features.

To model influences, we use conditional influence statements. A *conditional influence statement* (ci-statement) on variable X has the form $O(X_1), \ldots, O(X_k) :: o(X)$, where $o(Y)$ is an ordering over the values of variable Y, for $Y \in \{X_1, \ldots, X_n, X\}$. Variables $X_1, \ldots X_k$ are the influencing variables and variable X is the influenced variable.

A ci-table is a collection of ci-statements with the same influencing and influenced variables, and containing at most one ci-statement for each ordering of the influencing variables. An *I-profile* is a triple (P, O, S), where P is a profile, O is an ordering over the m features of the profile, and S is a set of ci-tables. We assume that the ci-tables of an I-profile must be such that each variable can be influenced only by variables in her level or in earlier levels, but not in the same ci-statement. Thus, ci-arcs in an I-profile can create cycles only within variables of the same level.

EXAMPLE 1. *There are three agents (thus three CP-nets, all compatible with the ordering $X \succ Y$), and two binary features: X and Y, with values, respectively, x and $\overline{x}$, and y and $\overline{y}$. The I-profile has six variables denoted by X_1, X_2, X_3, Y_1, Y_2, and Y_3. Each variable X_i (resp., Y_i), with $i \in \{1, 2, 3\}$, has two values denoted by x_i and $\overline{x_i}$ (resp., y_i and $\overline{y_i}$). Note that cp-statements are denoted by single-line arrows while ci-statements are denoted by doubled-line arrows. As it can be seen, agent 3 is influenced on feature X by agent 2.*



There is a very useful relationship between ci-statements and cp-statements:

THEOREM 1. *Given an influence function f, consider the set of cp-statements N corresponding the ci-statements $ci(f)$. Then the undominated outcomes of N coincide with the stable states of f.*

While this result allows for a very simple integration of ci- and cp-statements in the same profile, it is important to still distinguish between the initial inclinations (cp-statements) and the influences (ci-statements). In fact, influences modify the initial inclination by overriding the preferences, but the opposite does not hold.

3. AGGREGATING PREFERENCES

We propose a way to aggregate the preferences contained in an I-profile, while taking into account the influence functions. The method we propose includes three main phases:

- *Influence iteration within one level*: For each feature, we consider the influences among different variables modelling this feature. An iterative algorithm is used: it takes all variables regarding the same feature and starts with the assignment corresponding to the initial inclination. The output is a single state. Either the algorithm, by iteratively applying the influences, ended up in a stable state; or it detected a cycle and used a subroutine to select nevertheless a single state.
- *Propagation from one level to the next one*: Once the variables of a certain level have been fixed to some values, we propagate to the next level by considering the ci- and cp-statements that go from this level to the next one. As influence overrides preference, we first look at the ci-tables and set the inclination of the influenced variables according to such tables. For the variables whose inclination has not been determined after this step, their inclination will be determined by their cp-tables. After this, we are ready to handle the next level as we did for the first one, since all of its variables are now subject only to influence functions.
- *Preference aggregation*: Since at each level we obtain a possibly different value for the variables modelling the same feature, we may either aggregate at each level (LA, *Level Aggregation*) or only at the end of the procedure (FA, *Final Aggregation*) when each agent has its most preferred candidate. Under LA (using majority since variables are binary) we assign the same value to all variables. Then we propagate such a choice to the next level and start again with an influence iteration. Under FA, we leave the variable values in each level as they are after the influence iteration and proceed until all levels have been handled. At this point, we have a most preferred candidate for each agent, and we can obtain a winning candidate by any voting rule that needs the top choices, such as plurality.

The two approaches may yield different results (this can observed on our example, where the winner is $\langle X = x, Y = y\rangle$ under LA and $(X = \overline{x}, Y = \overline{y})$ under FA. However, the choice of the ordering O does not matter as far as the winner is concerned, no matter if we use LA or FA.

4. ACKNOWLEDGMENTS

This work has been partially supported by the MIUR PRIN 20089M932N project 'Innovative and multi-disciplinary approaches for constraint and preference reasoning'.

5. REFERENCES

[1] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *JAIR*, 21:135–191, 2004.

[2] M. Grabisch and A. Rusinowska. Iterating influence between players in a social network. In *Proceedings of the 16th Coalition Theory Network Workshop*, 2011.

[3] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical social sciences*, 57:304–324, 2009.

[4] K. Purrington and E. H. Durfee. Making social choices from individuals' cp-nets. In *AAMAS*, page 179, 2007.

[5] F. Rossi, K. Venable, and T. Walsh. mcp nets: Representing and reasoning with preferences of multiple agents. In *AAAI'04*, pages 729–734, 2004.

Manipulation with Randomized Tie-Breaking under Maximin

(Extended Abstract)

Michael Zuckerman and Jeffrey S. Rosenschein
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel
{michez, jeff}@cs.huji.ac.il

ABSTRACT

In recent papers, Obraztsova et al. initiated the study of the computational complexity of voting manipulation under randomized tie-breaking [3, 2]. The authors provided a polynomial-time algorithm for the problem of finding an optimal vote for the manipulator (a vote maximizing the manipulator's expected utility) under the Maximin voting rule, for the case where the manipulator's utilities of the candidates are given by the vector $(1, 0, \ldots, 0)$. On the other hand, they showed that this problem is NP-hard for the case where the utilities are $(1, \ldots, 1, 0)$.

This paper continues that line of research. We prove that when the manipulator's utilities of the candidates are given by the vector $(1, \ldots, 1, 0, \ldots, 0)$, with k 1's and $(m-k)$ 0's, then the problem of finding an optimal vote for the manipulator is fixed-parameter tractable when parameterized by k. Also, by exploring the properties of the graph built by the algorithm, we prove that when a certain sub-graph of this graph contains a 2-cycle, then the solution returned by the algorithm is optimal.

Categories and Subject Descriptors

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

General Terms

Theory, Algorithms, Economics

Keywords

Computational Social Choice, Voting, Game Theory

1. INTRODUCTION

Social choice theory provides tools for formalizing preference aggregation among agents, using a wide variety of voting rules. The work of Gibbard and Satterthwaite [1, 4] showed, however, that with any reasonable voting rule, there would always be the possibility of a situation where agents were better off voting strategically, reporting untrue preferences to the voting mechanism in an attempt to manipulate the outcome. One of the popular techniques to overcome the susceptibility to manipulation uses computational complexity. Manipulation is always potentially useful, but in practice it might be exponentially difficult to find a useful manipulation (in the worst case). Complexity could potentially serve as a useful defense, as it does in cryptography.

Most recently, attention has been turned to the question of ties and tie-breaking rules; in a recent paper, Obraztsova et al. proposed an algorithm for finding an optimal vote under the Maximin voting rule when randomized tie-breaking is used, for a particular special case of manipulator utilities [3]. This work was extended later by a subset of the same authors [2], where they showed that for another special case of manipulator utilities, finding a "good enough" manipulation under Maximin is NP-complete. The bottom line of these papers is that "ties matter", i.e., the way in which ties are broken influences fundamental characteristics of voting rules, in particular their computationally feasible susceptibility to manipulation. The current paper continues this line of research, by more fully characterizing the nature of manipulation in the Maximin voting rule for more general settings of manipulator utilities.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. PRELIMINARIES

Voting An election is given by a set $C = \{c_1, \ldots, c_m\}$ of *candidates* (also called *alternatives*), and a set $V = \{v_1, \ldots, v_n\}$ of *voters*. The voters submit linear orders, R_i, over the candidates. We will sometimes use $\succ_i$ instead of R_i, for readability. If $c_k \succ_i c_j$, we say that i prefers c_k to c_j. We denote by $\mathcal{L}(C)$ the set of all linear orders over C. A list of n linear orders $\mathcal{R} = (R_1, \ldots, R_n) \in \mathcal{L}(C)^n$ is called a *preference profile*.

A *voting correspondence* is a mapping $\mathcal{F} : \mathcal{L}(C)^n \to 2^C$ which for every profile of the votes R determines a non-empty set of winners $S \subseteq C$. If $|\mathcal{F}(R)| = 1$, $\mathcal{F}$ is called a *voting rule*. In order to transform a voting correspondence into a voting rule, we need a *tie-breaking rule*. Formally, a tie-breaking rule is a mapping $\mathcal{T}$ which, given a non-empty set of tied candidates S, returns the winning candidate $c \in S$. In this work, we use the *randomized tie-breaking rule*, i.e., the rule where ties are broken uniformly at random.

In this paper we consider the Maximin voting rule (or, more precisely, voting correspondence). The maximin score of a candidate $c \in C$ is defined as the number of voters who prefer c to c's toughest opponent, i.e., $\min_{d \in C \setminus \{c\}} |\{i \mid c \succ_i d\}|$. The candidates with maximum score win.

Manipulation: Given a preference profile $\mathcal{R} = (R_1, \ldots, R_n)$

over a set of candidates C, for any preference order $L \in \mathcal{L}(C)$ we denote by (R_{-i}, L) the profile $(R_1, \ldots, R_{i-1}, L, R_{i+1}, \ldots, R_n)$. In order to model the manipulation with randomized tie-breaking, we follow [3] and [2], and assume that the manipulator has non-negative utilities over the set of candidates, $u(c)$ for every $c \in C$. We assume that the utilities are consistent with the manipulator's preference order $\succ_i$, i.e., $u(a) \geq u(b)$ if and only if $a \succ_i b$. In this work, we deal with the case where for all $c \in C$, $u(c) \in \{0, 1\}$. Now, if a voting correspondence $\mathcal{F}$ outputs a set $S \subseteq C$, the manipulator's expected utility is $\hat{u}(S) = \frac{1}{|S|}\sum_{c \in S} u(c)$. L is said to be v_i's optimal vote if for all linear orders $L' \in \mathcal{L}(C)$ it holds that $\hat{u}(\mathcal{F}(R_{-i}, L)) \geq \hat{u}(\mathcal{F}(R_{-i}, L'))$.

3. RESULTS

3.1 Parameterized Complexity Result

The next theorem continues the line of research started by Obraztsova et al. [3, 2], providing the parameterized complexity of manipulation when the manipulator utilities are given by the vector $(1, \ldots, 1, 0, \ldots, 0)$ with k 1's.

THEOREM 1. *Let $1 \leq k \leq m-1$. Suppose that the utilities of the manipulator are as follows: $u(c_i) = 1$ for $1 \leq i \leq k$, $u(c_i) = 0$ for $k+1 \leq i \leq m$, where the order c_i on the alternatives is the preference order of the manipulator. Then the problem of finding an optimal manipulation is in FPT (fixed-parameter tractable), when parameterized by k. More specifically, there exists an algorithm for finding an optimal manipulation in $O(k!k^2 + (n+m)m^2)$ time.*

PROOF. We consider an election $E = (C, V)$ where $C = \{c_1, \ldots, c_m\}$, $V = \{v_1, \ldots, v_n\}$, and v_n is the manipulator. We denote for a candidate $c_i \in C$ by $s(c_i)$ the Maximin score of c_i in the election $E' = (C, V')$, where $V' = \{v_1, \ldots, v_{n-1}\}$. Let $s = \max_{c_i \in C}\{s(c_i)\}$. Suppose that the utilities of the manipulator are as defined above. Let $C_1 = \{c_1, \ldots, c_k\}$ be the set of candidates having utility 1, and $C_0 = C \setminus C_1$ be the set of candidates with utility 0. Let $X = \operatorname{argmax}_{c_i \in C_1}\{s(c_i)\}$. Since the manipulator can only increase the score of any candidate by 1 or by 0, if for $x \in X$, $s(x) < s - 1$ then, clearly, for any vote of the manipulator his utility will be 0. So let us assume that for $x \in X$, $s - 1 \leq s(x) \leq s$. Following Obraztsova et al. [3], we define a directed graph G with a vertex set C, where there is an edge from c_i to c_j when there are exactly $s(c_j)$ voters in V' that rank c_j above c_i. We color the vertices of G as follows. Let $x \in X$ be any candidate. All the candidates $c \in C \setminus X$ with the score $s(c) = s(x) + 1$ will be purple; all the candidates $c \in C \setminus X$ with the score $s(c) = s(x)$ will be red; and all the rest of the candidates will be green. Note that by construction, all the candidates in X are green.

In order to find an optimal vote of the manipulator, we will use the recursive procedure $\mathcal{A}(H)$ described in [3], where H is an input colored directed graph, with one small modification: in step 2, if H contains any of the vertices of X, we add them (in some arbitrary order) to the top of the list L built by the procedure and remove them from H. We call this modified procedure $\mathcal{A}'(H)$.

In our algorithm, we first call $\mathcal{A}'(G)$. This way, if the expected utility of an optimal vote is greater than 0, we get an ordering L in which the number of candidates with utility 0 having the highest scores is minimal (the proof of this is the same as the original proof). Also, L contains all the candidates of X in the $|X|$ top positions. In the next step, we go over all the $|X|! \leq k!$ permutations of the candidates in X and check in which permutation the number of candidates of X whose score grows by 1 is maximal. Then we return this permutation combined with L. Note that the permutation of the candidates in X does not affect the scores of the other candidates (what really matters here is that all the candidates in X are ranked above all the other candidates). So by changing the permutation from what was calculated by $\mathcal{A}'$, we do not hurt the optimality of the solution computed by $\mathcal{A}'$.

One can verify that the running time of this algorithm is $O(k!k^2 + (n+m)m^2)$, and so, the problem is in FPT. $\square$

COROLLARY 2. *When the number of 1's in the manipulator's utility vector, $k = O(\frac{\log m}{\log\log m} + \frac{\log n}{\log\log n})$ then the algorithm for finding an optimal manipulation runs in polynomial time.*

PROOF. When $k = O(\frac{\log m}{\log\log m} + \frac{\log n}{\log\log n})$, $k! = O(m+n)$, and the result follows. $\square$

3.2 Characterization Result

Here we state that the graph G of the election as defined above has some special property, which sometimes may help in computing the maximum expected utility of the manipulator.

THEOREM 3. *Suppose that the utilities and the set of candidates X are as defined above. Let G be the graph of the election as defined above, and let $H = (X, E)$ be the subgraph of G induced by the vertices of X. If there exist two vertices $x, y \in X$ such that $(x, y) \in E$ and $(y, x) \in E$ then H is complete, i.e., for all $a, b \in X$, $(a, b) \in E$ and $(b, a) \in E$.*

COROLLARY 4. *If the conditions of Theorem 3 hold, we can compute an optimal vote of the manipulator in polynomial time.*

4. ACKNOWLEDGMENTS

We would like to thank Edith Elkind and Nati Linial for helpful discussions on the topic of this paper. This work was partially supported by Israel Ministry of Science and Technology grant #3-6797, and the Google Inter-University Center for Electronic Markets and Auctions.

5. REFERENCES

[1] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[2] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *The Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 319–324, 2011.

[3] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *The Tenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 71–78, 2011.

[4] 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.

Learning Performance of Prediction Markets with Kelly Bettors

(Extended Abstract)

Alina Beygelzimer
IBM Research
beygel@us.ibm.com

John Langford
Yahoo! Research
jl@yahoo-inc.com

David M. Pennock
Yahoo! Research
pennockd@yahoo-inc.com

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Economics

Keywords

Auction and mechanism design, electronic markets, economically motivated agents, multiagent learning

1. INTRODUCTION

Consider a prediction market, in which participants can trade *shares* (binary options) at the current market price p_m. Each share is worth \$1 if the event occurs, and nothing otherwise. What fraction of your wealth w should you risk if you believe the probability of the event is p? Buying is favorable if $p > p_m$, in which case risking your entire wealth will maximize your expected profit with respect to your belief. However, that's extraordinarily risky: A single stroke of bad luck loses everything. On the other hand, risking a small fixed amount cannot take advantage of compounding growth.

The *Kelly criteria* prescribes investing f^*w dollars, where for $p > p_m$,

$$f^* = \frac{p - p_m}{1 - p_m}$$

(buy order). For $p < p_m$, you should bet against the outcome (sell order) with

$$f^* = \frac{(1-p) - (1-p_m)}{1 - (1-p_m)} = \frac{p_m - p}{p_m}.$$

Kelly betting maximizes the expected compounding growth rate of wealth, or equivalently the expected logarithm of wealth [2, 4, 10].

We consider a prediction market, where participant i starts with wealth w_i, with $\sum_i w_i = 1$. Each participant i uses Kelly betting to determine the fraction of their wealth to bet, depending on their prediction p_i.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

We model the market as an auctioneer matching supply and demand, taking no profit and absorbing no loss, with p_m selected to clear the market. Agents are "price takers" that optimize according to the current price and do not reason further about what the price might reveal about the other agents' information. (In the fractional Kelly setting, however, agents do consider the market price as information and weigh it along with their own.)

Observation 1. *The market prediction p_m is always a wealth-weighted average of the agents' predictions p_i,*

$$p_m = \sum_i p_i w_i.$$

Proof. The market equilibrium occurs at price p_m where the payin is equal to the payout. If the event occurs,

$$\sum_{i:p_i>p_m} \frac{p_i - p_m}{1 - p_m} w_i + \sum_{i:p_i<p_m} \frac{p_m - p_i}{p_m} w_i = \frac{1}{p_m} \sum_{i:p_i>p_m} \frac{p_i - p_m}{1 - p_m} w_i.$$

Simplifying, we get $\sum_i p_i w_i = p_m \sum_i w_i$. Applying $\sum_i w_i = 1$ finishes the proof. A similar calculation proves the observation if the event doesn't occur. □

An alternate derivation utilizes the fact that Kelly betting is equivalent to maximizing expected log utility. This result can be seen as a simplified derivation of that by Rubinstein [7, 8, 9] and is also discussed by Pennock and Wellman [6, 5] and Wolfers and Zitzewitz [11].

2. LEARNING PREDICTION MARKETS

Consider a sequence of prediction markets which may have varying true and predicted probabilities. What happens to the wealth distribution and hence the quality of the market prediction over time? We show that the market *learns* optimally for two well understood senses of optimal.

2.1 Wealth is redistributed according to Bayes' Law

In an individual round, if an agent's belief is $p_i > p_m$, their total wealth afterward depends on the outcome y according to

$$\text{If} \quad y = 1, \quad \left(\frac{1}{p_m} - 1\right) \frac{p_i - p_m}{1 - p_m} w_i + w_i = \frac{p_i}{p_m} w_i$$

$$\text{If} \quad y = 0, \quad -\frac{p_i - p_m}{1 - p_m} w_i + w_i = \frac{1 - p_i}{1 - p_m} w_i$$

Similarly if $p_i < p_m$, we get

$$\text{If} \quad y=1, \quad -\frac{p_m - p_i}{p_m} w_i + w_i = \frac{p_i}{p_m} w_i$$

$$\text{If} \quad y=0, \quad \left(\frac{1}{1-p_m} - 1\right)\frac{p_m - p_i}{p_m} w_i + w_i = \frac{1-p_i}{1-p_m} w_i,$$

which is identical.

If we treat the prior probability $P(i)$ that agent i is correct as w_i, the posterior probability of choosing agent i is

$$P(i \mid y=1) = \frac{P(y=1 \mid i)P(i)}{\sum_j P(y=1 \mid j)P(j)} = \frac{p_i w_i}{p_m},$$

which is precisely the wealth computed above for the $y = 1$ outcome, and similarly when $y = 0$. So Kelly bettors redistribute wealth according to Bayes' law, and the market price reacts exactly as if updating according to Bayes' law.

In the full version [1], we simulate a sequence of markets where an underlying true probability exists, showing that the market converges to the true objective frequency as if updating a Beta distribution, as the theory predicts.

Although individual agents are not adaptive, the market's composite agent computes a proper Bayesian update. Specifically, wealth is reallocated proportionally to a Beta distribution corresponding to the observed number of successes and trials, and price is approximately the expected value of this Beta distribution. A kind of collective Bayesianity emerges from the interactions of the group.

We also find empirically that, even if not all agents are Kelly bettors, among those that are, wealth is still redistributed according to Bayes' rule.

2.2 Market has low regret to the best agent

The assumptions in the section above are often too strong. The following result applies to *all* sequences of participant predictions p_{it} and *all* outcome sequences y_t, even when these are chosen adversarially. It states that even in this worst-case situation, the market performs no worse than $-\ln w_i$ compared to the best individual participant i, using standard analysis from learning theory [3].

We measure the accuracy of market predictions $\{p_t\}$ according to log loss as

$$L \doteq \sum_{t=1}^{T} I(y_t = 1)\log\frac{1}{p_t} + I(y_t = 0)\log\frac{1}{1-p_t}.$$

Similarly, the accuracy of participant i is measured as

$$L_i \doteq \sum_{t=1}^{T} I(y_t = 1)\log\frac{1}{p_{it}} + I(y_t = 0)\log\frac{1}{1-p_{it}}.$$

THEOREM 2. *For all sequences of participant predictions p_{it} and all sequences of revealed outcomes y_t,*

$$L \le \min_i L_i + \ln\frac{1}{w_i}.$$

PROOF. Initially, we have $\sum_i w_i = 1$. After T rounds, the total wealth of any participant i is given by

$$w_i \prod_{t=1}^{T}\left(\frac{p_{it}}{p_t}\right)^{y_t}\left(\frac{1-p_{it}}{1-p_t}\right)^{1-y_t} = w_i e^{L-L_i} \le 1,$$

where w_i is the starting wealth and the last inequality follows from wealth being conserved. Thus $\ln w_i + L - L_i \le 0$, yielding $L \le L_i + \ln\frac{1}{w_i}$. $\square$

Thus self-interested agents with log wealth utility create markets which learn to have small regret according to log loss.

3. FRACTIONAL KELLY BETTING

In the full version of the paper [1], we consider fractional Kelly betting, a commonly used, lower-risk variant of Kelly betting, and show that fractional Kelly agents behave like Kelly agents with beliefs weighted between their own and the market's. When a true underlying probability exists, the market price empirically converges to a time-discounted version of this probability [1]. We also propose a method for agents to learn their optimal fraction over time.

4. QUESTIONS

When agents have some utility other than log wealth utility, can we alter the structure of a market so that the market dynamics make the market price have low log loss regret? And similarly if we care about some other loss—such as squared loss, 0/1 loss, or a quantile loss—can we craft a marketplace such that log wealth utility agents achieve small regret with respect to these other losses?

5. REFERENCES

[1] A. Beygelzimer, J. Langford, and D. M. Pennock. Learning performance of prediction markets with Kelly bettors. *CoRR (arXiv)*, abs/1201.6655, 2012.

[2] L. Breiman. Optimal gambling systems for favorable games. In *Berkeley Symposium on Probability and Statistics, I*, pages 65–78, 1961.

[3] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[4] J. Kelly. A new interpretation of information rate. *Bell System Technical Journal*, 35:917–926, 1956.

[5] D. M. Pennock. *Aggregating Probabilistic Beliefs: Market Mechanisms and Graphical Representations*. PhD thesis, University of Michigan, 1999.

[6] D. M. Pennock and M. P. Wellman. A market framework for pooling opinions. Technical Report 2001-081, NEC Research Institute, 2001.

[7] M. Rubinstein. An aggregation theorem for securities markets. *Journal of Financial Economics*, 1(3):225–244, 1974.

[8] M. Rubinstein. Securities market efficiency in an Arrow-Debreu economy. *Americian Economic Review*, 65(5):812–824, 1975.

[9] M. Rubinstein. The strong case for the generalized logarithmic utility model as the premier model of financial markets. *Journal of Finance*, 31(2):551–571, 1976.

[10] E. O. Thorp. Optimal gambling systems for favorable games. *Review of the International Statistical Institute*, 37:273–293, 1969.

[11] J. Wolfers and E. Zitzewitz. Interpreting prediction market prices as probabilities. Technical Report 12200, NBER, 2006.

TrustBets: Betting over an IOU Network
(Extended Abstract)

Sharad Goel
Yahoo! Research
goel@yahoo-inc.com

Mohammad Mahdian
Google Research
mahdian@alum.mit.edu

David M. Pennock
Yahoo! Research
pennockd@yahoo-inc.com

Daniel M. Reeves
Beeminder
dreeves@beeminder.com

ABSTRACT
We consider the problem of operating a gambling market where players pay with IOUs instead of cash, and where in general not everyone trusts everyone else. Players declare their degree of trust in other players—for example, Alice trusts Bob for up to ten dollars, and Bob trusts Carol up to twenty dollars. The system determines what bets are acceptable according to the trust network. For example, Carol may be able to place a bet where she is at risk of losing ten dollars to Alice, even if Alice doesn't trust Carol directly, because the IOU can be routed through Bob. We show that if agents can bet on n events with binary outcomes, the problem of determining whether a collection of bets is acceptable is NP-hard. In the special case when the trust network is a tree, the problem can be solved in polynomial time using a maximum flow algorithm.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms
Economics

Keywords
Auction and mechanism design, electronic markets, economically motivated agents, peer-to-peer coordination, trust, reliability, and reputation

1. INTRODUCTION
A typical betting market is run by a central entity who is responsible for transferring payments from losers to winners. The market organizer collects cash deposits from the participants and carefully limits bets to ensure that all participants can cover their losses. Participants must tie up their cash in the system if they want to trade in the market.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

We consider an alternate framework where no central entity collects deposits or verifies the creditworthiness of participants. Instead, participants declare their degree of trust in one another by stating the maximum amount of money they are willing to loan to specific individuals. For example, Alice may say she trusts Bob for up to ten dollars, while Bob says he trusts Carol for up to twenty dollars.

In our setting, loans have no strings attached, meaning there are no restrictions on what recipients can do with the borrowed money. In particular, it is allowed and indeed expected that loan recipients can in turn loan out the money to other people that *they* trust. Thus we envision participants declaring relatively conservative levels of trust so that, if worse comes to worst, they are fully prepared to absorb any and all losses stemming from defaulted loans. In the above example, Alice is in effect vouching for Bob, promising to cover up to ten dollars of Bob's debt if he defaults in the system. Trust in this sense is a directed binary relation with a real-valued weight. The set of all declarations of trust forms a weighted directed graph, known as the trust graph or *trust network* [9].

A *payment* in our system takes the form of an "I owe you" (IOU) from one participant to another, or more generally a sequence of IOUs among several participants. For example, a \$10 payment from Carol to Alice might be a direct IOU from Carol to Alice, or it might consist of two IOUs, one from Carol to Bob and one from Bob to Alice. A *feasible payment* is a payment that respects the trust network. More specifically, it is a payment that can be achieved via a series of binary IOUs following links backwards in the trust graph. In the running example, assuming no other IOUs have been issued, Carol's \$10 payment to Alice is feasible, since it can be achieved by issuing a \$10 IOU from Carol to Bob and a \$10 IOU from Bob to Alice, both within the limits that Alice and Bob declared.

Our previous work introduced and formalized the concept of a trust network as a distributed payment system and examined how to conduct a multi-unit auction when the buyers and seller are nodes in the network, showing the problem is NP-hard [9]. Subsequent work analyzed the liquidity of such trust networks [5], and their formation by strategic agents [6]. We note, moreover, that the trust network framework generalizes the case of budget-constrained agents, which has many practical applications and has been studied in both the economics and computer science literature [2, 3, 1, 7]. In addition to this theoretical work, there are at least two prototype implementations of trust networks [8, 10].

2. PROBLEM STATEMENT & RESULTS

We now formally define the problem of betting on a trust network, denominating bets in a hypothetical currency, *utils*, that represents an abstract measure of utility [10].

A trust network [9] is given by a weighted directed graph defined on a set of vertices $V = \{0, \ldots, m\}$ representing m agents. Edges in this graph represent trust relationships among agents. Each edge $(i, j) \in E$ with weight c_{ij} specifies that i has extended j a credit line of c_{ij} utils. The power of a trust network so defined is that arbitrary payments can be made by passing obligations between agents that explicitly trust each other if the network is sufficiently well-connected. A payment of x utils from agent u to agent v is feasible as long as there is a way to route x utils from u to v in the network, respecting the capacities given by the c_{ij}'s.

We consider the following simple betting scenario. Agents can place bets on n events with binary outcomes, each with fixed, even odds. That is, each event has two possible outcomes, denoted by -1 and 1, and if an agent places a bet of x utils on the outcome $b \in \{-1, 1\}$, it should be paid x utils if the outcome is b; and it should pay x if the outcome is $-b$. The final outcome of the n events is denoted by a vector in $\{-1, 1\}^n$, and the bets of an agent $i \in V$ is denoted by an n-tuple in $\mathbb{R}^n$, where a negative value $-x$ in the ℓ-th entry indicates a bet of value x for the outcome -1 in the ℓ-th event, and a positive value indicates a bet for the outcome 1. Therefore, if $\vec{x_i} \in \mathbb{R}^n$ denotes the bets of agent $i \in V$ and $\vec{v} \in \{-1, 1\}^n$ denotes the outcome, then the overall payment that agent i should receive (or pay) is given by the dot product $\vec{v} \cdot \vec{x_i}$. We assume that the bets are balanced, i.e., the sum of all bets equals the zero vector ($\sum_{i \in V} \vec{x_i} = \vec{0}$). This guarantees that under any outcome, the sum of payment vectors is zero, and therefore it is not necessary to inject any additional money into the network.

We study the problem faced by a mediator who is given a set of bets and needs to decide if these bets are feasible given the constraints that the underlying trust network imposes on the routing of payments among agents. We can define multiple versions of this problem, for example, deciding whether a given set of bets can be supported by the trust network, or selecting a maximal set of bets that can be supported. Here we focus on the decision version of the problem. (The decision problem in fact captures the complexity of the problem in the sense that variants of the problem for which the decision problem can be solved efficiently correspond to variants where the optimization problem can be solved efficiently.) Formally, the problem can be stated as follows:

> GAMBLING FEASIBILITY PROBLEM
> INPUT: Trust network $G = (V, E)$ with capacities c_{ij} on the edges, an integer n, and a bet $\vec{x_i} \in \mathbb{R}^n$ for each $i \in V$.
>
> QUESTION: Decide whether for *every* $\vec{v} \in \{-1, 1\}^n$, the payments $\vec{v} \cdot \vec{x_i}$ for every $i \in S$ can be routed through the trust network.

With this statement of the gambling problem, we find two results: (1) the problem is NP-hard in general, even for seemingly simple trust network structures; and (2) despite this hardness result, the problem is tractable if the network is a tree.

THEOREM 1. *The problem of determining feasibility of a gamble over a trust network is:*

1. *NP-hard in general, even if the trust network is a bidirected complete graph with uniform weights.*
2. *solvable in polynomial time if the corresponding undirected graph is a tree.*

Though we omit proofs of these results due to space constraints, we note the following useful reformulation of the problem. By the max-flow min-cut theorem [4], under any fixed outcome $\vec{v} \in \{-1, 1\}^n$, the problem of whether the payments under this outcome can be routed is equivalent to determining if for every set T of nodes in G, the total capacity of the edges from T to $\overline{T}$ is at least the total amount that the bettors in T win under the outcome $\vec{v}$. This amount can be written as $\max(0, \sum_{i \in T} \vec{v} \cdot \vec{x_i})$. Therefore, the gambling feasibility problem is equivalent to deciding whether for every set T of nodes, the capacity of the cut $(T, \overline{T})$ is at least the *maximum* amount that bettors in T can win under *any* outcome.

Acknowledgments

We are grateful for input from Arpita Ghosh and Yiling Chen.

3. REFERENCES

[1] C. Borgs, J. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC)*, pages 44–51, 2005.

[2] Y.-K. Che and I. Gale. Standard auctions with financially-constrained buyers. *Review of Economic Studies*, 65:1–21, 1998. Reprinted in The Economic Theory of Auctions, edited by Paul Klemperer.

[3] Y.-K. Che and I. Gale. The optimal mechanism for selling to a budget-constrained buyer. *Journal of Economic Theory*, 92:198–233, 2000.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 1990.

[5] P. Dandekar, A. Goel, R. Govindan, and I. Post. Liquidity in credit networks: A little trust goes a long way. *ACM Conference on Electronic Commerce*, 2011.

[6] P. Dandekar, A. Goel, M. P. Wellman, and B. Wiedenbeck. Strategic formation of credit networks. *World Wide Web Conference*, 2012.

[7] S. Dobzinski, R. Lavi, and N. Nisan. Multi-unit auctions with budget limits. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

[8] R. Fugger. The ripple project. `http://ripple.sourceforge.net`, 2004.

[9] A. Ghosh, M. Mahdian, D. M. Reeves, D. M. Pennock, and R. Fugger. Mechanism design on trust networks. In *Workshop on Internet and Network Economics (WINE)*, 2007.

[10] D. M. Reeves, B. M. Soule, and T. Kasturi. Yootopia! *SIGecom Exchanges*, 6:1–26, 2006.

On the Social Welfare of Mechanisms for Repeated Batch Matching

(Extended Abstract)

Elliot Anshelevich
Rensselaer Polytechnic Institute
eanshel@cs.rpi.edu

Meenal Chhabra
Rensselaer Polytechnic Institute
chhabm@cs.rpi.edu

Sanmay Das
Rensselaer Polytechnic Institute
sanmay@cs.rpi.edu

Matthew Gerrior
Rensselaer Polytechnic Institute
gerrim@cs.rpi.edu

ABSTRACT

We study hybrid online-batch matching problems, where agents arrive continuously, but are only matched in periodic rounds, when many of them can be considered simultaneously. Agents not getting matched in a given round remain in the market for the next round. This setting models several scenarios of interest, including many job markets as well as kidney exchange mechanisms. We consider the social utility of two commonly used mechanisms for such markets: one that aims for stability in each round (greedy), and one that attempts to maximize social utility in each round (max-weight). Surprisingly, we find that in the long term, the social utility of the greedy mechanism can be higher than that of the max-weight mechanism. We hypothesize that this is because the greedy mechanism behaves similarly to a soft threshold mechanism, where all connections below a certain threshold are rejected by the participants in favor of waiting until the next round. Motivated by this observation, we propose a method to approximately calculate the optimal threshold for an individual agent, based on characteristics of the other agents, and demonstrate empirically that social utility is high when all agents use this strategy.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics

Keywords

Matching, Search, Social welfare

1. INTRODUCTION

Many matching scenarios operate in a hybrid online/batch mode, where agents arrive and wait until the next market clearing period. In any given clearing period, all candidates currently waiting are considered for a match. Those who are successfully matched leave the market, while others wait for the next clearing period. This describes scenarios ranging from kidney exchange (which clear every few weeks) to academic job markets (typically once a year) [2, 3, 1].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. MODEL AND STRATEGIES

Time proceeds in discrete steps, and at each unit of time (a *round*) all agents who are thus far unmatched participate in a batch matching. At time $t = 0$ there are n agents, and at each future time period r new agents arrive. The agents connect to each other with probability p. Agents can be thought of as nodes on a graph. The existence of an edge between two nodes means there is a non-zero utility to both from being matched with each other. An edge between two agent i and j is associated with a weight u_{ij} that determines the utility of that matching. It is assumed that agents lose utility by a factor of δ ($\delta \in (0\ 1)$) per time unit for waiting.

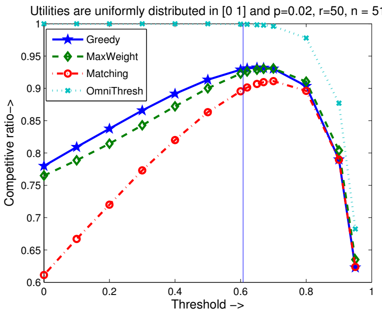
Social utility is additive, and given by:

$$U = \sum_{i,j \in \text{Matches}} u_{ij}(\delta^{t-t_i} + \delta^{t-t_j})$$

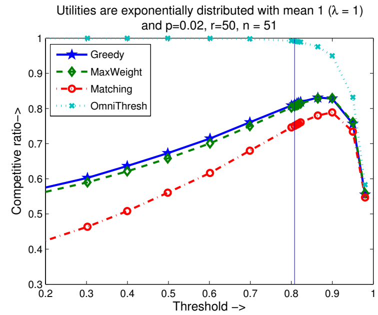
where t_i and t_j are the arrival times of agent i and agent j respectively, and t is the time at which they are matched.

We assume that u_{ij}'s are i.i.d draws from a stationary distribution $f(x)$ irrespective of the type of the agents to be connected and the time at which the edge is formed.

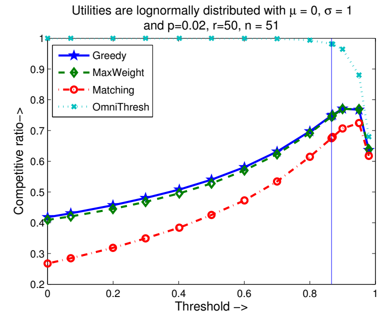
At each round, all unmatched agents can report their set of acceptable neighbors to the mechanism, so the mechanism finds an acceptable matching. Once this reporting is done, the agents cannot change their mind and have to accept the match chosen by the mechanism. Unmatched agents are eligible to be matched again in the next round. From an agent's perspective, selecting the acceptable matches can be seen as a sequential search problem. We can show that, under certain conditions, an agent's optimal strategy is the same in any round, and can be characterized by a reservation value t^* such that the agent should (pre-)reject all potential matches with utility less than t^* and be ready to accept any match with utility greater than t^*. The optimal threshold t^*



U1:Uniform($U[0\,1]$, $p = 0.02$)



E1:Exponential ($\lambda = 1$, $p = 0.02$)



L1:Lognormal(($\mu = 0, \sigma = 1$), $p = 0.02$)

Figure 1: Competitive ratio of social welfare (compared with the Omniscient matching) as a function of threshold. Surprisingly, greedy matching yields higher social welfare than max weight matching at lower thresholds. All the curves are unimodal which shows that there exists only one optimal threshold.

can be calculated by solving the following Bellman equation:

$$t^* = \delta(t^* \Pr(\neg \mathrm{M}) + \mathbb{E}(\mathrm{Utility}|\mathrm{M}) \Pr(\mathrm{M})) \qquad (1)$$

where M represents the event that the agent is matched with another with the utility of the match being greater than t^*. Quantifying Pr(M) is difficult. Therefore, we propose an algorithm to approximately calculate t^*. In Figure 1, the vertical blue line represents the threshold t^* calculated using our algorithm.

We calculate the social utility using the following set of mechanisms:

Online Maximum Weight Matching: The matching at each round is formed using the max-weight matching algorithm, using only edges such that $u_{ij} > \tau_{ij}$

Online Greedy: Similar to Online Maximum Weight Matching, except that matchings are formed using the greedy algorithm in each round. This is also roundwise stable.

Online Maximal Matching: After removing all edges below τ_{ij}, the mechanism picks an arbitrary maximal matching.

Omniscient Matching: This mechanism has foresight into the future and calculates the optimal solution to the offline problem.

OmniThresh Matching: This mechanism gives us an upper bound on the overall social utility of any threshold-based offline algorithm.

3. EXPERIMENTAL RESULTS

Figure 1 demonstrates the empirical performance of the threshold based mechanisms and the proposed threshold calculation algorithm. We observe the following:

1. Threshold mechanisms can significantly improve social welfare. Figure 1 shows the improvement in social welfare due to using threshold mechanisms. The vertical blue line represents the approximately optimal threshold t^* discussed in Section 2, and is close to the best threshold for maximizing social welfare (as well as to the best threshold for rational agents to use, as shown above).

In Figure 1, we also observe the unimodal behavior of the competitive ratio w.r.t threshold. This indicates that the social welfare exhibited by our thresholded online mechanisms is most likely a combination of two effects which counteract each other: (1) Having a high enough threshold removes some of the "online" nature of the mechanism, since it no longer matches pairs on low-quality edges, and instead waits to match them in future rounds, and (2) Having a high enough threshold removes high-quality edges from consideration, thus making a matching worse.

2. Greedy performs better than Max-Weight. Another interesting property apparent from the Figure 1 is the fact that the Greedy mechanism consistently performs better than the Max-Weight mechanism. The Greedy mechanism guarantees stability, while the Max-Weight mechanism maximizes social welfare. Thus, it seems surprising that, in aggregate, the Greedy mechanism is superior.

3. Thresholds matter more than edge weights: support for unweighted matching. Figure 1 shows that, while the Online Maximal Matching mechanism performs worse than the mechanisms that take actual edge weights into account, it still performs well (often within just a few percent of the other online mechanisms) when the threshold is picked appropriately.

Finally, we note that in order to scale to more realistic domains like kidney matching, our model needs to accommodate agents of multiple types. Our initial experiments with two agent types (the types are characterized by their probabilities of connecting to agents of all types) are promising: they suggest that threshold mechanisms, with thresholds chosen appropriately for each type, may continue to work well with multiple types. For further discussion and analysis of our results, we direct the reader to the full version of our paper.

4. ACKNOWLEDGEMENTS

The authors acknowledge support from NSF grants IIS-0952918, CCF-1101495 and CCF-0914782, and US-Israel BSF grant no. 2008404.

5. REFERENCES

[1] A. Abdulkadiroglu, P. Pathak, and A. Roth. Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *Amer. Econ. Rev.*, 99(5):1954–1978, 2009.

[2] D. Abraham, A. Levavi, D. Manlove, and G. O'Malley. The stable roommates problem with globally-ranked pairs. *Intern. and Net. Econ.*, pages 431–444, 2007.

[3] A. Roth and E. Peranson. The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review*, 89(4):748–780, 1999.

Merging Multiple Information Sources in Federated Sponsored Search Auctions

(Extended Abstract)

Sofia Ceppi
Politecnico di Milano
Milano, Italy
ceppi@elet.polimi.it

Enrico H. Gerding
Southampton University
Southampton, UK
eg@ecs.soton.ac.uk

Nicola Gatti
Politecnico di Milano
Milano, Italy
ngatti@elet.polimi.it

ABSTRACT

The recent increase of domain–specific search engines, able to discover information unknown by general–purpose search engines, leads to their federation into a single entity, called *federated search engine*. In this paper, we focus on how it can effectively merge sponsored search results, provided by the domain–specific search engines, into a unique list. In particular, we discuss the case in which the same ad can be provided by multiple sources, which requires information about the ad to be merged. We approach the problem of merging and sharing the revenue using mechanism design techniques. The main impossibility result we obtain points out there exists no mechanism that satisfies the customarily required properties. Thus, we present several mechanisms that violate at most one of these properties, and we experimentally analyze them using a real–world Yahoo! dataset.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Economics

Keywords

Mechanism Design, Sponsored Search Auctions

1. INTRODUCTION

Recently, we can see an increasing number of domain–specific search engines (DSSEs), e.g., bravofly.com, booking.com. Their advantage is that, for their specific domain, they are able to scour the *deep web* finding information (hidden in e.g. databases) that current general–purpose engines are unable to discover. This naturally leads to a new search paradigm where *federated search engines* (FSEs) integrate search results from heterogeneous DSSEs [1, 4] with the aim of providing the users with a 'one–stop shop'. However, similar to their general–purpose counterpart, DSSEs rely on revenue from sponsored search pay–per–click auctions. Currently, publishers, that use organic search results from general–purpose search engines, can use this service for their own websites for free, but in return the publishers require to also show the ads of the search engine. The revenue from clicks is then shared in a fixed way between the publisher and the search engine. However, this solution is not practical for an FSE since it then needs to display a separate list of ads for each domain–specific search engine. This is especially an issue if the same ad appears in more than one search engine (i.e. ads are shared among DSSEs). Although there is considerable literature on sponsored search auctions, to our knowledge [2] is the only other paper that considers the problem of merging sponsored search results for an FSE. However, the strong assumption is made that ads cannot be shared, which we relax in this paper.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. FEDERATED SEARCH ENGINE

Background The solution to the problem highlighted before, proposed in [2], is as follows. The FSE merges a selection of the ads into a coherent and unique list that it will display. To do this effectively, it needs detailed information about the ads that are known only by the DSSEs, i.e., the *qualities* (which is used to calculate an ad's click probability) of each ad, as well the *values* (i.e., the amount that the advertisers pay the search engines when their ad is clicked).

The authors of [2] approached the problem using *mechanism design* techniques. They obtained the following results. If the click probabilities are not influenced by the presence of other ads (i.e., there are no externalities), the standard VCG mechanism is *dominant–strategy* incentive compatible for this setting (i.e., truthfully elicits the values and qualities from the DSSEs). Furthermore, in the case of externalities between the ads, incentive compatibility can still be achieved using an *execution–contingent* VCG mechanism, where the payment is conditional on the realization of events (in this case the actual ads clicked by the user). However, this can only be achieved in *ex–post*, which requires others to be truthful (and thus is slightly weaker than having dominant strategies). Furthermore, although both mechanisms are weakly budget balanced (i.e. the FSE does not make a loss), this is only in expectation w.r.t. to events.

The New Challenge The work described above is based on the strong assumption that ads cannot be shared. When this assumption does not hold, as it commonly happens in practice, the nature of the problem changes fundamentally. Specifically, if an ad appears in multiple DSSEs, the FSE needs to merge all the reports received for this ad in order to accurately predict its click probability and produce efficient allocations. However, this could incentivize a DSSE to report a lower quality for some of its low–valued ads (for which it does not expect to gain any profit) purely in order to discredit the same ads from its competitors (thereby improving the allocation for its remaining ads and/or reducing the payment). Worse still, it could fabricate ads and simply pretend to have the same ads as its competitors. Our aim is to solve this challenge using mechanism design techniques.

3. CONTRIBUTIONS AND RESULTS

Our contributions and results are as follows:

1) We extend the models of federated sponsored search auctions described in [2] to the case in which advertisers are allowed to submit a bid to more than one DSSE, and we show how this can be posed as a mechanism design problem. In particular, the main differences with the model proposed in [2] are *(i)* the introduction a *value merge function*, $mv_a(\hat{\mathbf{v}})$, which returns the highest value associated with a certain ad a given the reported value $\hat{\mathbf{v}}$ (we assume that the DSSE that provides the highest value for an ad is the only DSSE that receives money from the advertiser when the ad is clicked), and a *quality merge function*, $mq(\hat{\mathbf{q}})$, which returns the merged quality for all the ads give the reported qualities $\hat{\mathbf{q}}$ (thus, the *click probability function*, α_a, results to be a function of the allocation and the quality merge function); *(ii)* the analysis of two scenarios: one where DSSEs are able to hide their ads and fabricate advertisers if this is in their interest, and one (called *verified*) where this is not possible since an ad verification mechanism is in place.

2) We theoretically prove that in general there exists no execution–contingent mechanism that simultaneously guarantees *allocation efficiency* (AE), *incentive compatibility* (IC), *individual rationality* (IR), and *weak budget balance* (WBB), even considering *ex–post* implementation.

3) Due to the result described above, we focus on mechanisms that guarantee three of these properties. A natural candidate to consider is the unique state–of–the–art mechanism for an FSE model [2]. We observe that, in the general setting in which ads can be shared, the mechanism violates both IR and WBB.

4) We provide a range of different mechanisms that violate at most one property among WBB, IR, and AE.

Mechanisms that violate the WBB property: We propose a mechanism for the general case, *MinRep*, and a mechanism for the verified case, *VerifiedMinRep*. The idea behind them is similar to the one proposed (for a completely different domain) in [5]: in the computation of a DSSE's payment, the mechanism considers virtual qualities (that substitutes the ones actually reported by the agent) such that the social welfare is minimized, and in turn also her payment. The formula of the payment of a DSSE s in expectation w.r.t. the set of clicked ads ω, depending on the reported qualities and values $\hat{\theta}$, given the true ones, θ, is given by:

$$\mathbb{E}_\omega[p_s(\hat{\theta}|\omega)|\theta] = \min_{\mathbf{q}'_s \in Q} sw^*(\hat{\mathbf{v}}, \langle \mathbf{q}'_s, \hat{\mathbf{q}}_{-s}\rangle) - \sum_{a \in f(\hat{\theta}) \setminus A^*_s} \alpha_a(f(\hat{\mathbf{v}}, \hat{\mathbf{q}}), mq(\mathbf{q})) \cdot mv_a(\hat{\mathbf{v}}_{-s}), \quad (1)$$

where Q is the set of virtual qualities, f is the efficient allocation, and sw^* is the social welfare of the efficient allocation (where $\mathbf{x}$ is an allocation):

$$sw^*(\mathbf{v}, \mathbf{q}) = \max \sum_{a \in \mathbf{x}} \alpha_a(\mathbf{x}, mq(\mathbf{q})) \cdot mv_a(\mathbf{v})$$

Note that in Eq. (1), $mq(\mathbf{q})$ is based on the true qualities since the payment is execution–contingent.

The difference between the two mechanisms lies in the ads considered in the set of possible virtual reports, Q. In particular, in the computation of a DSSE's payment, MinRep takes into account virtual qualities for all the ads, while VerifiedMinRep considers only the one corresponding to the DSSE's actual ads.

Mechanisms that violate the IR property: We propose a mechanism for the general case, *MaxRep*, and a mechanism for the verified case, *VerifiedMaxRep*. The basic idea is the same as the one presented for the mechanisms that violate the WBB property, but instead of minimizing the payment, we now maximize it.

Mechanism that violates the AE property: We focus our investigation on strictly randomized mechanisms. We propose a mechanism for the verified case, *VerifiedRand*, that randomly selects with uniform probability which ads to display and, if the ad is provided by multiple DSSEs, it randomly selects with uniform probability which is the one that will receive the money directly from the advertiser if the ad is clicked. Payments are equal to zero. Focusing on the non verified case, we prove that there exists no strictly randomized mechanism that violates only the AE property. Indeed, DSSEs can always influence the randomization by hiding ads that give low expected utility, thus increasing the chance that their 'good' ads are selected.

5) We experimentally evaluate and compare the mechanisms presented in the paper and the state of the art in terms of the FSE's expected revenue and the DSSEs' expected utility. The experimental analysis we propose is based on the *Yahoo! Webscope A3* dataset. Results of the analysis show that, as expected, different mechanisms are appropriate for domains with different requirements. Furthermore, MaxRep and VerifiedMaxRep turn out to be impractical due to the extreme negative utility they provide to the DSSEs. Similarly, MinRep is impractical because the FSE always gets a negative revenue. In contrast, VerifiedMinRep turns out to be suitable for the FSE when the number of DSSEs that share the same ad is not too high (with 5 DSSEs if less that the 60% of them shares the same ads). This pushes us to design a verification mechanism that the FSE can use to define the set of ads that belongs to each DSSE.

4. REFERENCES

[1] D. Braga, S. Ceri, F. Daniel, and D. Martinenghi. Mashing up search services. *IEEE Internet Computing*, 12(5):16–23, 2008.

[2] S. Ceppi, N. Gatti, and E. Gerding. Mechanism design for federated sponsored search auctions. In *AAAI*, 2011.

[3] E. H. Gerding, S. Stein, K. Larson, A. Rogers, and N. R. Jennings. Scalable mechanism design for the procurement of services with uncertain durations. In *AAMAS*, pages 649–656, May 2010.

[4] D. Martinenghi and M. Tagliasacchi. Proximity rank join. *The Proc. of the VLDB Endowment*, 3(1):352–363, 2010.

[5] S. D. Ramchurn, C. Mezzetti, A. Giovannucci, J. A. Rodriguez-aguilar, R. K. Dash, and N. R. Jennings. Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. *Journal of Artificial Intelligence Research*, pages 119–159, 2009.

A Truthful Learning Mechanism for Multi–Slot Sponsored Search Auctions with Externalities

(Extended Abstract)

Nicola Gatti
Politecnico di Milano
ngatti@elet.polimi.it

Alessandro Lazaric
INRIA Lille – Nord Europe
lazaric@inria.fr

Francesco Trovò
Politecnico di Milano
trovo@polimi.it

ABSTRACT

In recent years, effective sponsored search auctions (SSAs) have been designed to incentivize advertisers (advs) to bid their truthful valuations and, at the same time, to assure both the advs and the auctioneer a non–negative utility. Nonetheless, when the click–through–rates (CTRs) of the advs are unknown to the auction, these mechanisms must be paired with a learning algorithm for the estimation of the CTRs. This introduces the critical problem of designing a learning mechanism able to estimate the CTRs as the same time as implementing a truthful mechanism with a revenue loss as small as possible. In this paper, we extend previous results [2, 3] to the general case of multi–slot auctions with position– and ad–dependent externalities with particular attention on the dependency of the regret on the number of slots K and the number of advertisements n.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Algorithms, Economics, Theory

Keywords

Sponsored search auction, Learning mechanism

1. NOTATION AND BACKGROUND

We consider a standard model of SSAs. We denote by $\mathcal{N} = \{1, \ldots, n\}$ the set of ads. Each ad i is characterized by a *quality* ρ_i, defined as the probability that i is clicked once observed by the user, and by a *value* $v_i \in [0, V]$, that the adv receives once i is clicked (the value is zero if not clicked). While qualities ρ_i are common knowledge, values v_i are private information of the advs. We denote by $\mathcal{K} = \{1, \ldots, K\}$ with $K < n$ the set of available slot. An ad–slot allocation rule α is a full bijective mapping from n ads to n slots such that $\alpha(i) = k$ if ad $i \in \mathcal{N}$ is displayed at slot k. For all the non–allocated ads, $\alpha(i)$ takes an arbitrary value from $K+1$ to n so as to preserve the bijectivity of α. We also define the inverse slot–ad allocation rule $\beta = \alpha^{-1}$ such that $\beta(k) = i$ if slot k displays ad i (i.e., $\alpha(i) = k$). We denote by $\mathcal{A}$ and $\mathcal{B}$ the set of all the possible ad–slot and slot–ad mappings respectively. Finally, we define $\mathcal{A}_{-i} = \{\alpha \in \mathcal{A}, \alpha(i) = n\}$ as the set of allocations where ad i is never displayed. We adopt the cascade model [1, 5] to describe the user's behavior. The discount factor $\gamma_k(i)$ is the probability that a user, observing ad i in the slot $k-1$, will observe the ad in the next slot (γ_1 is set to 1 by definition). The cumulative discount factors $\Gamma_k(\beta)$, i.e., the probability with which a user observes the ad displayed at slot k given a slot–ad allocation β, is defined as $\Gamma_k(\beta) = \prod_{l=2}^{k} \gamma_l(\beta(l-1))$ for $2 \leq k \leq K$. With abuse of notation, we use interchangeably $\Gamma_k(\beta)$ and $\Gamma_k(\alpha)$. Given an allocation rule α, $\Gamma_{\alpha(i)}(\alpha)\rho_i$ is the *click through rate* (CTR), representing the probability of ad i to be clicked. Finally, we define the *social welfare* of an allocation α as the cumulative advs' expected values

$$\mathrm{SW}(\alpha) = \mathrm{SW}(\beta) = \sum_{i=1}^{n} \Gamma_{\alpha(i)}(\alpha)\rho_i v_i = \sum_{k=1}^{n} \Gamma_k(\beta)\rho_{\beta(k)} v_{\beta(k)}.$$

At each round, advs submit bids and the auction defines an allocation rule α and payments p_i. The Vickrey–Clark–Groves mechanism (VCG) satisfies a number of interesting properties, notably the incentive compatibility (IC) (i.e., no adv can increase its utility by misreporting its true value and $\hat{v}_i = v_i$), and it allocates ads according to the *efficient* allocation $\alpha^* = \arg\max_{\alpha \in \mathcal{A}} \mathrm{SW}(\alpha)$, and payments are set to 0 if the ad is not clicked and to

$$\tilde{p}_i = \frac{\mathrm{SW}(\alpha^*_{-i}) - \mathrm{SW}_{-i}(\alpha^*)}{\Gamma_{\alpha(i)}(\alpha)\rho_i}, \tag{1}$$

if the ad is clicked, so that $\mathbb{E}[\tilde{p}_i] = p_i = \mathrm{SW}(\alpha^*_{-i}) - \mathrm{SW}_{-i}(\alpha^*)$.

In many practical problems, the qualities ρ_i are not known in advance and must be estimated at the same time as the auction is deployed. This introduces a tradeoff between *exploring* different possible allocations so as to collect information about the quality of the advs and *exploiting* the estimated qualities so as to implement a truthful high–revenue auction. Let $\mathfrak{A}$ be an IC mechanism run over T rounds. At each round t, $\mathfrak{A}$ defines an allocation $\hat{\alpha}_t$ and prescribes an expected payment p_{it} for each ad i. The objective of $\mathfrak{A}$ is to obtain a revenue as close as possible to a VCG mechanism. More precisely, we measure the performance of $\mathfrak{A}$ as its cumulative regret over T rounds:

$$R_T(\mathfrak{A}) = T \sum_{i=1}^{n} p_i - \sum_{t=1}^{T} \sum_{i=1}^{n} p_{it}. \tag{2}$$

The mechanism $\mathfrak{A}$ is a *no–regret* mechanism if its per–round regret decreases to 0 as T increases, i.e., $\lim_{T\to\infty} R_T/T = 0$.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. REGRET BOUNDS

Similar to [3], we define an exploration–exploitation algorithm to approximate the VCG, which we refer to as A–VCG. The algorithm estimates the quality of each adv during a pure exploration phase of length τ when all the payments are set to 0. Then, quality estimates are used to set up a VCG for all the remaining $T-\tau$ rounds. At each round, we can collect K samples (click or not–click events), one from each slot. Let α_t (for $t \leq \tau$) be an arbitrary explorative allocation rule independent from the bids. It is easy to define a sequence of explorative allocations $\{\alpha_t\}_{t=1}^{\tau}$ such that the number of samples collected for each ad i is $S_i = \lfloor K\tau/n \rfloor$. We denote by $c^i_{\alpha_t(i)}(t) \in \{0,1\}$ the click–event at time t for ad i when displayed at slot $\alpha_t(i)$. Depending on the slot we have different CTRs, thus we reweigh each sample by the cumulative discount factor of the slot and we compute the estimated quality $\hat{\rho}_i$ as

$$\hat{\rho}_i = \frac{1}{S_i} \sum_{s=1}^{S_i} \frac{c^i_{\alpha_t(i)}(t)}{\Gamma_{\alpha_t(i)}(\alpha_t)}. \tag{3}$$

Depending on the specific user–model of the auction, we can build a high–probability confidence interval on $|\rho_i - \hat{\rho}_i|$ of size

$$\eta_p := \sqrt{\left(\sum_{k=1}^{K} \frac{1}{\Gamma_k^2}\right) \frac{n}{2K^2\tau} \log \frac{n}{\delta}}\,; \quad \eta_{pa} := \frac{1}{\Gamma_{\min}} \sqrt{\frac{n}{2K\tau} \log \frac{n}{\delta}},$$

with $\Gamma_{\min} = \min_{\alpha,k} \Gamma_k(\alpha)$, for pos– and pos/ad–dependent externalities respectively. After the exploration phase we define an upper–bound on the quality as $\hat{\rho}_i^+ = \hat{\rho}_i + \eta$. Given $\hat{\rho}_i^+$, we compute the estimated social welfare as $\widehat{\mathrm{SW}}(\alpha) = \sum_{i=1}^{n} \Gamma_{\alpha(i)}(\alpha)\hat{\rho}_i^+ v_i$. The corresponding efficient allocation is denoted by $\hat{\alpha} = \arg\max_{\alpha\in\mathcal{A}} \widehat{\mathrm{SW}}(\alpha)$. Once the exploration phase is over, if ad $i \in \mathcal{N}$ is clicked, then the adv is charged

$$\tilde{p}_i = \frac{\widehat{\mathrm{SW}}(\hat{\alpha}^*_{-i}) - \widehat{\mathrm{SW}}_{-i}(\hat{\alpha}^*)}{\Gamma_{\hat{\alpha}(i)}\hat{\rho}_i^+} \tag{4}$$

which corresponds to an expected payment $\hat{p}_i = \tilde{p}_i \Gamma_{\hat{\alpha}(i)}\rho_i$.

Position–dependent externalities. In case of pos–dependent externalities, the discount coefficients reduce to $\Gamma_k(\alpha) = \Gamma_k$, thus simplifying the computation of both the optimal allocation and the payments. In this case, A–VCG achieves the following regret performance.

THEOREM 1. *In a SSA auction with pos–dependent externalities, by optimizing the parameters τ and δ, the A–VCG is always **truthful** and it achieves a regret*

$$R_T \leq 18^{1/3} V T^{2/3} \Gamma_{\min}^{-2/3} K^{2/3} n^{1/3} (\log(n^2KT))^{1/3}. \tag{5}$$

where $\Gamma_{\min} = \min_k \Gamma_k \geq 0$.

Remark 1 (Bound). Up to numerical constants and logarithmic factors, the previous bound is $R_T \leq \tilde{O}(T^{2/3}K^{2/3}n^{1/3})$. We first notice that the A–VCG is a zero–regret algorithm since its per–round regret (R_T/T) decreases to 0 as $T^{-1/3}$, thus implying that it asymptotically achieves the same performance as the VCG. Furthermore, the dependence of the regret on n is sub–linear ($n^{1/3}$) and this allows to increase the number of advs without significantly worsening the regret. Finally, according to the bound (5) the regret has a sublinear dependency $\tilde{O}(K^{2/3})$ on the number of slots, meaning that whenever one slot is added to the auction, the performance does not significantly worsen. By analyzing the difference between the payments of the VCG and A–VCG, we notice that during the exploration phase the regret is $O(\tau K)$ (e.g., if all K slots are clicked at each explorative round), while during the exploitation phase the estimation errors sum over all the K slots, thus suggesting a linear dependency on K for this phase as well. Nonetheless, as K increases, the number of samples available per each ad increases as $\tau K/n$, thus improving the accuracy of the quality estimates by $\tilde{O}(K^{-1/2})$. As a result, as K increases, the exploration phase can be shortened (the optimal τ actually decreases as $K^{-1/3}$), thus reducing the regret during the exploration, and still have accurate enough estimations to control the regret of the exploitation phase.

Pos/ad–dependent externalities. In this case the learning problem is more complicated, and the regret is:

THEOREM 2. *In a SSA auction with pos/ad–dependent externalities, by optimizing the parameters τ and δ, the A–VCG is always **truthful** and it achieves a regret*

$$R_T \leq 6^{1/3} \frac{V}{\rho_{\min}} T^{2/3} \Gamma_{\min}^{-2/3} K^{2/3} n (\log(KT))^{1/3}. \tag{6}$$

Remark 1 (Differences with the previous bound). Up to constants and logarithmic factors, the previous bound is $R_T \leq \tilde{O}(T^{2/3}K^{2/3}n)$. We first notice that moving from pos– to pos/ad–dependent externalities does not change the dependency of the regret on the number of rounds T. The main difference is in the dependency on n and on the smallest quality $\rho_{\min}$. While the regret still scales as $K^{2/3}$, it has now a much worse dependency on the number of ads (from $n^{1/3}$ to n). We believe that it is mostly due to an intrinsic difficulty of the position/ad–dependent externalities. The intuition is that now in the computation of the payment for each ad i, the errors in the quality estimates cumulate through the slots (unlike the pos–dependent case where they are scaled by $\Gamma_k - \Gamma_{k+1}$). Nonetheless, this cumulated error should impact only on a portion of the ads (i.e., those which are actually impressed according to the optimal and the estimated optimal allocations). Thus we conjecture that this additional n term is indeed a rough upper–bound on the number of slots K and that we could obtain a regret $\tilde{O}(T^{2/3}K^{4/3}n^{1/3})$, where the dependency on the number of slots becomes super–linear. A more detailed discussion on this conjecture and on the dependency on $\rho_{\min}$ can be found in the extended version of this paper [4].

3. REFERENCES

[1] G. Aggarwal, J. Feldman, S. Muthukrishnan, and M. Pál. Sponsored search auctions with markovian users. In *WINE*, pages 621–628, 2008.

[2] M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. *CoRR*, abs/0812.2291, 2008.

[3] N. R. Devanur and S. M. Kakade. The price of truthfulness for pay-per-click auctions. In *ACM EC*, pages 99–106, 2009.

[4] N. Gatti, A. Lazaric, and F. Trovò. A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities. Technical report, 2012. http://hal.inria.fr/hal-00662549.

[5] D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. In *WINE*, pages 585–596, 2008.

Strategy-proof mechanisms for two-sided matching with minimum and maximum quotas

(Extended Abstract)

Suguru Ueda[1], Daniel Fragiadakis[2], Atsushi Iwasaki[1], Peter Troyan[2], and Makoto Yokoo[1]
1: Kyushu University, Japan, {ueda@agent., iwasaki@, yokoo@}inf.kyushu-u.ac.jp
2: Stanford University, California, {danielfragiadakis, petetroyan}@gmail.com

ABSTRACT

We consider the problem of allocating objects to agents when the objects have minimum quotas. There exist many real-world settings where minimum quotas are relevant. For example, in a hospital-resident matching problem, unconstrained matching may produce too few assignments to a rural hospital. Surprisingly, almost 50 years have passed after the seminal work by Gale and Shapley, no existing mechanism can guarantee minimum quotas so far; we did not know how to guarantee that a rural hospital has at least one resident.

In this paper, we propose mechanisms that can satisfy minimum quotas as well as standard maximum quotas. More specifically, we propose extended seat (ES) and multi-stage (MS) mechanisms modeled after the well-known deferred-acceptance (DA) and top trading cycles (TTC) mechanisms. Our proposed mechanisms are all strategy-proof, but a tradeoff exists between the DA and TTC based mechanisms regarding Pareto efficiency and elimination of justified envy. In addition, there exist a tradeoff between ES and MS mechanisms depending on the size of minimum quotas.

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

Game theory, two-sided matching, deferred-acceptance, top trading cycle, minimum quotas

1. INTRODUCTION

The matching theory literature has developed numerous mechanisms to solve the problem of assigning objects to a group of agents when the agents have privately known preferences and the objects have priorities over the agents. Many problems fit broadly into this context, including assigning students to schools [1] and kidneys to patients [5]. Most of the previous literature considers only maximum quotas. However, in many real-world problems, minimum quotas may be imposed as well. This paper proposes several new mechanisms to deal with both quotas, starting from assigning students to labs at a university.

Two popular mechanisms that are often used with no minimum quotas are the deferred-acceptance (DA) mechanism [2] and the top-trading cycles (TTC) mechanism [4]. Both are strategy-proof, but TTC always produces a Pareto efficient assignment, while DA does not. However, DA eliminates justified envy, while TTC, which allows students to trade their priorities, does not.

The standard DA and TTC mechanisms may fail to produce feasible matchings that meet minimum quotas. Thus, we take the standard DA and TTC mechanisms and make two modifications to each: "extended seat" (ES) and "multi-stage" (MS). Thus, our four mechanisms are ES-DA, ES-TTC, MS-DA and MS-TTC. These mechanisms produce feasible matchings while preserving the good properties of DA and TTC.

Strategy-proofness is often taken as an important property in matching markets, and indeed, all four of our mechanisms are strategy-proof. However, there is a tradeoff between efficiency and fairness. The two TTC based mechanisms produce Pareto efficient assignments, but will lead to justified envy. It is known that there is no strategy-proof mechanism that completely eliminates justified envy when minimum quotas are imposed. Additionally, Hamada et al. showed that there may be no stable matching [3], extending the concept of justified envy. Even worse, they also showed that finding a matching with the minimum number of blocking pairs is hard to approximate even if labs have a master list (i.e., all labs use the same priority ordering).

Thus, we propose a slight strengthening of justified envy that eliminates some potential blocking pairs under the standard definition, and show that our two DA mechanisms eliminate all such justified envy. Thus, the choice between using a TTC mechanism or a DA mechanism depends on which goal, either Pareto efficiency or elimination of justified envy, policymakers consider more important.

2. MODEL

A market is a tuple $(S, L, p, q, \succ_S, \succ_L, \succ_{ML})$. $S = \{s_1, s_2, \ldots, s_n\}$ is a set of students, $L = \{l_1, l_2, \ldots, l_m\}$ a set of labs, and $p = (p_{l_1}, \ldots, p_{l_m})$ and $q = (q_{l_1}, \ldots, q_{l_m})$ are the minimum and maximum quotas, respectively, for

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

each lab. We assume $0 \le p_l \le q_l$ for all $l \in L$ and $\sum_{l \in L} p_l \le n \le \sum_{l \in L} q_l$ to ensure a feasible matching exists. Define $e = n - \sum_{l \in L} p_l$ to be the number of "excess students".

Each student s has a strict preference relation $\succ_s$ over the labs, while each lab l has an idiosyncratic strict priority relation $\succ_l$ over the students. The vectors of all such relations are denoted $\succ_S = (\succ_s)_{s \in S}$ for the students and $\succ_L = (\succ_l)_{l \in L}$ for the labs. We assume that all labs are acceptable to all students and vice versa. In addition to the idiosyncratic lab preferences, there is a separate "master list (ML)" over all of the students. Without loss of generality, we let $s_1 \succ_{ML} s_2 \succ_{ML} \cdots \succ_{ML} s_n$.

A **matching** is a mapping $M : S \cup L \to S \cup L$ that satisfies: (i) $M(s) \in L$ for all $s \in S$, (ii) $M(l) \subseteq S$ for all $l \in L$, and (iii) for any s and l, we have $M(s) = l$ if and only if $s \in M(l)$. A matching is **feasible** if $p_l \le |M(l)| \le q_l$ for all $l \in L$. The labs are not strategic, i.e., $\succ_L$ and $\succ_{ML}$ are fixed and known to all students. Student s **envies** student s' at matching M if $M(s') \succ_s M(s)$ and if student s envies student s', this envy is **justified** if $s \succ_{M(s')} s'$. Also, we say s **strongly justifiably envies** s' at matching M if s envies s' and $s \succ_{ML} s'$ and $s \succ_{M(s')} s'$.

If student s justifiably envies student s', who is assigned laboratory $l = M(s')$, at matching M, then we say that student s and lab l form a blocking pair. A blocking pair can also be formed by a student and a lab with an empty seat, provided that moving the student to the lab with the empty seat results in a new matching that is feasible. Formally, given a matching M, student s **claims an empty seat at lab** l if (i) $l \succ_s M(s)$ (ii) $|M(l)| < q_l$ and (iii) $|M(M(s))| > p_{M(s)}$. Then, student s and lab l' form a **blocking pair** if either (i) s justifiably envies some student $s' \in M(l')$ or (ii) s claims an empty seat at l'.

3. EXTENDED-SEAT MECHANISMS

We consider an extended market $(S, \tilde{L}, \tilde{q}, \tilde{\succ}_S, \tilde{\succ}_L)$ where the set of students is unchanged, but for each "standard lab" l_j, we create an "extended lab" l_j^*. Thus, the set of labs is now $\tilde{L} = L \cup L^* = \{l_1, \ldots, l_m, l_1^*, \ldots, l_m^*\}$. In addition, we remove all minimum quotas, and define new maximum quotas $\tilde{q}_l$ for $l \in \tilde{L}$ as follows: if $l \in L$, we set $\tilde{q}_l = p_l$, while if $l^* \in L^*$, we set $\tilde{q}_{l^*} = q_l - p_l$.

For the lab priorities, if $l \in L$, then $\tilde{\succ}_l = \succ_l$; if $l^* \in L^*$, then $\tilde{\succ}_{l^*} = \succ_{ML}$. That is, the standard labs use the priorities from the original market, while all of the extended labs use the ML. For student s, the preferences over $L \cup L^*$ are created by taking the original preference relation $\succ_s$ and inserting lab l_j^* immediately after lab l_j. That is,

preference relation $\succ_s$: $l_j \; l_k \; \cdots$ becomes $\tilde{\succ}_s$: $l_j \; l_j^* \; l_k \; l_k^* \; \cdots$

Finally, no more than $e = n - \sum_{l \in L} p_l$ students can attain seats in extended labs. This restriction ensures that all quotas in the original matching problem will be satisfied.

Example 1. *[ES-DA] There are five students $s_1, \ldots, s_5$ and three labs l_1, l_2, l_3. For each lab, $p_l = 1$ and $q_l = 3$. The preferences and priorities are as follows:*

$$\begin{array}{llll} \succ_{s_1}: & l_1 & l_2 & l_3, \\ \succ_{s_2}: & l_2 & l_1 & l_3, \\ \succ_{s_3}: & l_2 & l_3 & l_1, \\ \succ_{s_4}: & l_2 & l_3 & l_1, \\ \succ_{s_5}: & l_2 & l_1 & l_3. \end{array} \qquad \begin{array}{llllll} \succ_{l_1}: & s_3 & s_5 & s_1 & s_2 & s_4, \\ \succ_{l_2}: & s_1 & s_4 & s_3 & s_5 & s_2, \\ \succ_{l_3}: & s_1 & s_2 & s_4 & s_5 & s_3. \end{array}$$

To run ES-DA, our extended market uses labs $L \cup L^* = \{l_1, l_2, l_3, l_1^*, l_2^*, l_3^*\}$, and maximum quotas $\tilde{q}_l = 1$ for $l \in L$ and $\tilde{q}_{l^*} = 3 - 1 = 2$ for $l^* \in L^*$. Note that there are no minimum quotas in this problem. We additionally modify all students' preferences by inserting lab l_j^* after lab l_j. For example, the modified preferences of student s_1 are as follows: $\tilde{\succ}_{s_1}$: $l_1 \; l_1^* \; l_2 \; l_2^* \; l_3 \; l_3^*$. For the lab priorities, we set $\tilde{\succ}_l = \succ_l$ for $l \in L$, while for $l^* \in L^*$, we set $\tilde{\succ}_{l^*} = \succ_{ML}$.

In round 1 of ES-DA, student s_1 applies to lab l_1 and students $s_2, \ldots s_5$ apply to lab l_2. Labs l_1 and l_2 tentatively accept s_1 and s_4, respectively. Lab l_2 rejects s_2, s_3 and s_5. In round 2, students s_2, s_3 and s_5 apply to l_2^*. Since only $e = 2$ students can be assigned to extended labs at the final matching, student s_5 is rejected. At the end, the following matching is produced:

$$l_1^* - \{s_1\}, \; l_2^* - \{s_2\}, \; l_3^* - \emptyset, l_1 - \{s_5\}, \; l_2 - \{s_4\}, \; l_3 - \{s_3\}.$$

Mapping this back to a matching in the original model:

$$l_1 - \{s_1, s_5\}, \; l_2 - \{s_2, s_4\}, \; l_3 - \{s_3\}.$$

The mechanism satisfies strategy-proofness and elimination of all strong justified envy. Furthermore, this idea can be applied to TTC and we obtain ES-TTC that is strategy-proof and efficient.

4. MULTI-STAGE MECHANISMS

The MS mechanisms proceed slightly differently. For these mechanisms, we first "reserve" a number of students equal to the sum of the minimum quotas across all labs. Then, we run the standard DA or TTC on the remaining set of students. This procedure is then repeated until all students are assigned. Because at each stage we reserve a number of students equal to the sum of the minimum quotas remaining, at the end of the mechanism, all minimum quotas will be satisfied. The MS mechanisms inherit the desirable properties from the ES mechanisms, i.e., MS-DA is strategy-proof and eliminates all strong justified envy, while MS-TTC is strategy-proof and efficient.

A tradeoff exists between the classes of ES and MS mechanisms, depending on the size of the minimum quotas. We empirically show that when the minimum quotas are small, the ES mechanisms tend to create many traditional blocking pairs compared to MS. When the minimum quotas are large, the reverse happens. Policymakers may find it advantageous to use the MS (ES) mechanisms when the minimum quotas are small (large).

5. REFERENCES

[1] A. Abdulkadiroglu and T. Sonmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

[2] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[3] K. Hamada, K. Iwama, and S. Miyazaki. The hospitals/residents problem with quota lower bounds. In *19th Annual European Symposium on Algorithms*, pages 180–191, 2011.

[4] L. S. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1:23–28, 1974.

[5] M. U. Unver. Dynamic kidney exchange. *Review of Economic Studies*, 77(1):372–414, 01 2010.

Incentives for Truthful Reporting in Crowdsourcing (Extended Abstract)

Ece Kamar
Microsoft Research
Redmond, WA 98052
eckamar@microsoft.com

Eric Horvitz
Microsoft Research
Redmond, WA 98052
horvitz@microsoft.com

Categories and Subject Descriptors

I.2 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Design, Economics

Keywords

crowdsourcing systems, peer-prediction rules

1. INTRODUCTION

A challenge with the programmatic access of human talent via crowdsourcing platforms is the specification of incentives and the checking of the quality of contributions. Methodologies for checking quality include providing a payment if the work is approved by the task owner and hiring additional workers to evaluate contributors' work. Both of these approaches place a burden on people and on the organizations commissioning tasks, and may be susceptible to manipulation by workers and task owners. Moreover, neither a task owner nor the task market may know the task well enough to be able to evaluate worker reports. Methodologies for incentivizing workers without external quality checking include rewards based on agreement with a peer worker or with the final output of the system. These approaches are vulnerable to strategic manipulations by workers. Recent experiments on Mechanical Turk have demonstrated the negative influence of manipulations by workers and task owners on crowdsourcing systems [3]. We address this central challenge by introducing incentive mechanisms that promote truthful reporting in crowdsourcing and discourage manipulation by workers and task owners without introducing additional overhead.

We focus on a large class of crowdsourcing tasks that we refer to as *consensus tasks*. Consensus tasks are aimed at determining a single correct answer or a set of correct answers to a question or challenge based on reports collected from workers. These tasks include numerous applications where multiple reports collected from people are used to make decisions. We adapt the *peer prediction rule* [4] to formulate a payment rule that incentivizes workers to contribute to consensus tasks truthfully in crowdsourcing. The rule pays workers depending on how well their report helps to predict another worker's report for the same task. To address several shortcomings of the peer prediction rule, we introduce a novel payment rule, called the *consensus prediction rule*. This payment rule couples payment computations with planning to generate a robust signal for evaluating worker reports. The consensus prediction rule rewards a worker based on how well her report can predict the consensus of other workers. It incentivizes truthful reporting, while providing better fairness than peer prediction rules.

A more detailed presentation of the ideas investigated in this work, including a comparison with existing payment rules, an investigation of considerations in applying payment rules in real-world applications, and a detailed empirical evaluation can be found in [2].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. SOLVING CONSENSUS TASKS

A task is a consensus task if it has a correct answer, has access to a population of workers who are able to share assessments about the correct answer, and where a worker's inference is stochastically relevant to the assessment of a randomly selected worker. The goal of a consensus-centric crowdsourcing system is to deduce an accurate prediction of the correct answer of a task by making use of multiple worker reports.

Let us assume that a crowdsourcing system has access to inferential models that can be used to predict the correct answer, to make hiring decisions, and to calculate payments. These models include an *answer model* (M_A) and a *report model* (M_R). $M_A(a, f)$ is the probability of the correct answer being a given the feature set of the task (f). $M_R(r_i, a^*, f_i)$ is the probability of worker i reporting r_i, given that the correct answer of the task is a^* and the set of features relevant to the worker report is f_i. At each point during execution, the system makes a decision about whether to hire a worker randomly from the worker population, or to terminate the task. When the system decides to not hire additional workers, it provides a final consensus answer $\hat{a}$ based on aggregated worker reports and delivers this answer to the owner of the task. Let π be the policy for making hiring decisions. We define a function M_π such that for a given sequence of worker reports r and feature set f, $M_\pi(r, f)$ is $\emptyset$ if π does not terminate after receiving r, and is $\hat{a}$, the consensus answer, otherwise.

Detailed investigations of learning answer and report models and policies for consensus tasks have been presented separately [1].

3. PAYMENT RULES

We now present payment rules that ensure that truthful reporting is an equilibrium of a consensus task. We start by presenting definitions and assumptions that are needed to formalize payment rules for consensus tasks.

In consensus tasks, workers report on a task once and maximize their individual utilities for the current task. We make the assumptions that the probability assessments performed by models M_A and M_R are accurate and common knowledge. $\tau_i(r_i, r_{-i}) \rightarrow \bar{\mathbb{R}}$ denotes the system's payment to worker i, based on r_i, worker i's report, and r_{-i}, a sequence of reports collected for the same task excluding r_i. Ω_R is the domain of worker inferences and reports. Let s_i^t be a reporting strategy of worker i such that for all possible inferences c_i she can make for task t, $s_i^t(c_i \in \Omega_R) \rightarrow r_i \in \Omega_R$. A strategy s_i^t is truth-revealing if for all $c_i \in \Omega_R$, $s_i^t(c_i) = c_i$. $\mathcal{M} = (t, \pi, \tau)$, mechanism for task t with policy π and payment rule τ, is strict Bayesian-Nash incentive compatible if truth-revelation is a strict Bayesian-Nash equilibrium of the task setting induced by the mechanism.

S is a proper scoring rule for the forecast of a categorical random variable with domain Ω. It takes as input p, probability vector over Ω, and $\omega_i \in \Omega$, the realized outcome of the variable, and it outputs a reward in $\bar{\mathbb{R}}$. The expected reward is maximized if the reported forecast agrees with the true forecast.

3.1 Peer Prediction Rule

The *peer prediction rule* is an adaptation of the rule proposed by Miller et. al. to the domain of crowdsourcing. It rewards a worker based on how well her report can predict the report of another worker.

PROPOSITION 1. *For a given consensus task t and policy π, let r_j be the report of a random worker from I_{-i}. $M = (t, \pi, \tau^p)$ is strict Bayesian-Nash incentive compatible, where worker i's payment, τ_i^p, for reporting to task t is,*

$$\tau_i^p(r_i, r_j) = S(p^p, r_j), \textit{ where}$$
$$\textit{for all } r_k \in \Omega_R,\ p_k^p = Pr_f(C_j = r_k | C_i = r_i)$$

In the equilibrium when all workers report their true inference, $Pr_f(C_j|C_i)$ can be computed by applying Bayes rule and by making use of answer and report models presented in Section 2.

$$Pr_f(C_j = r_j | C_i = r_i) = \frac{\sum_{a \in A} M_A(a, f) M_R(r_i, a, f_i) M_R(r_j, a, f_j)}{\sum_{a \in A} M_A(a, f) M_R(r_i, a, f_i)}$$

3.2 Consensus Prediction Rule

Despite its incentive compatibility properties, the peer prediction payments may not be fair in the way it rewards workers. A worker reporting correctly may receive a low payment if paired with a worker reporting incorrectly. We now present a novel incentive compatible payment rule that provides higher levels of fairness. The *consensus prediction rule* rewards a worker according to how well her report can predict the outcome of the system (i.e., the consensus answer that will be decided by the system), *if she were not participating*. This payment rule forms a direct link between a worker's payment and the outcome of this system. Because the outcome of a successful system is more robust to erroneous reports than the signal used in peer prediction rules, this payment rule has better fairness properties.

Let $\hat{A}_{-i}$ be a random variable for the consensus answer decided by the system if the system runs without access to worker i. An inference of a worker provides evidence about the task, its correct answer, and other workers' inferences, which are used to predict a value for $\hat{A}_{-i}$. Thus, it is realistic to assume that a worker's inference is stochastically relevant for $\hat{A}_{-i}$, given feature set f.

PROPOSITION 2. *For a given consensus task t and policy π, let $\hat{a}_{-i}$ be the consensus answer predicted based on r_{-i}. $M = (t, \pi, \tau^c)$ is strict Bayesian-Nash incentive compatible for any worker i, where*

$$\tau_i^c(r_i, r_{-i}) = S(p^c, \hat{a}_{-i}), \textit{ where}$$
$$\textit{for all } a_k \in A,\ p_k^c = Pr_f(\hat{A}_{-i} = a_k | C_i = r_i)$$

Next, we demonstrate how payments can be calculated with the consensus prediction rule for consensus tasks in the equilibrium when all workers report their true inferences. The calculation of τ_i^c payments is a two-step process; generating a forecast about $\hat{A}_{-i}$ based on worker i's report, and calculating a value for $\hat{a}_{-i}$ based on r_{-i}.

To generate a forecast for $\hat{A}_{-i}$, we simulate consensus system for $L_\emptyset$, the set of all possible sequences of worker reports that reach a consensus about the correct answer.

$$Pr_f(\hat{A}_{-i} = a | C_i = r_i) = \sum_{r' \in L_\emptyset} Pr_f(r'|r_i)\, \mathbf{1}_{\{a\}}(M_\pi(r', f))$$

$$Pr_f(r'|r_i) \propto \sum_{a^* \in A} M_A(a^*, f)\, M_R(r_i, a^*, f_i) \prod_{r_l \in r'} M_R(r_l, a^*, f_l)$$

The second step of τ_i^c calculation is predicting the realized value for $\hat{A}_{-i}$ based on r_{-i}, the actual set of reports collected from workers excluding worker i. Doing so requires simulating $L_{r_{-i}}$, the set of all report sequences that start with r_{-i} and reach a consensus on the correct answer as follows:

$$\hat{a}_{-i} = \operatorname*{argmax}_{a \in A} \sum_{r' \in L_{r_{-i}}} Pr_f(r'|r_{-i})\, \mathbf{1}_{\{a\}}(M_\pi(r', f))$$

4. FUTURE WORK AND CONCLUSIONS

We presented an approach to developing truthful and fair incentive mechanisms for crowdsourcing. Future work includes exploring new approaches for relaxing assumptions of common knowledge and designing truthful incentive mechanisms for a larger variety of tasks. We believe that the use of truthful and fair mechanisms promises to enhance the operation of crowdsourcing for both task authors and contributors, and can promote the wider use of such systems as a trusted methodology for problem solving.

5. REFERENCES

[1] E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS*, 2012.

[2] E. Kamar and E. Horvitz. Incentives and truthful reporting in consensus-centric crowdsourcing. Technical report, MSR-TR-2012-16, Microsoft Research, 2012.

[3] A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *SIGCHI*, 2008.

[4] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, pages 1359–1373, 2005.

Cooperation among Malicious Agents: A General Quantitative Congestion Game Framework (Extended Abstract)

Zaojie Rui
School of Computer Science and Engineering, Southeast University, Nanjing 211189, China;
Research Center for Learning Science, Southeast University, Nanjing 210096, China
culturejie@gmail.com

Tuanjie Fu, Darong Lai
School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
tuanjiefu@gmail.com
daronglai@seu.edu.cn

Yichuan Jiang*
School of Computer Science and Engineering, Southeast University, Nanjing 211189, China;
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
yjiang@seu.edu.cn

ABSTRACT
Malicious behaviors and cooperation have been well studied separately. However, rare systematic study has been conducted on the combination of them: malicious cooperation. In this paper, a general quantitative utility function of malicious cooperation is firstly formulated in a congestion game framework. Both objective and subjective factors are incorporated (e.g., malicious social networks and moral degrees). Then, Nash equilibrium and the condition of malicious cooperation are given theoretically. Meanwhile, we show empirically that malicious cooperation may even improve system performance (i.e., *catfish effect*).

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence−*Multiagent Systems*

General Terms
Theory, Performance, Experimentation

Keywords
Malicious Cooperation, Malicious Cooperative Agents, Catfish Effect, Congestion Game

1. MOTIVATION
Malicious agents are traditionally regarded as non cooperative. However, they may cooperate with each other to conduct malicious behaviors. For instance, coordinated attacks (e.g. DDoS) in cyberspace are launched via cooperation among hackers. We call this type of cooperation malicious cooperation and these agents malicious cooperative agents (MCA). Although malicious behaviors and cooperation have been well studied in agent society, rare attention is paid to malicious cooperation. Thus, we look into the MCA and the malicious cooperation. We mainly focus on two problems: 1) the attraction of malicious cooperation (utility of the MCA from malicious cooperation) and 2) the effect of malicious cooperation on the system.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4–8 June 2012, Valencia, Spain.

* Corresponding author.

2. MODEL DESCRIPTION
For the sake of generality, we present our model in a modified congestion game framework [1], where malicious cooperation is explicitly incorporated. There are two types of agents in the game, *MCA*, and regular agents (*RA*). We assume that all agents are informed of the structure of road network and cost functions. Also Agents are self-interested and are aimed at minimizing own time delay of the journey. The behaviors of agents are based on that of gossip agents in *Gossip Networks* [2]. Agents will communicate with other agents to exchange load information of roads in order to make a more reliable route. Furthermore, we made a modification to MCA to reflect malicious cooperation. Besides tampering the road congestion information in their own routes to be high congested to cheat others out of their path as in [2], MCA in our model will also cheat others for his cooperators.

3. UTILITY FUNCTION OF MCA
Here, a general utility function of the MCA is proposed in a congestion game framework. In order to imply the social factors, which are often ignored in agent research, the function is based on that of in social study [3]. Also, we add the formulation of malicious cooperation to overcome its absence in [3].

***Network and Malicious Effort*:** The adjacency matrix G of network **g** denotes the direct connections of the social network among the MCA. Meanwhile, malicious cooperation is among their social connections. We denote cooperation network as P. Then malicious effort (e_i) of a MCA i is defined as the percentage of malicious peers in its social networks, i.e., $e_i = \sum_{j=1}^{n} p_{ij} / \sum_{j=1}^{n} g_{ij}$.

***Benefit*:** The benefits of agent i come from both his own behaviors and his cooperators' malicious behaviors for him. Thus $B_i = a + e_i \sum_{j=1}^{n} g_{ji} b_j$, where $a>0$ is a constant, b_j denotes the ability of agent j to make benefits from its malicious behaviors. Here b_j in a congestion game roots in cheating RA, so $b_j = b\,\gamma\,|RA_j|$, where b denotes the benefit per successful cheating, γ denotes the success possibility of cheating, and $|RA_j|$ denotes the number of encountered RA in the agent's journey.

***Cost*:** MCA's cost includes *i*) the cost of being detected (c_i^{de}) (including the cost of his malicious behaviors for both himself and his cooperators), *ii*) ignoring high congested information from RA in his or his cooperators' routes (c_i^{ig}), *iii*) *moral cost* for malicious behaviors and *conformity cost* of failing to conform to friends as indicated in [3]. So the sum cost of agent i is $C_i = \sum_{j=1}^{n} p_{ij} f |RA_i| + ce_i^2 + d(e_i - \overline{e_i})^2$, where α, β denotes the possibility of being detected and the false information turning to be true, respectively; *c, d* denotes *moral degree* and *conformity degree*, respectively; $\overline{e_i}$ denotes average malicious efforts of friends, and $f = c^{de}\alpha + c^{ig}\beta$.

Utility Function: Finally, utility function of agent i is:

$$U_i = a + e_i \sum_{j=1}^{n} g_{ij}(b_j - f|RA_i|) - ce_i^2 - d(e_i - \overline{e_i})^2 . \quad (1)$$

4. PROPERTIES OF MCA

Proposition 1 (Nash Equilibrium): Consider the general case when 1) all MCA have different ability to make benefits, 2) social network among MCA are heterogeneous, 3) agents' *conformity degrees* are different. Assume $b_j > f\,|RA_i|$. Then a unique Nash equilibrium in pure strategies of the game is:

$$e_i^* = d\overline{e_i} / c + d + \sum_{j=1}^{n} g_{ij}(b_j - f\,|\,RA_i\,|) / 2(c+d). \quad (2)$$

Proof 1. The utility function is nearly the same as the one in [3], with (b_i-pf) replaced by $\sum_{j=1}^{n} g_{ij}(b_j - f\,|\,RA_i\,|)$. Then we can apply $\alpha = \sum_{j=1}^{n} g_{ij}(b_j - f\,|\,RA_i\,|)$ into the proof of *Proposition 2* in [3]. The assumption of $b_i - pf > 0$ in our case is $\sum_{j=1}^{n} g_{ij}(b_j - f\,|\,RA_i\,|) > 0$. It is always satisfied since $b_j > f\,|RA_i|$. □

Proposition 2 (Condition of Malicious Cooperation): Assume $|RA_i| \sim N(u,\sigma^2)$, agents interact in equal possibility and c=d=0, then the condition that MCA trend to contribute more malicious efforts is: $b / f > u$.

Proof 2. Under the conditions, the utility function turns to be $U_i = a + e_i \sum_{j=1}^{n} g_{ij}(b\gamma\,|\,RA_j\,| - f|RA_i|)$. Thus $E(U_i) = a + e_i u \sum_{j=1}^{n} g_{ij}(b\gamma - f)$. Then we get $\partial E(U_i)/\partial e_i = u \sum_{j=1}^{n} g_{ij}(b\gamma - f)$. If $b\gamma - f > 0$, then MCA will trend to make more benefits if they contribute more malicious efforts. As agents interact in equal possibility, $\gamma = 1/u$. Thus $b\,\gamma - f > 0$ turns to be $b / f > u$. □

Note that ***Proposition 2*** is consistent with the conclusion in a recent *Nature* letter [4], which demonstrates an extraordinary simple rule that cooperators are advantageous over defectors if the benefit of the cooperative act (b), divided by the cost (c), is larger than the average number of neighbors (k), i.e., $b/c > k$. Their c and k are f and u in our model, respectively.

5. EXPERIMENTAL RESULTS

Based on [2], the network simulates a city center which consists of 162 edges and 100 junctions. Each simulation consists of ten iterations. Two hundred agents are randomly generated and they will finish at least 20 journeys from their sources to targets in the iteration. And each journey is tagged with different source-target pair. MCA are initialed with a social network. And each MCA is initialized with a malicious effort ([0, 1]). Similar to [2], the attraction of malicious cooperation in simulations is evaluated when the detection is absent. And the utility of agents in simulations are denoted by time delay per kilometer (*TDK*). A smaller *TDK* indicates a higher utility. Noting that moral cost and peers effect have been validated in [3], we just focus on second term of Equation 1. In below, n denotes the number of MCA.

Validation of Utility Function: Two set of simulations are done to test the effect of characteristics of social networks and malicious efforts.

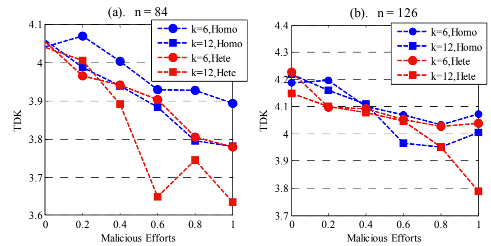


Figure 1. Effect of Social Networks and Malicious Efforts

Figure 1 illustrates that utility increases as MCA contributes more efforts (e_i), which can be explained by the increase of second term of Equation 1. Also, compared to homogeneous social networks, heterogeneous networks can improve the utility of MCA.

Catfish Effect: Here, we will show the effect of malicious cooperation on RA and the system. We find a counterintuitive phenomenon, denominated as "*catfish effect*": malicious cooperation may benefit the system and even RA, comparing to the situation where malicious cooperation is absent (i.e., $e_i = 0$).

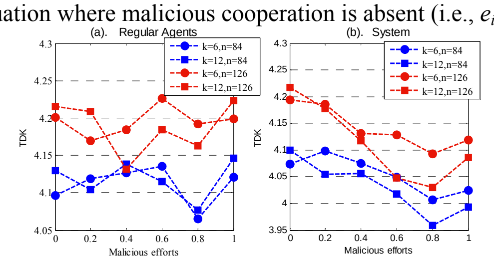


Figure 2 Effect of Malicious Cooperation on RA and System

Figure 2 illustrates that there exists an appropriate malicious effort, i.e., *catfish effort*, where *TDK* is minimal for RA or system. It implies that a catfish effect exists if e_i is appropriate. A potential reason for this is that false information helps RA avoid encountering MCA, which may avoid high congested roads to some extent.

6. CONCLUSION AND FUTURE WORK

We have presented a general quantitative congestion game framework for analyzing cooperation among malicious agents. We show theoretically the Nash equilibrium of malicious efforts and condition of malicious cooperation. Also, we show the catfish effect via simulations. Our future work may study the malicious cooperation in other evaluation scenarios. Furthermore, it is interesting to find a way to utilize catfish effect to benefit system.

Acknowledgements

This work is supported by projects NSFC (61170164, 60803060), NCET (NCET-09-0289) and SRFDP (20090092110048).

7. REFERENCES

[1] I. Ashlagi, D. Monderer, and M. Tennenholtz. Two-terminal routing games with unknown active players. *Artificial Intelligence*, 173(15): 1441-1455, 2009.

[2] R. Lin, S. Kraus, and Y. Shavitt. On the benefits of cheating by self-interested agents in vehicular networks. *In AAMAS*, 2007.

[3] E. Patacchini, and Y. Zenou. Juvenile delinquency and conformism. *Journal of Law, Economics, and Organization*, 2009.

[4] H. Ohtsuki, et al. A simple rule for the evolution of cooperation on graphs and social networks. *Nature,* 441(7092): 502-505, 2006.

Opinion Convergence in Agent Networks (Extended Abstract)

Sreerupa Chatterjee
University of Tulsa
src569@utulsa.edu

Alexander Ruff
University of Tulsa
alexander-ruff@utulsa.edu

Sandip Sen
University of Tulsa
sandip-sen@utulsa.edu

ABSTRACT

We empirically investigated the dynamics of opinion adaption on random networks, scale-free networks and regular lattice structures where agents adopt the opinion held by the majority of their direct neighbors only if the fraction of these exceed a certain laggard threshold [1]. We observed that either due to initial random distribution of opinion to agents or through opinion adaptation in the first few iterations, isolated pockets of agents with a different opinion than those of the surrounding population form and are sustained. Such population configurations thereafter converge to mixed or heterogeneous states. For certain values of the laggard threshold, we also observe a phase of uncertain convergence: for identical system parameters, the population will converge to homogeneous opinions whose value may be different for different random initializations. We identify the regions of consistent homogeneous convergence, heterogeneous convergence and uncertain homogeneous convergence for different values of the laggard threshold.

Categories and Subject Descriptors

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

General Terms

Management, Performance, Reliability

Keywords

Agent Networks, Opinion Formation

1. THE MODEL

Each agent i in our model represents a node in a network whose state represents its opinion on the topic of interest. We consider only binary opinions. Linked nodes are in contact with each other and know each other's opinions. The opinion formation process of node i, initially in state 0(1), is a three step process [2]:

- The state of all the neighbouring nodes to i are checked.
- If the fraction of state 1(0) nodes of i's neighbours exceeds a threshold p_u , i adopts opinion 1(0).
- Otherwise i remains in the same state 0(1).

We define *mixed-convergent graph* as one in which no agent changes its state from the previous time step but the system has not reached total consensus, i.e., all agents do not share the same opinion. Given a graph $G = (V, E)$ where agents represent nodes and edges represent neighborhood relationships, a convergent graph is reached when:

$$\forall_{v \in V} |\text{Adj}_v \cap \text{St}_{\neg v}| \leq P_u \cdot |\text{Adj}_v|,$$

where Adj_v is the set of neighbors to v and St_v ($\text{St}_{\neg v}$) is the set of nodes with the same (or opposite) state as v.

We define a *stationary configuration* to be a subset of the nodes such that they all have the same state and none of these nodes will ever change state irrespective of the state changes outside of this subset. The most stringent conditions for characterizing stationary configurations can be derived by assuming the worst case scenario of all nodes outside of the configuration adopting the opposite state. Thus a stationary configuration consists of a set of agents (nodes) $S \subset V$ such that

$$\forall_{v \in S} |\text{Adj}_v \cap S| \geq (1 - P_u) \cdot |\text{Adj}_v|.$$

To simplify our analysis for idenitfying stationary configurations, we consider only d-regular graphs[1] and choose $P_u = \frac{d-2}{d}$. We empirically evaluate 4-regular graphs and a special case of 4-regular graphs, the toroidal grid. The corresponding laggard threshold is $P_u = \frac{4-2}{4} = 0.5$. We assume that the likelihood of existence of the smallest cycles significantly outweighs the likelihood of existence of larger ones. For general 4-regular graphs the smallest cycles would be 3-cycles and for the toroidal grid, where there are no 3-cycles, it is 4-cycles or 2x2 squares.

We assume initial node opinions are randomly distributed. Even then, some groups of nodes may form stationary configurations at the outset, surrounded by nodes of opposite opinion. Or they may settle into a stationary configuration after one or few iterations and get stuck there forever.

We now calculate the probabilities of the occurrence of such stationary configurations. To simplify the computational complexity of computing the probabilities, we have assumed that the probability of each node to have neighbours in stationary states is independent of each other.

[1] A d-regular graph is graph where all nodes have degree d.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

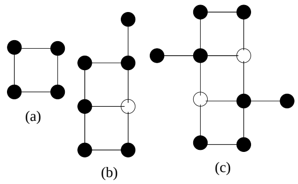


Figure 1: Some stationary configurations for toroidal grids: (a) basic stationary square configuration, (b) and (c) shows two configurations that can lead to the basic configuration in one iteration.

In Figure 1(a) we compute the probability of each of the nodes in the square to be a part of this stable configuration as the product of the probabilities of each of its neighbours to be in the stationary state (here, '1') ie, $(1-a_0)^2$, where a_0 is the initial percentage of agents of opinion 0 in the population. Hence the probability of having one such stationary square is $\mathcal{P} = ((1-a_0)^2)^4$. The maximum number of such stationary square configurations possible in a network of N nodes is $N/4$. Therefore the probability of having at least one such stationary square in the grid is $\mathcal{P}_1 = 1-(1-\mathcal{P})^{N/4}$. We can similarly calulate the probability of having a stationary configuration as in Figure 1(b) in the grid as $\mathcal{P} = a_0^3(1-a_0)^{15}$. So the probability of having at least one such configuration is again $\mathcal{P}_2 = 8(1-(1-\mathcal{P})^{N/4})$ considering 8 possible orientations of the stationary state in the grid. Similarly, for Figure 1 (c) probability of having at least one such configuration is $\mathcal{P}_3 = 4(1-(1-\mathcal{P})^{N/4})$ considering four different orientations of the configuration in the grid where $\mathcal{P} = a_0^6(1-a_0)^{17}$.Similarly we can consider cycles bigger than size 4 stuck in a particular opinion with nodes of opposite opinion filling up the whole interior of it. Hence the probabilty of having atleast one cycle of dimention sxs is $\mathcal{P}_{cycle_s} = 2(1-(1-\mathcal{P})^{N/s^2})$ where, $\mathcal{P} = (1-a_0)^2$. Now s can vary from 3 to M for a MXM grid at an increment of 2. Hence the overall probability of having such stationary configurations stuck in opinion '1' in the whole grid is $Prob('1') = \mathcal{P}_1 + \mathcal{P}_2 + \mathcal{P}_3 + \mathcal{P}_4 + \mathcal{P}_{line} + \mathcal{P}_{cycle}$ assuming independent cases. Similarly we can compute $Prob('0')$ for stationary configurations stuck in state '0'. Hence the percentage of runs where we have at least one such stationary state stuck at either state '0' or '1', i.e., where mixed convergence occurs is $Prob('0') \times Prob('1')$.

We performed similar analysis for random graphs, where we considered cycles of 3 nodes to be the simplest and most frequent stationary configuration. We do not include the corresponding expressions due to space constraints.

2. EMPIRICAL RESULTS

Because of our simplifying assumptions for computing the probabilities of stationary configurations, it is worthwhile to evaluate the accuracy of our analytical predictions using simulations. We have simulated opinion evolution in toroidal grids varying the total number of nodes from 100 to 900 , where the connectivity for each node is 4 and the laggard threshold $P_u = 0.5$.

We experimentally studied the regions of 1-convergence and 0-convergence in grid as well as in random networks for various values of a_0. For extreme values of a_0 consensus is always achieved, but for intermediate values, the network

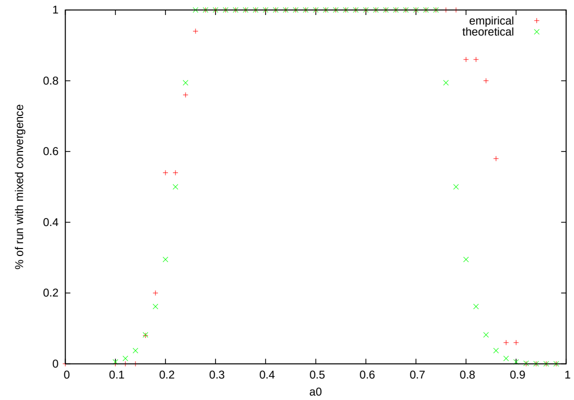


Figure 2: % runs with mixed convergence (toroidal grid: $N = 900$, $P_u = 0.5$, $\bar{k} = 4$).

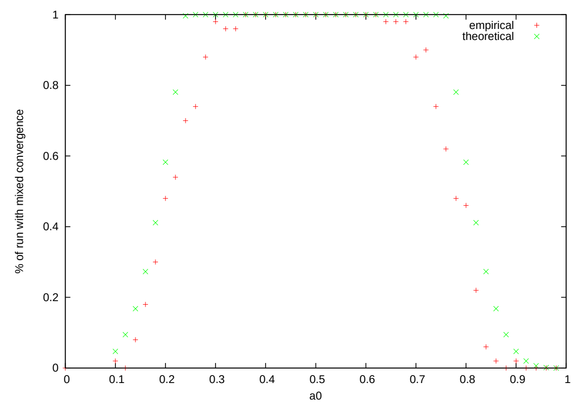


Figure 3: % runs with mixed convergence (random graph: N = 1000 , Pu = 0:5,ŕk = 4).

reaches mixed convergence. From both Figures 2 and 3 we observe that our analytical predictions closely match the empirical results despite the simplifying assumptions made to ease calculations.

An interesting observation from experimental data for toroidal grids was that as we increased the the number of nodes (N) in the grid for a given $\bar{k} = 4$, the region for mixed convergence became wider.

3. CONCLUSION

We studied the problem of opinion convergence in a society of agents situated in a fixed topological structure and identify stable subgraph configurations that will produce mixed convergence and calculated approximate probabilities for the same. Our analytical predictions approximate matched data from simulations for 4-regular graphs. We want to expand our model to cover a wider range of graphs.

4. REFERENCES

[1] P. Krapivsky and S. Redner. Dynamics of majority rule in two-state interacting spin systems. *Physical Review Letter*, 90(23):238701, 13 JUNE 2003.

[2] P.Klimek, R.Lambiotte, and S.Thurner. Opinion formation in laggard societies. *Europhysics Letters*, 82(2):1–5, February 2008.

Behavior Modeling From Learning Agents: Sensitivity to Objective Function Details

(Extended Abstract)

Robert Junges
Örebro University
70182 Örebro, Sweden
robert.junges@oru.se

Franziska Klügl
Örebro University
70182 Örebro, Sweden
franziska.klugl@oru.se

ABSTRACT

The process of finding the appropriate agent behavior is a cumbersome task – no matter whether it is for agent-based software or simulation models. Machine Learning can help by generating partial or preliminary versions of the agent low-level behavior. However, for actually being useful for the human modeler the results should be interpretable, which may require some post-processing step after the actual behavior learning. In this contribution we test the sensitivity of the resulting, interpretable behavior program with respect to parameters and components of the function that describes the intended behavior.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents*

General Terms

Design, Experimentation, Performance

Keywords

Multiagent Simulation, Agent Learning

1. MOTIVATION

The basic idea behind agent-based simulation is that phenomena are generated from simulation of mostly local, low-level actions and interactions of agents. In such a bottom-up approach, a central issue concerns what behaviors the agents must exhibit so that the intended outcome is produced. Currently, dependent on the experience of the modeler, the development of an agent-based simulation may result in a painful trial and error process. The goal is to develop a systematic way of bridging the gap between agent behavior and macro-level outcome.

Our idea is to support the process of designing the agent behavior using self-adaptive agents [2]. A human modeler shall focus on describing the targeted phenomenon as well as the overall simulation settings, including the interfaces between environment and agents.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

A basic assumption hereby is that it is possible to characterize good performance in terms of an objective function, also in scenarios in which the features of the actually necessary behavior program are not exactly known. As in every learning approach, the definition of the objective function essentially drives the generation of the agent behavior. Our application of learning is special as we combine traditional Reinforcement Learning, using the given objective function, with a post processing step – a decision tree learner – in which the behavior model shall be abstracted to a human-readable representation. The central question in our contribution is therefore how robust the final behavior outcome is with respect to small changes in the objective function.

2. LEARNING FOR MODELING

As stated in the introduction, we propose a learning-driven analysis and design approach using self-adaptive agents in the behavior modeling task for simulation models. The actual optimality of the learnt agent control is only one relevant criterion; the interpretability of the outcome by a human is essential.

We start with the definition of an environmental model and a set of sensors and actuators that determine what the agents are able to perceive and manipulate. The second step is the definition of a learning architecture that is apt to connect perceptions and actions of the agent. After that, a reward function, providing feedback to the agents, is defined. The reward has to measure performance, as the agents will use it to explore how the environment reacts to their actions. The resulting behavioral model should then be analyzed by the human modeler for preventing artifacts that come from an improper environmental or reward model, or weak interfaces.

The chosen learning architecture for this contribution combines Reinforcement Learning with Decision Tree learning.

We selected Q-learning for our investigation because it is the simplest reinforcement learning technique directly producing situation-action pairs. However, even in simple scenarios the high number of pairs prevents a designer to oversee the actually learnt behavior. For tackling this readability problem we use a decision tree representation of the implicit behavior of the best situation-action pairs. In this contribution we selected the C4.5 algorithm to generate decision trees, which are a well-suited representation model for decision-making processes [1].

The best situation-action pairs are taken by first excluding those pairs that haven't been tested enough, as the confidence on their expected utility is lower. Then, we select for each situation the action with the highest Q-value, considering only those with non-zero, positive Q-values. The generated decision tree basically accomplishes the lacking abstraction that makes the resulting behavior description transparent for the designer.

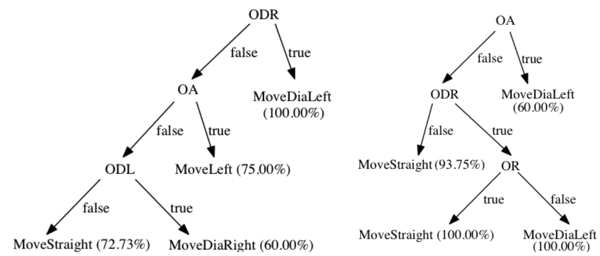
3. EXPERIMENTAL SETUP

Our test scenario is a pedestrian evacuation model. The environment is represented by a room, a number of round obstacles and one exit. The agents are randomly placed in the half on the opposite to the exit. They have two objectives: leave the room as fast as possible and do not collide with obstacles or other agents. Perception is discretized into sectors, actions according to movement directions. A reward is given to each agent individually after each step containing the following components: (1) *Exit Reward* indicating whether the exit was reached; (2) *Collision Reward* punishing collisions; and (3) *Distance Reward* indicating whether the agent came closer to the exit. More details can be found in [3]. For testing the consequences of different setups to the overall outcome we focussed on the relation between the different components indicating the pressure towards/against particular situations: a) when one or more elements are not considered; b) when one elements contributes half or twice as much as the others; c) when all contribute with the same weight.

4. RESULTS AND DISCUSSION

We systematically run experiments with different configurations for the objective function. The agents learnt to sufficiently perform, that means move directly to the exit while avoiding fixed and dynamic obstacles. Only little differences in the performance measured in terms of number of collisions were observable. Naturally, the best performance was measured when the weights of the *Collision Reward* were higher than for *Distance Reward*. After that, using the decision tree learner for generalization, we tested how well the classification result from the decision tree resembles the originally learnt behavior. It turned out that the best preservation of information at the generalization step occurred when the *Distance Reward* was weighted higher than the *Collision Reward*.

Although performance and accuracy measures were almost the same for the different settings, the resulting decision trees were quite different. With a higher *Distance Reward*, we can see that the agent tend to develop actions that lead to shorter paths, at the same time as they try to avoid collisions. A shorter path means the selection of movements that direct the agent to a sector closer to the obstacle. This is different in the case with higher *Collision Reward*: the decision tree points to a selection of perceptions and actions that lead to the development of a wider collision-avoidance path. This comes from the fact that is hard to predict other agents' movements as agents cannot distinguish between pedestrians and columns. If the weight on the collision avoidance is higher, the agents learn to be more "cautious". They take wider deviations from the direct route to avoid eventually colliding with another agent, at the expense of evacuation time. That means finally that those trees are more elaborated than when learning with the other configurations - this can be seen in Figure 1. The codes in the nodes correspond to different perceptions: O for Obstacle, D for Diagonal, A for Ahead, L for Left and R for Right.



(a) Collisions (b) Evacuation

Figure 1: Behavior trees for different objective function details putting more focus on collision avoidance or fast evacuation.

5. CONCLUSION AND FUTURE WORK

Testing the robustness of the finally abstracted behavior output with respect to variations of the initial objective function showed a high sensitivity. Due to the abstraction and generalization done using the decision tree learner, we expected more robustness. However this result shows the relevance of a careful formulation of the criteria for valid agent behavior. Really surprising is that the complexity of the resulting behavior program is not mirrored in the classical numerical metrics that describe learning performance. This makes the formulation of the objective function describing what the agents shall achieve even more critical than describing how they should achieve it.

These results lead to next steps for establishing agent learning as a tool for agent simulation design. First, we have to analyze the learnt behavior more directly by controlling agents using the decision trees generated. This will show whether they actually perform in the intended way or whether too much information has been lost. This may lead us to testing other generalization techniques than the simple decision tree learner that we used. There is a lot of research going on for state abstraction in reinforcement learning. Although not aiming at readability of the abstracted program, they might be applicable in our case. A second future direction of our research directly addresses the formulation of the objective function: Instead of formulating an objective function, we may use learning by demonstration and imitation techniques for directly mapping observable actor or stakeholder behavior to generate a behavior program for the corresponding simulated agent.

6. REFERENCES

[1] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51:141 – 154, 2011.

[2] R. Junges and F. Klügl. Generating inspiration for agent design by reinforcement learning (to appear). *Information and Software Technology*, 2011.

[3] R. Junges and F. Klügl. Programming agent behavior by learning in simulation models (to appear). *Applied Artificial Intelligence*, 2012.

Emergent Behavior of Bacteria in a Multiagent System

(Extended Abstract)

Philip Hendrix*
US Army Engineer R&D Center
Vicksburg, MS 39180

Elena O. Budrene
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139-4307
ebudrene@math.mit.edu

Benoit Morel
Carnegie Mellon
Pittsburgh, PA 15213
bm1v@andrew.cmu.edu

Igor Linkov
US Army Engineer R&D Center
Concord, MA
igor.linkov@usace.army.mil

ABSTRACT

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation, Verification

Keywords

Bacteria, *E. coli*, Emergent Behavior, Clustering

1. INTRODUCTION

Bacteria forming clusters in a Petri dish is a common example of emergent behavior, i.e. many seemingly simple bacteria, when under stress, form clusters through basic rules of interaction. Under suitable conditions, the clusters form spectacular patterns [1]. Those properties of bacteria have inspired a lot of modeling work, attempting to reproduce the patterns and make sense of them.

The goal of the research presented in this abstract is to show that bacteria and the emergent behavior of bacteria can be accurately modeled using a multi-agent framework. The bacteria model was designed to closely reflect the rules of interaction obtained from empirical studies. The bacteria modeled are *Escherichia coli*, commonly referred to as *E. coli*. The novelty of this effort stems from modeling bacteria as bacteria actually behave. Most previous efforts to model the complex patterns of bacteria use partial differential equations (pde's). These pde's are used to produce a pattern as an end result. In contrast, the multi-agent approach allows for a time sequence that shows the formation of the patterns.

Bacteria movement is composed of *runs* and *tumbles*. A *run* is the forward movement of bacteria in a straight line, and a *tumble* is the reorientation of a bacterium so that it may change direction. As the bacteria move, they both excrete and sense aspartate, a chemical attractant. The bacteria change their tumble frequency based on the sensed amount of aspartate in such a way that they move up a chemoattractant gradient, thereby forming into clusters.

We wish to accurately model bacteria, because they are some of the most simple biological organisms. Accurately modeling the behavior of these very simple organisms will provide a strong stepping stone from which research can model more complex organisms, and provide a form of validation for the mapping of simulated experimental results to real world empirical results.

The self-organization of bacteria into clusters for higher resistance to stress has been acquired presumably because it provides an evolutionary advantage to the bacteria. Similar behavior exists in other species, but bacteria may be the simplest organisms displaying such behavior.

The results presented in this abstract show bacteria-agents forming clusters and moving swarm rings in a simulated Petri dish. The work presented in this abstract strives to accurately model bacteria, but does not model all aspects of bacterial behavior. However, the results gained from the approach shown in this abstract offer a significant advancement in the modeling of bacteria, as the results show that it is possible to reproduce the self-organization in clusters of a population of bacteria observed empirically, using what is known of their movement in runs and tumbles.

2. RESULTS AND DISCUSSION

Since an abstract cannot display movies, we show time series images of the simulation. Figure 1 shows the time series of six stages of a clustering experiment over a 10 minute period. Each of the six images displays the bacteria in the Petri dish. The lighter colors within the dish represent higher concentrations of bacteria. The number in the upper-left hand corner of each image displays the time step in the form minutes : seconds. The first image in the figure displays the initial condition at time step 0. The second image shows the bacteria after 2 minutes. In this image we can see that some areas are darker than others, meaning the distribution of bacteria is no longer uniform. After 6 minutes distinct clusters can be seen. The clusters become more pronounced

---

*Philip Hendrix is now at Google in Mountain View, CA, USA, and may be reached at hendrix@google.com.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

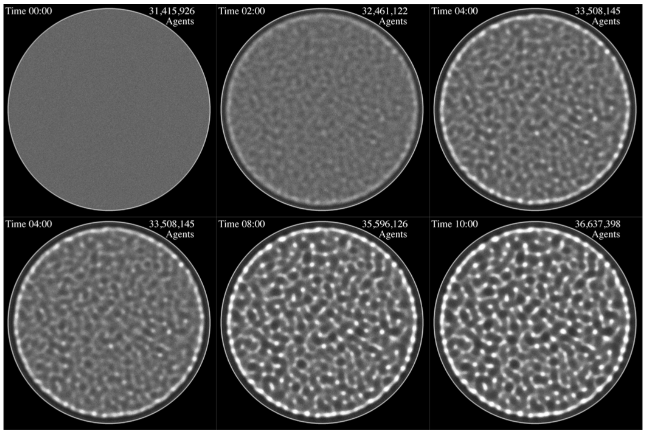


Figure 1: Simulation of *E. coli* bacteria in a $20mm$ Petri dish over a 10 minute time period. Agent density was set to $100,000 Agents/mm^2$. Time step 0 shows the initial setup of bacteria agents with locations and directions randomly sampled from uniform distributions. The beginnings of clustering can be seen at minute 4, and are clearly visible at minute 6. Agents were exposed to trigger levels of $5mM$.

after 8 and 10 minutes. Figure 3 shows a side by side comparison of the simulated bacteria clusters (left) to that of the real world empirical results (right).

The results of the swarm ring are presented in Figure 2. The swarm ring in the upper left hand corner is a snapshot of the Petri dish after 3 hours of simulation time. The remaining five images are also snapshots of the swarm ring and were taken in one hour time increments from left to right form 4 hours to 8 hours.

Using only runs and tumbles for movement the simulated bacteria were able to form clusters similar to that of real-world bacteria. By observing the images of clusters from the simulation results on the left-hand side of Figure 3 to the real-world results shown on the right-hand side of Figure 3 we can compare the results. The real-world result has very dense clusters (in white) and not as dense regions (in gray). Similarly, in the simulation results, we can observe that the bacteria density of the bacteria differ between clusters and other regions. Some clusters appear to be darker (more dense) than others. The simulation can also reproduce the variation in size and distance between the clusters, as they depend on the amount of aspartate/chemoattractant secreted by the bacteria, which is an adjustable parameter in the simulation.

The swarm ring grew at roughly the same pace as the initial growth of the empirical swarm rings provided by Budrene and Berg in [1]. The growth rate of the simulated swarm ring falls between the growth rate of the empirical results with initial succinate amounts of $2mM$ and $3mM$.

These simulations show that it is possible with very few inputs to reproduce important properties of population of bacteria. From the perspective of random walks, the aspartate concentration dependent movement in "runs" and "tumbles" is a case of "biased random walk". Those simulations also shed light on important aspects of the parallel mathematical modeling inspired by the emergence of the clusters. In the related work, to reproduce the patterns from diffusion equations, an aspartate concentration dependent chemotactic term had to be added. These simulations

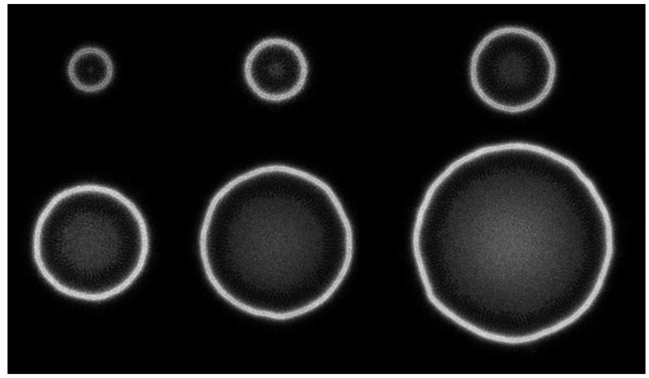


Figure 2: Simulation of *E. coli* bacteria over an 8 hour time period. At time 0, $10,000$ bacteria agents were placed in the middle of the Petri dish. Initial succinate was set to $2mM$. The upper left hand swarm ring is a snapshot of the swarm ring after 3 hours of simulation time. The remaining five images are also snapshots of the swarm ring and are in one hour time increments from left to right from 4 hours to 8 hours.

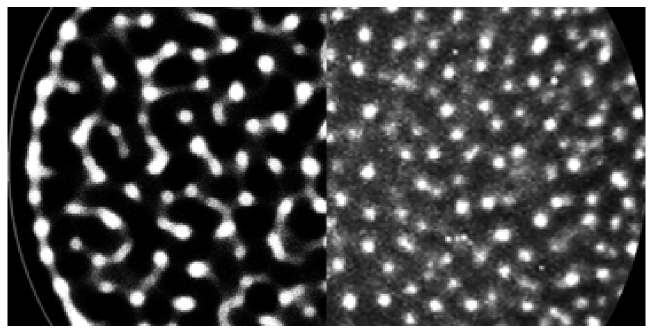


Figure 3: Clustering result of *E. coli* in a $12mm$ Petri dish after 10 minutes.

show that the non linear dependence of the chemotaxis of the bacteria does not conceal any mysterious property not yet known of the bacteria. It is a natural consequence of the way a multi-agent population of bacteria measure concentration gradients: by successive measurements of concentration. The study of the emergence of clusters has triggered a few theoretical questions on the response function of bacteria, among other things bacteria simulations may potentially be used to answer questions using simulations, as it is possible to change the conditions in which the bacteria are put.

3. ACKNOWLEDGMENTS

This work was supported by the Environmental Quality Research Program of the US Army Engineer Research and Development Center. The authors thank E. Ferguson, the manager of this programme. Permission was granted by the USACE Chief of Engineers to publish this material. The views and opinions expressed in this paper are those of the individual authors and not those of the U.S. Army, or other sponsor organizations.

4. REFERENCES

[1] E. Budrene and H. Berg. Dynamics of formation of symmetrical patterns by chemotactic bacteria. *Nature*, 376:49–53, July 1995.

Investigating the Role of Social Behavior in Financial Markets through Agent-Based Simulation

(Extended Abstract)

Alessia Mauri
alessia.mauri@yahoo.it

Andrea G. B. Tettamanzi
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 56, I-26013 Crema (CR), Italy
andrea.tettamanzi@unimi.it

ABSTRACT

An evolutionary agent-based model inspired by the adaptive market hypothesis is used to investigate the link between the microscopic parameter of sentiment and market price movements. Agents model cognitive and social behaviors by means of rules wired into their decision-making models and of parameters encoded in their genome. Results show that co-evolution and social interaction among traders are responsible for bubbles and crashes.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.6.5 [**Simulation and Modeling**]: Model Development; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics

Keywords

Agent-Based Simulation, Emergent Behavior, Co-evolution, Complex Systems

1. INTRODUCTION

Agent-based modeling [7], has become a widely accepted tool for studying the dynamics of financial markets [2].

While usually agent-based models of economies rely on very simple agents, which make them resemble interacting particle systems we wanted to investigate the use of richer, more sophisticated agent types, that more closely reproduce the characters of the participants in real-world economies, namely people. Previous work by the authors [1] added more realism in the way agents are modeled in agent-based simulations of financial markets by using cognitive agents and a real-world auction mechanism as the basic ingredients for the simulation, coupled with an evolutionary algorithm (EA) responsible for the adaptation of agent behaviors. The agent-based model we used for this study is an adaptation of that work.

An important reason to include the concept of evolution in a model that aims at achieving realism is that this idea is perfectly in line with the adaptive market hypothesis [5] (AMH), an attempt to reconcile the efficient market hypothesis with a growing number of criticisms leveled against it by psychologists and behavioral economists. According to the AMH, the markets would be in a perennial unstable equilibrium and, as their participants must evolve against each other, in constant flux.

The purpose of our work was to replicate complex phenomena like price bubbles emerging from the interaction of co-evolving, cognitive agents acting mostly rationally but with possible undertones of irrationality.

2. MARKET STRUCTURE

We used a realistic, albeit simplified, market structure: a single *asset* is traded on the market as a commodity against the payment of *money*. No transaction fees are claimed by the market and no expenses are payable for the storage and transportation of the asset.

Agents participate in the market by submitting buy or sell orders. The orders are matched by the market with a single-price auction every time a new order is received. An agent's *net asset value* (NAV) is the primary indicator of its performance.

Agents operate in a regime of incomplete information, in that every agent knows its current money balance and asset inventory, as well as the orders it submits, but cannot directly observe neither the orders entered by the other market participants, nor the balance and inventory of its peers. Instead, to determine their behavior, the agents have access to a set of technical indicators made available by the market and to the *sentiment* of other agents in their social neighbourhood.

Every given number of periods, a generation of the EA is performed, allowing the agent population to evolve towards more profitable behaviors.

3. AGENTS AND SOCIALITY

The agents model a combination of knowledge, experience, and psychology aimed at giving an outline of those decisional factors that are distinctive of human traders. All the agents share the same architecture, comprising two modules: a decisional module, inspired by neural networks [3], and an

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

operational module, inpired by particle swarm optimization algorithms [4]. The decisional model generates the kind of order to be sent to the market based on beliefs, cognitive inertia, and current money balance and asset inventory. Beliefs are formed based on technical indicators, weighted by each agent according to a genetically determined individual degree of confidence; cognitive inertia is a kind of intention persistence; knowledge of the current money balance and asset inventory causes the agents to target an ideal balance. Agents are part of a social network determined by their genealogy. Each agent takes into account the decisions made by its close relatives, weighted by their genetic similarity and financial success. From such network arises the herd behavior, a self-organised pattern which is actually typical of real trader groups and is expressed by market *sentiment* on a macroscopic level.

Finally, the operational module defines the specifics of the order to be submitted to the market, namely the asset quantity and proposed bid or ask price.

A salient feature of the agents in this model is that they are evolutionary, i.e., they are the individuals of an EA. Therefore, every agent has a *genome*, in practice a set of 43 parameters that influence the way the agent reacts to stimuli from the environment. These genetically determined parameters do not change during the agent's entire lifetime, whereas the agent's mental state may change in response to changes in the market and in the agent's financial conditions.

An EA essentially following the general principles of the one proposed in [1] for the same purpose is used by the simulator to make the agents participating in the market evolve according to their trading proficiency. The EA proceeds concurrently with the trading activity in the market. A generation is performed at regular intervals, whose length, l, measured in trading periods, is a parameter of the simulation. The fitness of an agent is given by its NAV. At each generation, the 30% of the individuals having the lowest NAV is eliminated from the simulation, and their wealth is redistributed to the remaining 70%, which undergoes fitness-proportionate selection, crossover, and mutation. New agents inherit their wealth from both parents.

Social networks are an important base of real economies. Herd behavior is the kind of sociality that was reproduced in our model. It has been argued that this kind of behavior determines bubbles and crashes [6].

Our agents are influenced in the trading decisions by the sentiment of their social the neighbourhood in a way that makes them imitate their successful relatives, with the exception of those that have not made proof of good trading skills. To allow us to study its effects, sociality may be turned on and off.

4. EXPERIMENTS AND RESULTS

To study the relationship between herd behavior and price dynamics, we introduced an exogenous shock to the market, consisting in sudden changes of the interest/dividend paid by the asset or the money. As it can be observed in Figures 1 and 2, sociality emphasizes and amplifies market reactions to shocks.

A bubble grows when returns are distributed on assets in simulations with sociality turned on. On the other side, the bubble crashes when returns are shifted to money and traders loose interest on the assets. This does not happen when sociality is turned off.

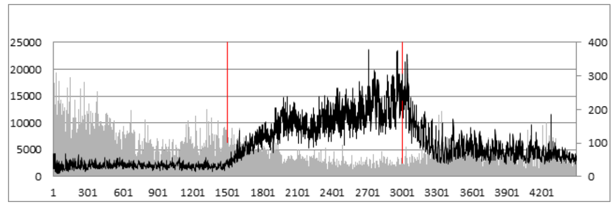


Figure 1: A shock applied to a simulation with sociality turned on.

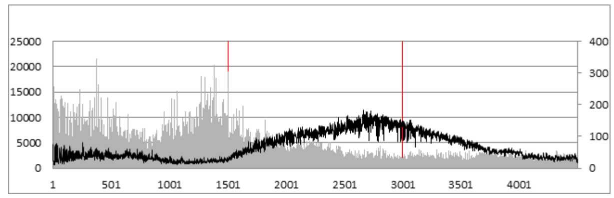


Figure 2: A shock applied to a simulation with sociality turned off.

Through the evolutionary process, only the agents who are able to gain wealth can survive. The evolution leads to a system in which the agents adopt complementary market behaviors keeping the market in an unstable equilibrium. If the EA is turned off, even at a stage when well-adapted trading strategies have emerged, all trading activity wanes and comes to a grinding stop after a few periods. We have observed that agents showing extreme behaviors do not manage to survive and moderate behaviors of the conservative kind tend to become commonplace in a co-evolved population.

5. CONCLUSION

We have studied and analized the effect of social behavior in financial markets by means of an evolutionary agent-based model, finding evidence that herd behavior is responsible for emergent phenomena like bubbles and crashes.

6. REFERENCES

[1] C. da Costa Pereira, A. Mauri, and A. G. B. Tettamanzi. Cognitive-agent-based modeling of a financial market. In *IAT 2009*, pages 20–27, 2009.

[2] N. Ehrentreich. *Agent-Based Modeling*, volume 602 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, 2008.

[3] S. Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, Upper Saddle River, New Jersey, 1998.

[4] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE Int'l Conference on Neural Networks*, pages 1942–1948, 1995.

[5] A. W. Lo. The adaptive markets hypothesis. *The Journal of Portfolio Management*, 30th Anniversary Issue:15–29, 2004.

[6] T. Lux. Herd behavior, bubbles and crashes. *Economic Journal*, 105(431):881–896, 1995.

[7] L. Tesfatsion. Agent-based computational economics: modeling economies as complex adaptive systems. *Information Sciences*, 149(4):263–269, 2003.

An Agent-Based Model for Pedestrian and Group Dynamics: Experimental and Real-World Scenarios

(Extended Abstract)

Giuseppe Vizzari, Lorenza Manenti*
Complex Systems and Artificial Intelligence (CSAI) research center
University of Milano-Bicocca
Viale Sarca 336/14, 20126, Milano
{vizzari, manenti}@disco.unimib.it

ABSTRACT

The paper presents a modeling and simulation effort aimed at studying the implications of the presence of groups of pedestrians in different situations (e.g. changing density, configurations of the environment) in experimental and real world scenarios.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Applications

General Terms

Experimentation

Keywords

pedestrian and crowd modeling, interdisciplinary approaches

1. INTRODUCTION

The agent–based approach to the simulation of complex systems is a relatively recent but extremely successful application area of concepts, abstractions, models defined in the area of autonomous agents and multi–agent systems (MAS). Crowds of pedestrians represent a typical example of complex system: the overall behavior of the system can only be defined in terms of the actions of the individuals that compose it, and the decisions of the individuals are influenced by the previous actions of other pedestrians sharing the same space. Despite the substantial amount of research efforts this area is still quite lively and we are far from a complete understanding of the complex phenomena related to crowds of pedestrians in the environment: one of the least studied and understood aspects of crowds of pedestrians is represented by the implications of the presence of groups [1]. In particular, little work has been done on the modeling and simulation of relatively large groups within a crowd of pedestrians [3].

This work presents the results of an agent–based model of pedestrians considering groups as a first–class abstraction influencing the behaviour of its members and, in turn, of the whole system. The model was tested in a schematic situation, also analyzed by means of experiments, to characterize the implications of groups in the overall pedestrian dynamics and in a real world scenario in which pedestrians were organized in large groups for sake of crowd management.

*Crystals Project, Centre of Research Excellence in Hajj and Omrah (HajjCORE), Umm Al-Qura University, Makkah, Saudi Arabia.

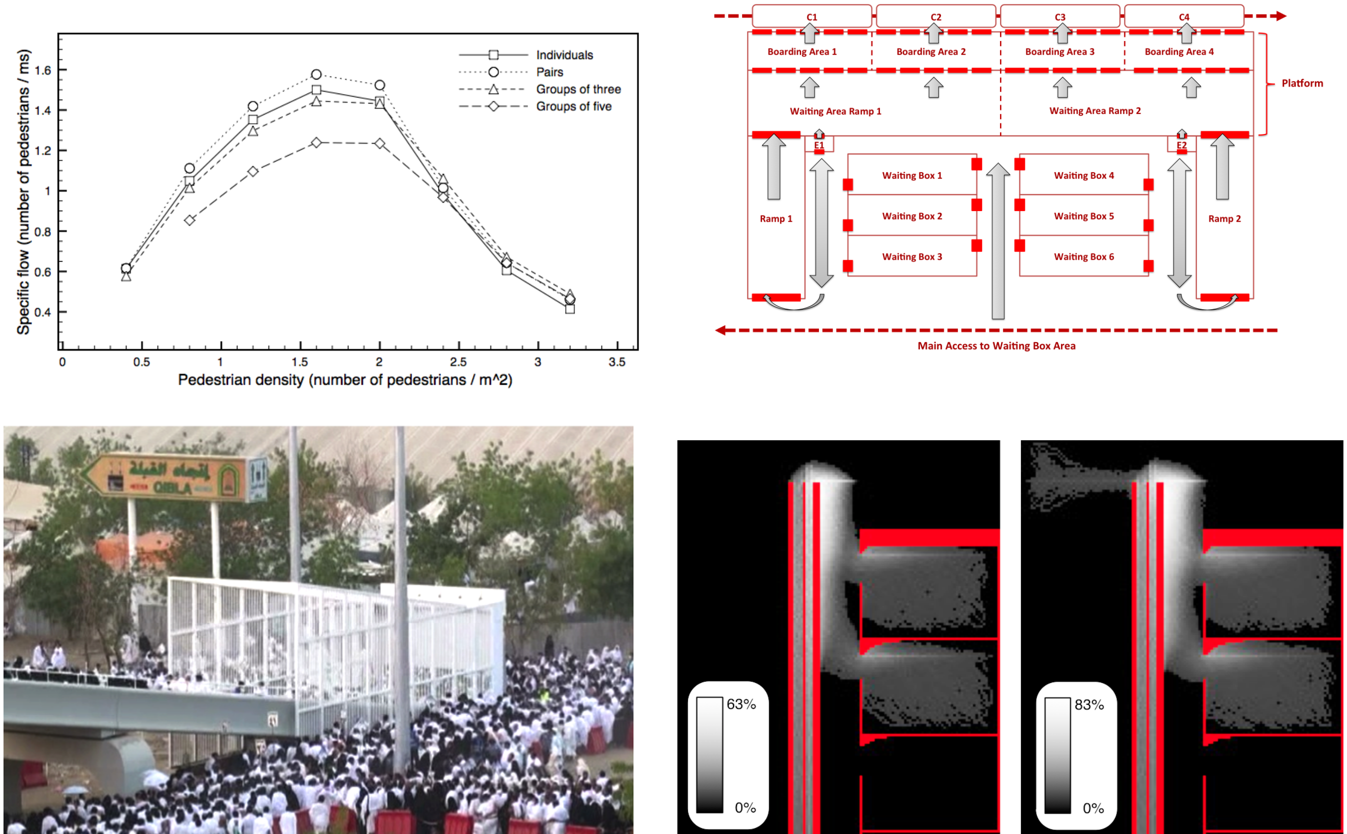
Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. MODEL AND SIMULATION RESULTS

The adopted approach is discrete both in space and in time; the environment in which the simulation takes place is a lattice of cells, each representing a portion of the simulated environment and comprising information about its current state, both in terms of physical occupation and in terms of additional information. To support pedestrian navigation in the environment, the lattice is provided with a static floor field [4] which specifies the shortest path to destinations and targets. Pedestrians have a limited form of autonomy: they can choose were to move according to their perception of the environment and their goal. The choice of the actual movement destination is based on the elaboration of an utility value, called *likability*, representing the desirability of moving into a given position: the presence of members of the group a pedestrian belongs to in a nearby cell is considered positively, unlike the presence of other pedestrians. A complete description of the model can be found in [2].

A set of simulations in an experimental and in a real world situation aimed at evaluating the impact of the presence of groups has been investigated. The figures shown in Figure 1 report an overview on simulation results. The top left part shows a fundamental diagram representing data obtained through several simulations in a simple scenario (a corridor in which pedestrians flow in opposite directions). The results are in tune with the experimental data coming from observations and available in the literature; an original result is the fact that groups of different size have a different ability to flow smoothly in different density situations.

The model was also employed in a real world scenario: a station of the Mashaer line, a newly constructed rail line in the area of Makkah to reduce the congestion caused by the presence of other collective means of pilgrim transportation (i.e. buses) during the Hajj. The simulations were focused on the Arafat I station, in particular in the process lead-



(a) Two waiting boxes - total space utilization

(b) Two waiting boxes and external flow - total space utilization

Figure 1: Images and diagrams related to simulation results.

ing pilgrims from outside the station area to the platforms. The flow is organized adopting waiting-boxes: groups of 250 pilgrims wait in special areas for an authorization by the station agents to move towards the ramps or elevators. The top right part of Figure 1 is a schematic representation of an area of the station with the permitted pedestrian flows; the bottom left photo shows a situation in which the waiting-box principle, preventing the possibility of two flows simultaneously converging to a ramp, was not respected, causing a higher than average congestion around the ramp. Different simulation scenarios were realized to understand the capability of the model to reflect the increase in the waiting times and the space utilization when the waiting box principle was not respected. The bottom right diagrams report the space utilization, i.e., the relative frequency of the cell occupation on the whole simulation time. The second scenario, in which the waiting box principle is not respected, is characterized by a noticeably worse performance not only from the perspective of the size of the area characterized by a medium-high space utilization, but also from the perspective of the highest value of space utilization (83% respect to 63% of the regular scenario). This confirms that increasing the number of pilgrims that are simultaneously allowed to move towards the ramp highly increases the number of cases in which their movement is blocked because of overcrowding. According to these results, the management of the movement of group of pilgrims from the tents area to the ramps should try to avoid exceptions to the waiting box principle as much as possible.

Acknowledgments

This work is a result of the Crystal Project, funded by the Centre of Research Excellence in Hajj and Omrah (Hajj-CORE), Umm Al-Qura University, Makkah, Saudi Arabia, `http://www.csai.disco.unimib.it/CSAI/CRYSTALS/`.

3. REFERENCES

[1] Understanding crowd behaviours: Supporting evidence. `http://www.cabinetoffice.gov.uk/news/understanding-crowd-behaviours`, 2009.

[2] S. Bandini, F. Rubagotti, G. Vizzari, and K. Shimura. An agent model of pedestrian and group dynamics: Experiments on group cohesion. IN *AI*IA*, volume 6934 of *LNCS*, pages 104–116. Springer, 2011.

[3] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4):e10047, 04 2010.

[4] A. Schadschneider, A. Kirchner, and K. Nishinari. CA approach to collective phenomena in pedestrian dynamics. In *ACRI 2002*, volume 2493 of *LNCS*, pages 239–248. Springer, 2002.

The Impact of Cultural Differences on Crowd Dynamics (Extended Abstract)

Natalie Fridman
MAVERICK Group
Computer Science Dept
Bar-Ilan University, Israel
fridman@cs.biu.ac.il

Avishy Zilka
MAVERICK Group
Computer Science Dept
Bar-Ilan University, Israel

Gal A. Kaminka
MAVERICK Group
Computer Science Dept
Bar-Ilan University, Israel
galk@cs.biu.ac.il

1. INTRODUCTION

Accurate models of crowd dynamics are an important challenge for agent-based social simulations. Unfortunately, existing models of physical crowds do not yet account for cultural factors. In this extended abstract, we briefly summarize our results in treating culture as a first-class object in models of physical crowds. Specifically, we examine the impact of cultural differences on crowd dynamics in pedestrian and evacuation domains.

In the pedestrian domain we relate culturally-aware simulation to pedestrian data which we recorded from videos of pedestrians in five different countries: Iraq, Israel, England, Canada, and France. We characterize these cultures along five individual-level parameters: personal spaces, speed, avoidance side and group formations. We use established crowd-level quantitative measures (e.g., flow, number of collisions, and mean speed) to identify crowd-level effects. We also show that the model can faithfully replicate the observed pedestrian behavior in these videos.

In the evacuation domain, we examine individual cultural parameters (documented in social science literature) as to how seriously people treat possible threats, their tendency to notify others, and their tendency to form groups. We then use the simulations to explore the impact of these on the resulting crowd behavior (measured quantitatively in evacuation time, panic levels, etc.).

2. BACKGROUND

In social psychology there is an extensive research on the cultural differences in micro level interactions among groups of people, but it only rarely addresses the effect of these differences on macro-level crowd phenomena (e.g., pedestrian flow). Cultural differences have been found in variety of human behaviors such as in different pedestrian dynamics [5, 6, 8], evacuation behavior [1] and more.

Work on computer modeling of collective behavior has been carried out in other branches of science, in particular for modeling and simulation. Researchers are developing computational models for simulation of collective behavior in order to be able to predict the resulting macro level behavior from micro level interactions [3, 7, 10]. However, to the best of our knowledge, existing computational models for crowd behaviors do not yet account for cultural differences.

This work follows on our earlier work on modeling pedestrians, validated against human crowd data [4]. It also builds on our work on the ESCAPES [11] agent-based simulation. ESCAPES is a an evacuation simulation, incorporating for individual and social factors. Neither of these earlier works accounted for cultural differences.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

3. PEDESTRIAN CULTURES

In modeling pedestrians, and based on the literature, we focus on the following individual (micro-level) cultural parameters: personal space ([2, 5, 6]), base walking speed ([8], avoidance side ([9]) and group formations (in particular gender-heterogeneity, size, and shape [10]). We quantitatively measure these in movies taken in five different cultures: Iraq, Israel, England, Canada and France. Overall, we collected over a hundred hours of pedestrian footage in different locations. For the purposes of the analysis, we used a total of 45 minutes. Each movie was analyzed by two different subjects and we used the mean value for each measure in our results. For lack of space we only report here that indeed the five countries differ from each other in these four cultural parameters.

We used an agent-based simulation to examine the effect of the presented above four cultural factors on macro-level pedestrian dynamics. We ran extensive simulations with the above values, totaling over 100 hours of simulation. All results below are the averaged value over 30 trials. We examine the impact of each of the cultural parameter and also the effects of mixing individuals with different cultural parameters in the same physical crowd on crowd dynamics. In this extended abstract we summarize a small subset of preliminary results.

First, an important question is whether the fidelity of the simulation is sufficient to support conclusions as to human crowds. Thus we examined whether the simulation can produce similar behavior to that of the observed human pedestrian crowd. To carry out the comparison, we recreated the initial settings in four of the videos in simulation. Specifically, we set the density of the pedestrian crowd (how many pedestrians per unit area); we set the individual parameters of agents and groups per the measured quantized values from the videos. Then we quantitatively compared the macro level measures (flow and mean speed) generated by the simulation to those of the crowds in the videos.

The results of flow comparison show fairly low error rates (where the error is measured in percentage of difference between the simulated and the observed values). In two of the France movies, we have 15% and 4% errors, in a *Canada* movie we have 16% error (this was the maximum error across all movies from all cultures), and in `London` we have 10% error. The mean flow error is 11%. The results of mean speed comparison show that in one `France` movie we get 21% error (the maximal error), in `Canada` we have 10% error and in `London` we have 6% error. The mean error for crowd speed is 13%. Note that because the simulation is using low-resolution discrete results (e.g., only three values for speed) and mean values overall, a perfect match is essentially impossible.

Encouraged by the fidelity results, we examined the impact of different personal spaces among the pedestrians on their number of collisions, and on their mean speed. The preliminary results show that there is a difference in number of collisions between collisions, as we vary the percentage of pedestrians who maintain shorter (*close*) and longer (*far*) personal spaces around them. In homogeneous groups, where everyone maintains *close* or everyone maintains *far*, there are relatively high number of collisions, though the groups do differ between them. Surprisingly, the lowest number of collisions have been found in the 50%-mixed group. The number of collisions is lower than both the homogeneous *close* and *far* groups.

The results also show that agents with *close* personal space have higher mean speed than agents with *far* personal space, although both of the groups were initialized with the same speed individually (so the effect is definitely due to just personal space preferences). Moreover, there is a difference between the *close*- and *far*-homogeneous groups. The differences in mean speed also have been found between the homogeneous groups and the heterogeneous 50%-mixed group.

We also examine the effect of *complete cultures* on crowd dynamics, where a complete culture is defined by a set of values assigned to the cultural parameters, as extracted from the video analysis. To do this, we simulated mixed-culture pedestrians moving on a sidewalk. As an exemplar, we report here on crowds mixing two cultures: Iraq and Canada. We vary the number of pedestrians in the crowd who are initialized with Iraqi cultural parameters, from 0% (homogeneous Canada crowd) to 100% (homogeneous Iraq crowd). We measure the impact of such mixing on the number of collisions, and on mean speed, as above.

The results of the collisions numbers show that in the heterogeneous groups, the higher the percent of Canadian in the population the higher the number of collisions. The lowest number of collisions has been found in population of 20% Canada pedestrians. For example, there is a difference between this population, and the population with 80% Canada pedestrians. Interestingly, the number of collisions in the homogeneous Iraq population jumps up, compared to the 20% Canada crowd.

The mean speed results show that an increased number of Canadian pedestrians in the population leads to higher mean crowd speed (indeed, our human pedestrians analysis shows that the Canada pedestrians had higher mean speed than Iraq pedestrians). The lowest mean speed has been found in population with 80% Iraq pedestrians.

4. EVACUATION CULTURES

Cultural differences have also been found in evacuation domain. Based on our literature survey, we model the following cultural parameters of individual evacuees (evacuating agents): (1) their tendency to notify others regarding an event that have caused them to evacuate, (2) the seriousness with which people (agents) hearing about such an event take it (that is, whether they decide to evacuate too, as a result) and finally, (3) we model the tendency towards evacuating in groups or individually [1]. We then examine the impact of these cultural parameters on the resulting macro level crowd evacuation behavior. Again, for lack of space, we report only on small subset of preliminary results.

We examined the agents' tendency to notify others regarding an event on evacuation rate, with no guards present. The preliminary results show that the more agents communicate, the faster the evacuation time. However, there was no significant difference between agents that pass knowledge of the event to all close neighbors (100% message passing) and agents that pass information to 80% of close neighbors (80% message passing). Thus the evacuation time essentially hits a floor at the level where 80% of the neighbors are informed. We also examined the same settings, except for adding authority figures who act to inform and guide evacuation. The general trend is the same, but the addition of authority figures makes a significant difference in relatively non-communicating agents. For example, the mean evacuation time in population of non-communicating agents (0% notify others) with five guards is cut by almost a half.

5. SUMMARY

We briefly described our first steps to explore the impact of micro-level, individual agent, cultural parameters on macro-level crowd behavior. Building on existing literature which investigates culture in human crowds, we identified important cultural parameters in two physical crowd domains (pedestrian movement and evacuation). We implemented these in established agent-based simulations for these domains, and used the simulations to measure their impact on crowd dynamics. We thus go beyond existing work, which focused on describing cultural parameters of individuals, without investigating their crowd-level effects.

6. REFERENCES

[1] K. Andrée and B. Eriksson. Cultural differences in an evacuation scenario - a study comparing australian and swedish responses. Technical report, Lund University, 2008.

[2] C. Beaulieu. Intercultural study of personal space: A case study. *Journal of applied social psychology*, 34(4):794–805, 2004.

[3] V. J. Blue and J. L. Adler. Cellular automata microsimulation of bidirectional pedestrian flows. *Transportation Research Record*, pages 135–141, 2000.

[4] N. Fridman and G. A. Kaminka. Modeling pedestrian crowd behavior based on a cognitive model of social comparison theory. *Computational and Mathematical Organizational Theory*, 16(4):348–372, 2010. Special issue on Social Simulation from the Perspective of Artificial Intelligence.

[5] E. Hall. A system for the notation of proxemic behavior. *American anthropologist*, 65(5):1003–1026, 1963.

[6] E. Hall and M. Hall. *Understanding cultural differences*. Intercultural Press, 1990.

[7] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews on Modern Physics*, 73, 2001.

[8] R. Levine and A. Norenzayan. The pace of life in 31 countries. *Journal of cross-cultural psychology*, 30(2):178, 1999.

[9] M. Moussaïd, S. Garnier, G. Theraulaz, and D. Helbing. Collective information processing and pattern formation in swarms, flocks, and crowds. *Topics in ocgnitive science*, 1(3):469–497, 2009.

[10] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS One*, 5(4):e10047, 2010.

[11] J. Tsai, N. Fridman, M. Brown, A. Ogden, I. Rika, X. Wang, S. Epstein, A. Zilka, M. Taylor, M. Tambe, E. Bowring, S. Marsella, G. A. Kaminka, and A. Sheel. ESCAPES - evacuation simulation with children, authorities, parents, emotions, and social comparison. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-11)*, 2011.

The Spanish Steps flower scam - agent-based modeling of a complex social interaction

(Extended Abstract)

Ladislau Bölöni
Dept of EECS, University of Central Florida
lboloni@eecs.ucf.edu

ABSTRACT

We model in detail a short human interaction scenario, the Spanish Steps flower scam. The scenario involves elements of negotiated commercial transaction, deceit, clash of cultural values and manipulation of public perception. The behavior of the actors is difficult to fit into a model of utility maximizing agents (even if we allow for bounded rationality). To model the scenario, we introduce a model where agents consider *vectors* of metrics which are not directly and linearly convertible into each other. The vectors consist of a mix of *concrete* and *culture sanctioned* metrics, with some of the latter being evaluated from the perspective of the self, the peers as well as the general public.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Multiagent systems*

General Terms

Human Factors, Economics, Experimentation

Keywords

agents, social models, simulation

1. INTRODUCTION

In this paper we model a flower selling scam, perpetrated in many tourist sites in Italy, such as the Spanish Steps in Rome. The intention of the seller is to pressure a client (typically a woman or a romantic couple) to purchase of an artificial rose at a high price:

- The seller offers a bouquet of flowers to the client. The client declines to purchase.
- The seller offers a single flower, relying on gestures implying that it is a gift. If the client refuses to take the flower, he repeats the offer several times, pushes the flower into the client's hands, or inserts it into her bag.
- The seller waits for 15-60 seconds several steps away from the client, who assumes that the interaction had concluded.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

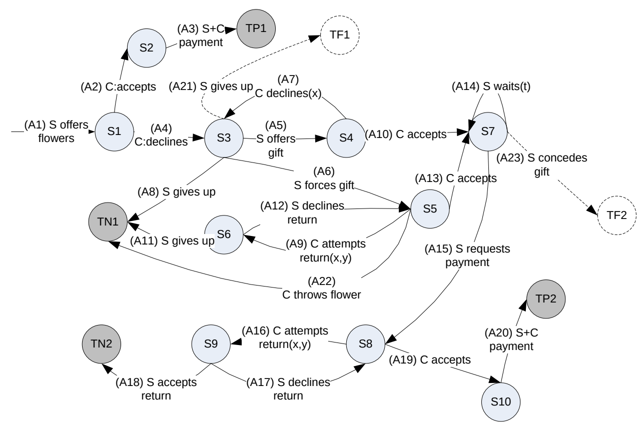


Figure 1: Action-state graph of the Spanish Steps scam.

- The seller approaches the client and requests payment.
- The client attempts to return the flower. The seller refuses to take it. The action concludes by either the client paying or by escalating her verbal efforts to return the flower until the seller decides to take it back.

The actions and states of the interaction are outlined in the action-state graph shown in Figure 1. Note however, as both the states and the actions must be further specified by *detail variables* which characterize the beliefs and mental states of the participants, and the mode of execution of the actions. For instance, actions A7, A9 and A16 are parametrized by their "loudness" x which determines how many onlookers will overhear the transaction and their "offensiveness" y which will determine how the action will impact the social metrics of the actor and target of the action. The action A14 is parametrized with the waiting time t it involves. As the detail variables encode the history of the participants, the action-state graph is *not* an MDP. To unroll of the graph into a MDP would require the us to quantify the detail variables, and it would be several orders of magnitude larger.

The Spanish Steps flower scam, despite being physically simple, is based on a series of complex decisions. It is, at its roots, a negotiated commercial transaction, which, however, is initiated by a *deceit* – the implication that the flower is a gift. The deceit is facilitated by the *blocking of the normal channels of communication* – the seller is usually a good speaker of several languages, but faking reduced communication ability helps position the deceit as a misunderstanding.

The successful conclusion of the scam relies on the *manipulation of the public perception:* the client needs to have the impression that everybody around believes that he accepted the commercial transaction.

Why do some clients accept to pay for the flower, well knowing that they are cheated? Conversely, why does the seller, in some cases, give up, without pushing the selling process to the extremes? It is obvious that as long as we consider a utility function which maximizes financial value, the actors do not act as rational agents.

2. RELATED WORK

A number of recent approaches implement agent based models of human social, cultural and emotional behavior. For instance, Bosse, Jonker and Treur [1] model a theory of neurologist Antonio Damasio about the three levels of perception of the emotional state. Miller et al. [4] operationalize the Brown and Levinson politeness model [2], while in a follow-up work [5] investigate how the relationship between culture (as examplified by Hofstede's cultural factors) and conversational politeness levels affect directive compliance. The POLLy system [3] also rely on the Brown and Levinson model to generate dialog for language learning.

3. CULTURE SANCTIONED SOCIAL METRICS

Our model assumes that the agents explicitly maintain a vector of *metrics*, separated in two classes. *Concrete* metrics such as financial worth or time are easily measurable and come with their native measurement units (e.g. dollars or euros for financial worth, seconds or minutes for time). The second class of metrics we consider are *Culture Sanctioned Social Metrics* (CSSMs). We say that a culture *sanctions* a metric if it (a) has a name for it, (b) provides an (informal) algorithm for its evaluation, (c) expects its members to continuously evaluate these metrics for themselves and salient persons in their environment and (d) provides rules of conduct which depend on these metrics. A person can know more then one culture, and simultaneously evaluate CSSMs according to multiple cultures. However, evaluating the CSSMs can be a significant cognitive load, and busy people might not necessarily perform highly detailed evaluations of their ongoing environment. Similarly, there is no guarantee that the agents would obey the rules of a culture concerning a certain metric (but they would be aware of the transgression). CSSMs can be evaluated from the perspective of the self, peers or general public.

To model the Spanish Steps scenario we used two concrete metrics: financial worth W and time T. The CSSMs used were *dignity* D and *politeness* P. Both sides consider the values from the perspective of the self and the public; the client also considers a peer (the other member of the romantic couple). With these assumptions, the vector of metrics for the client is $\{W^c, T^c, D^c, D^c_p, D^c_r, P^c, P^c_p, P^c_r\}$ while the vector of the seller is $\{W^s, T^s, D^s, D^s_p, P^s, P^s_p\}$.

4. BELIEFS AND PUBLIC PERCEPTIONS

The impact of an action on a CSSM is modulated by the beliefs of the agent about specific aspects of the current context. To model observed behavior of the real world players in the Spanish Steps scenario, we need to consider at least the following beliefs:

B^c_{gift} the client's belief that seller intends the flower to be a gift

B^c_{agr} and B^s_{agr} the client's and, respectively, seller's belief that the general public thinks that a transaction had been agreed upon.

We have used the Dempster-Shafer theory of evidence [6] to trace the beliefs, with the actions of the participants being considered as evidence for and against the beliefs. Beliefs are *dynamic*, in the sense that the passage of time, without any specific event can also constitute an evidence. For instance, B^c_{gift} increases with the time the agent is holding the flower without being asked for payment.

5. EXPERIMENTAL RESULTS

We have implemented the model in the YAES simulation environment and used it to trace the evolution of the CSSMs in a number of scenarios observed from the real world. We found that the model can provide satisfactory explanations to different outcomes of the scam. For outcomes where the seller was successful, the perceived beliefs had evolved such that the client can not escalate its efforts without massively lowering his public and peer politeness and dignity. We have also modeled situations where the seller, being in a rush, did not wait enough in action A14 to establish the public perception of an accepted transaction B^c_{agr}. In this situation, the client can escalate its efforts without being penalized in public perception, thus the scam will fail.

Acknowledgements: The research reported in this document/presentation was performed in connection with Contract Number W911NF-10-2-0016 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, or the U.S. Government unless so designated by other authorized documents.

6. REFERENCES

[1] T. Bosse, C. Jonker, and J. Treur. Formalisation of Damasio's theory of emotion, feeling and core consciousness. *Consciousness and cognition*, 17(1):94–113, 2008.

[2] P. Brown and S. Levinson. *Politeness: Some universals in language usage*, volume 4. Cambridge Univ Pr, 1987.

[3] S. Gupta, M. Walker, and D. Romano. How rude are you?: Evaluating politeness and affect in interaction. In *Proc. of the Second International Conference on Affective Computing and Intelligent Interaction (ACII-2007)*, pages 203–217. Springer, 2007.

[4] C. Miller, P. Wu, and H. Funk. A computational approach to etiquette: operationalizing Brown and Levinson's politeness model. *IEEE Intelligent Systems*, pages 28–35, 2008.

[5] C. Miller, P. Wu, V. Vakili, T. Ott, and K. Smith. Culture, politeness and directive compliance. In *Proc. of the 5th International Conference on Universal Access in Human-Computer Interaction*, pages 568–577, July 2009.

[6] G. Shafer. *A mathematical theory of evidence.* Princeton university press Princeton, NJ, 1976.

Effect of defectors for cooperation: How strictly should defectors be eliminated from the newcomers?

(Extended Abstract)

Hitoshi Yamamoto
Faculty of Business Administration, Rissho University
3-2-16, Osaki, Shinagawa-ku
Tokyo, Japan
hitoshi@ris.ac.jp

Isamu Okada
Faculty of Business Administration, Soka University
1-236 Tangi-cho, Hachioji City
Tokyo Japan
okada@soka.ac.jp

Yuki Ogawa
Center for Service Research, AIST
Central2, 1-1-1 Umezono, Tsukuba-shi
Ibaraki, Japan
yuki.ogawa@aist.go.jp

ABSTRACT

Yamamoto et. al.[6] have discovered that cooperation can be robustly maintained in a metanorms game by introducing into the population a small number of agents that always act non-cooperatively. They call this a "social vaccine" effect. In this paper we focus on the implications of a social vaccine. We therefore consider a model where there is a constant flow of newcomers into the population. How strictly should non-cooperators be eliminated from the newcomers in such a model? In this paper, by assuming a case where cooperative participants and non-cooperative participants are trying to participate in a population where metanorms are functioning, we investigate how well cooperation within the population is maintained by a strict population management policy where only cooperative participants are allowed to participate, and a simple population management policy where non-cooperative participants are admitted to some extent.

Categories and Subject Descriptors

I.6.6 [SIMULATION AND MODELING]: Simulation Output Analysis

General Terms

Design

Keywords

Social vaccine, Meta-norms, Evolution of cooperation, Agent-based simulation, Public goods game

1. METANORMS GAME WITH NEWCOMERS

The metanorms game[1] is a well-known model for maintaining norms in a population. As an extension of the n-person prisoner's dilemma, this game provides an excellent model for studying how norms are maintained in a population without a centralized authority, such as problems involving cooperation on international affairs. For example, Heck[2] and Horne[3] performed a psychological experiment in which metanorms were shown to exist.

We consider a situation where there is an influx of new participants into a population. Assuming a case where cooperative participants and non-cooperative participants are attempting to join a population, we discuss what sort of control policy is effective for maintaining cooperation when the population operates a strict control policy of only admitting cooperative participants, and when it operates a simple control policy where non-cooperative participants are also admitted to some extent. It could be said that this is a highly abstracted model of the problem of whether the stability of a society is more effectively maintained by adopting an immigration policy of only admitting people who have a strong affinity with the country's policies, or by adopting a lenient policy and accepting some degree of risk.

We consider groups on a social network with a population size of 100. However, a metanorms model featuring mutual surveillance among all members of a group leads to an upper limit in the number of group members due to cognitive limits and that a system of mutual surveillance is an unrealistic, severe restriction. In light of these criticisms, extending the metanorms game to a partial group[5] and limiting the study to mutual surveillance in a small world network[4] have been proposed. In order to understand the basic properties of the model, the initial state of the network structure is assumed to be a non-oriented regular network. The average degree of the population is D.

The agents play a metanorms game on a network where they are all interconnected by links. An agent j that has been defected by agent i and has received a payoff of H is an agent with a link to agent i, and an agent capable of punishing agent i must also have a link to this agent. In the evolution process, agents that are capable of becoming the parent of each agent must also be linked agents.

At the stage where the first generation of the metanorms game has completed, the F agents with the lowest payoffs in the population are withdrawn. An equal number of agents are then admitted to the population. The strategies of these newly admitted agents are discussed below. Newcomer agents are linked with randomly selected exist-

Appears in: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012), Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

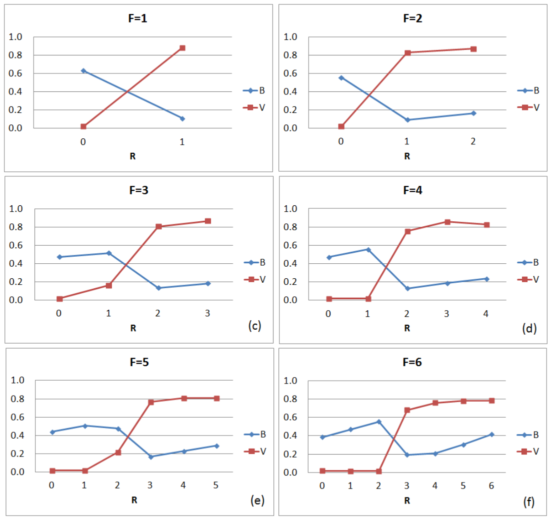


Figure 1: Effect of a social vaccine in newcomers

ing agents up to the average degree (D).

A wide variety of policies can be considered with regard to how newcomers are controlled, but in this paper we assume a simple model to observe the effects of the social vaccine. Here, newcomers are assumed to be either good or bad. In this context, a bad newcomer refers to a social vaccine. The population control policy is expressed as a level of rigor ranging from a strict monitoring policy where bad newcomers are never admitted, to a lenient policy where a blind eye is turned to these admissions to some extent.

Specifically, the strategy of good newcomers is taken to be $(B, V) = (0, 0)$, and the strategy of bad newcomers is taken to be $(B, V) = (1, 0)$ where B and V stand for boldness and vengefulness, respectively, as well as Axelrod[1]. The degree of rigor is expressed as the number R out of F newcomers for which a blind eye is turned to the admission of bad individuals ($R \leq F$).

For the payoff parameters of this section, we used the same values as in the Axelrod's experiment[1], i.e., $T = 3, H = -1, E = -2$ and $P = -9$. Also, the strategies (B, V) of first-generation agents are each given by uniform random numbers.

2. RESULTS

We analyze the effects of control rigor on the numbers and influx of newcomer agents, with the average degree D fixed at 20 the same as the population size of Axelrod's basic model (Fig. 1). Each of the graphs in Fig. 1 shows a plot of R on the horizontal axis and the average values of B and V of a population at the end of the simulation on the horizontal axis, for values of F ranging from 1 through 6.

When the number of newcomers is $F = 1$ (Fig. 1(a)), in the state where $R = 0$ - i.e., where bad individuals are completely prevented from entering the population and only good people can enter - it can be seen that cooperation is not achieved. However, when $R = 1$ - i.e., when newcomer agents adopt a defection strategy - cooperation is achieved at a high level. Similarly, when $F = 2$ (Fig. 1(b)), cooperation is not achieved when $R = 0$ but is achieved when bad agents are admitted. A similar trend was observed for $F \geq 3$, but with a gradual increase in the threshold value of R for which the social vaccine functions effectively. For example, when the number of newcomer agents is $F = 4$ (Fig. 1(d)), a value of $R = 1$ indicates the state where one bad individual (social vaccine) enters the population, while the other new entrants are all good. In this case, cooperation is not achieved. However, cooperation is achieved when $R = 2$. Cooperation is also maintained for $R = 3$ and $R = 4$, albeit not to as great an extent as for $R = 2$.

A characteristic feature of these experimental results is that the value of V for the population (i.e., its vindictiveness with regard to defection) differs widely in the vicinity of the threshold value at which cooperation is achieved, and is maintained at a high level in environments where cooperation is achieved. Specifically, the admission of a certain level of social vaccine into a population prevents the value of V for the group from decreasing, and as a result realizes a society that is robust against defection. This phenomenon resembles the immune function of resistance to a pathogen whereby inoculation with a weakened pathogen leads to the creation of antibodies to the pathogen. We call this a "social vaccine" effect.

3. CONCLUSION

We assumed a state where there is a constant influx of new participants into the population in a metanorms game, and we analyzed what sort of admissions policy the population should apply to newcomers in order to ensure that cooperation is robustly maintained. For simplicity, we expressed the admissions policy as the degree to which a blind eye is turned to the admission of uncooperative agents when fully cooperative agents and fully non-cooperative agents are both trying to enter the population.

In simulation experiments, we found that cooperation collapses in populations where entrants are subject to constant rigorous monitoring so that only cooperative agents are allowed to enter, but conversely cooperation is maintained at a high level when the entry of non-cooperative agents is overlooked to some extent.

4. REFERENCES

[1] R. Axelrod. An evolutionary approach to norms. American Political Science Review, 80(4):1095–1111, 1986.

[2] D. Heckathorn. Collective sanctions and compliance norms: A formal theory of group-mediated social control. American Sociological Review, 55(3):366–384, 1990.

[3] C. Horne and A. Cutlip. Sanctioning costs and norm enforcement: An experimental test. Rationality and Society, 14(3):285–307, 2002.

[4] D. Newth. Altruistic punishment social structure and the enforcement of social norms. In R. Khosla et al. (Eds.): KES 2005, LNAI 3683, pages 806–812, 2005.

[5] M. Prietula and D. Conway. The evolution of metanorms: quis custodiet ipsos custodes? Computational and Mathematical Organization Theory, 15(3):147–168, 2009.

[6] H. Yamamoto and I. Okada. Vulnerability and evolution of cooperation in the metanorms game. In Proceedings of 3rd World Congress on Social Simulation, 2010.

Patterns of Migration and Adoption of Choices By Agents in Communities

(Extended Abstract)

Feyza Merve Hafızoğlu
Department of Computer Science
University of Tulsa, Tulsa, OK, USA
feyza-hafizoglu@utulsa.edu

Sandip Sen
Department of Computer Science
University of Tulsa, Tulsa, OK, USA
sandip@utulsa.edu

ABSTRACT

We study the migration and behavior adoption patterns of agents situated in geographically distributed communities. We consider agents with two types of states or opinions, binary and continuous. Agents either probabilistically adopt the predominant state in their community or migrate to another community more supportive of their state. We observe an interesting range of emerging population patterns based on different migration and adoption biases.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Opinion dynamics, Emergence, Migration, Adoption

1. INTRODUCTION

We are interested in studying emerging patterns of opinions in population of agents in a society adopting one of several choices or opinions as a convention or norm [1, 2]. We believe that agents are governed, among other forces, by two somewhat conflicting but important influences: the desire to "fit in" in their social environment, and the attraction of environments more receptive or supportive of their preferences. We are therefore interested in investigating the issues of "peer pressure" and migration inertia on the emergence of divergent opinions in spatially distributed, yet connected, sub-populations (communities). We believe that better understanding of opinion dynamics under such constrained interactions and the interplay of behavior adoption and migration patterns can improve our understanding of real-life multiagent systems and help us better design effective interaction models and infrastructure to facilitate smooth, coherent functioning of such open multiagent systems.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

We assume that agents are communities that are connected in some known topological structure. Each agent may start with a bias for one of the several available opinion options but can be influenced by other agents in its community to change its choice. Agents also can leave for "greener pastures," i.e., if an agent is unsatisfied with the emergent convention or opinion in its community, it may move to a community perceived to have a more widespread support for the option it prefers.

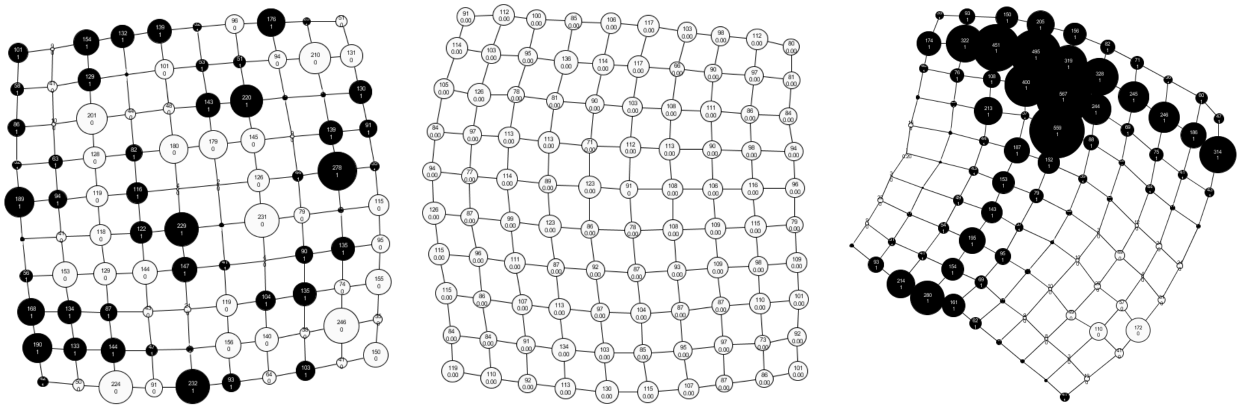
Agents' opinions (choices) are represented by either binary or continuous valued variables. The opinion o_i of a binary agent i can be either zero or one: $o_i \ \epsilon \{0, 1\}$. In real life, people have different degrees of preference for the opposing opinions. For modeling this situation, the opinion of a continuous agent, $o_i \epsilon [0, 1]$, can be interpreted as the probability of adopting one of two possible states.

A community is assumed to have reached *opinion consensus*, when all agents present in the community have the same opinion. The entire population is said to have *converged* if all communities have reached opinion consensus. The *state* of a community is the average opinion over all its members.

We assume that agents prefer to interact with other agents having similar opinions. Our agents are assumed to be cognizant of the state of their community as well as those of the immediate neighboring communities. Agents decide to stay or migrate based on the dissimilarity between its opinion and the state of its community. The migration probability, P_i^M, is a function of the disparity between opinion of agent i, o_i, and the state, s_j, of its community: $P_i^M = |o_i - s_j|^\beta$, where β determines the migration inertia. If an agent decides to migrate, it migrates to a more supportive neighbor community with a higher probability.

Adoption is a result of social influence in communities. Binary agents adopt the opposite opinion with a probability proportional to the fraction of agents having opposite opinion. Adoption probability of binary agent i in community j, $P_{i,j}^A$, is defined as follows: $P_{i,j}^A = |o_i - s_j|^\gamma$, where γ is used to modify the adoption rate.

Continuous agents use an interaction based adoption. Each agent randomly picks another agent in its community, they flip coins biased by their respective opinion values; the result of the coin toss is either zero or one. If both agents pick the same value, they are coordinated. Otherwise, there is a conflict. When coordinated, agents increase or decrease their opinion values simultaneously by a certain amount, Δ. When a conflict occurs, they change their opinions in the opposite way to the result of coin toss.



(a) Heterogeneous communities: moderate migration, conservative adoption

(b) Unanimous communities: eager migration, conservative adoption

(c) Clustered communities: eager migration, eager adoption

Figure 1: State and population distribution of communities of continuous agents

2. RESULTS

Simulation proceeds in discrete timesteps, where first migration and then adoption takes place in each timestep with synchronous updates. We present results from experiments with communities situated in a two-dimensional toroidal grid. There are 100 communities and the initial population is 100 in each community. Agent opinions are initialized by using a uniform distribution.

For conservative, moderate, and eager migration, the values of β are 10, 1, and 0.25, respectively. Binary agents use γ parameter with a value of 0.33 for eager adoption and 3 for conservative adoption. Conservative, moderate, and eager adoption of continuous agents are represented by the values of Δ: 0.01, 0.05, and 0.1, respectively. All combinations of migration levels and adoption rates are analyzed.

Results show that increasing the migration tendency and reducing the adoption tendency primarily affect the distribution of community sizes. We observe an interesting emergent pattern for high standard deviations of populations (eager migration and conservative adoption), the population of some communities declines drastically or they can even completely "die out". It is very likely such smaller communities are attached to larger communities by the end of simulations, thus demonstrating an *emergent phenomena of big cities with smaller suburbs.*

The entire population converges in all cases except communities of binary agents using eager adoption irrespective of migration inertia. This is because even when the opposite opinion is supported by only a small minority, eager adoption makes them switch from the majority opinion. In this case, binary agents become incredibly capricious: they change their opinion frequently.

Figure 1 presents snapshots of the communities of continuous agents at the end of typical runs. A community is represented by a circle whose size is proportional to its population size. The size and state of a community are written on the circle. The color indicates the state of the community: black (white) circle means the *state* is 1 (0). Mixed states are indicated with different shades of gray proportional to its value.

Using low values of Δ for adoption and relatively higher migration tendency, we obtain a heterogeneous grid with respect to the community sizes (see Figure 1(a)). Interestingly, unanimous population (all communities converge to the same opinion) is obtained with the continuous agents using eager migration and conservative or moderate adoption (see Figure 1(b)). The entire population is homogeneous with respect to both community sizes and opinions. The population converges to one or zero, depending on the initial average opinion of the entire population (whether it is above or below 0.5).

The most interesting emergent pattern is shown in Figure 1(c) for continuous agents using eager migration and eager adoption. This grid consists of clusters of communities of same opinion. Additionally, in contrast to the general trend, the standard deviation of community sizes increase as the adoption tendency increases in this case. The grid is almost unanimous except for a couple of small communities converging to a differing opinion. When we look at the convergence time versus the final population of communities: the larger communities appear to be converging earlier! With the aid of high migration and adoption, some communities converge to a stable state earlier and become attractive for the agents, who are willing to find a community where they will be satisfied. It can be thought as agents appear to be flocking towards the completely converged communities since they are eager to migrate. These early adopters act as "black holes" by sucking in deserters from surrounding communities! The difference in the patterns of community sizes in Figures 1(b) and 1(c) is due to the different convergence time of the communities.

3. REFERENCES

[1] P. Holme and M. E. J. Newman. Nonequilibrium phase transition in the coevolution of networks and opinions. *Phys. Rev. E*, 74:056108, Nov 2006.

[2] G. Iñiguez, J. Kertész, K. K. Kaski, and R. A. Barrio. Opinion and community formation in coevolving networks. *Phys. Rev. E*, 80:066119, Dec 2009.

Agent-based simulation of mobility in real-world transportation networks: effects of acquiring information and replanning en-route

(Extended Abstract)

Maicon de B. do Amarante and Ana L. C. Bazzan
Instituto de Informática/UFRGS, 91501-970, P.Alegre, RS, Brazil
{mbamarante,bazzan}@inf.ufrgs.br

ABSTRACT

Macroscopic and equilibrium-based models for traffic assignment and simulation disregard many details of traffic movement. For some applications, one needs to understand and analyze microscopic properties. This paper discusses an agent-based simulation of route choice under different conditions of demand generation, number, and types of travelers. The effects of en-route decision-making and vehicle-to-vehicle communication were tested in a real-world scenario. The analysis has considered different classes of travelers, which is only possible if a microscopic, agent-based simulation is used. The main conclusion is that for travelers whose trips are long, there is a benefit of using communication and replan en-route, depending on the demand volume.

Categories and Subject Descriptors

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

General Terms

Experimentation, Human Factors

Keywords

Traffic Simulation, Agent-based Simulation,V2V

1. INTRODUCTION

Macroscopic simulation methods for assignment of traffic demand are generally based on equilibrium computation and assume steady state conditions on the links. Thus it is less useful when one intends to do microscopic investigations related to short time frames and travelers' individual features such as local perception of the traffic state, ability to communicate (e.g. by means of vehicle to vehicle communication or simply V2V), or replan en-route. Also, the computation of the exact equilibrium is not only a non-trivial problem, but also it is probably a useless effort given that this equilibrium will not last long due to the dynamic nature of the environment. Therefore, we follow a different line, namely dynamic traffic network disequilibrium. Here, one basic assumption of equilibrium-based approaches is relaxed: users are no longer assumed to be fully and accurately informed. Rather, disequilibrium research focuses on the adaptation process by which users' experiences in one period of time affect their decisions in subsequent periods.

In order to tackle this detailed representation of the demand and, especially, of the user's adaptation, in the present paper agent-based simulation is used. Thus, instead of using volume-delay functions (functions that express the average or steady-state travel time on a link in terms of volume in this link) to estimate costs, the actual travel time of each driver in each link results from the actual driving.

Intuitively one assumes that broadcasting information to the user of a transportation network is only beneficial if it is *not* the case that everyone receives the same information, at least not in networks that operate close to the saturation level in key portions of it. Therefore, the aim of this work is to investigate whether drivers could do better by using only local perception, i.e., the (partial) information is collected by the driver itself. Later we extend this concept to include information that is also gathered by means of V2V.

Several publications suggest the application of intelligent agent architectures to different travel-related choice processes such as route and mode choice. Agent based approaches seem to be particularly relevant when networks are dynamic or when dynamic information is available. As mentioned, a large number of works about the effect of information on route choice uses abstract scenarios based on static assignment and/or VDF. These abstract scenarios are mostly inspired by congestion or minority games. Examples are for instance Bazzan *et al.* [4] and Chmura and Pitz [3]. Similar goals to our work appeared in [1] but it was mentioned that when en-route replanning is to be included, the situation becomes considerably more complicated. This explains why en-route replanning is seldom considered in other approaches. Thus, the contributions of this paper are the use of agent-based paradigms to simulate the effect of en-route replanning and different levels of information.

2. METHODS

In order to implement this kind of simulation we use the microscopic traffic simulator ITSUMO [2] to represent the actual movement of the vehicles. Next we briefly describe the main steps.

Demand generation. Demands are represented by an

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

O-D (origin-destination) matrix. The O-D used here resembles roughly the main origins and destinations of the city of P. Alegre (Brazil). Besides these real-world data, in this paper we also use a uniformly distributed demand. See [2] for a discussion of these two types, in a smaller scenario.

Agent-based routing. ITSUMO allows the use of various algorithms (Dijkstra, A*, ARA*, anytime and dynamic shortest path algorithms for route computation). In [2] all algorithms were evaluated with the conclusion that normally A* is a good compromise between efficiency and cost. Therefore here A* is used here. In the traditional, macroscopic approach, routes are computed by a central entity and are assigned to users. Then, an iterative process occurs in which only some users' routes are adjusted in order to converge to the equilibrium. This makes sense in a centralized, non-agent based approach in which there is no autonomy by the agent itself. In contrast, in the agent-based case, given an O-D pair, the agent's knowledge about links traffic volume, and an algorithm such as A*, the agent itself computes its initial best route and departures. After, as the journey proceeds and more information is incorporated, this is used in the next journey or even during the same journey to perform some en-route re-planning.

Drivers and en-route re-planning. One of the important features is the driver's ability to re-plan during the trip when facing congestion (henceforth *en-route* planning).

Types of Agents. Combining the capabilities and knowledge of the agents, we came out with four kinds of simulation: FNR stands for full knowledge and no replanning; FR means that drivers do en-route replanning; P means local perception (thus partial knowledge); PC is local perception plus V2V. If the agent has access to full information, it knows the current demand at all links. If it does not, it may perceive it locally, thus having only partial information. In this case, it only knows volumes of those links it has traveled recently. Further, we test drivers equipped with V2V devices, which enable agents to have further information about traffic volume. Regarding en-route replanning, this is done periodically and is based on the information the driver has about the other links of the network.

3. SCENARIO AND RESULTS

Due to lack of space, we focus on the main conclusions that were drawn after running the simulations for the different demand sizes and all four classes of agents types. In all cases we have analyzed the number of trips performed (within the simulation horizon) and average travel time over all drivers (given in simulation steps). To take advantage of agent-based simulation, we have generated agents' performances by individual classes.

Analysis over All Classes of Trip Duration. When all drivers are considered, drivers with full knowledge perform better than those with partial knowledge. However, the important conclusions related to differences noticed when different classes of trip duration are considered, as next.

Analysis Within Classes of Trip Duration. Intuitively, one expects that the effects of en-route replanning and of the V2V be more significant for drivers whose trips take longer. This is what was observed in most cases. However there are differences in performance if the demand is uniform or O-D-based. In the former, for drivers whose trip take long, PR and P outperform FR, which outperforms FNR, i.e., partial information is valuable. In the O-D demand, full knowledge pays off only if the driver is able to replan en-route (the best performance is FR). There are two reasons for this. First, local perception seems not to be valuable (and hence neither V2V as this only spreads non-accurate perceptions). This is probably due to the fact that the network being big, the trips take long so that the acquired knowledge is not accurate after some time. The second is that if the driver is not able to replan en-route, then there is a chance that many drivers plan their routes over the same portion of the network and get stuck to them.

These results indicate that in more realistic network, as it is the case of O-D demands, there is a positive utility in performing en-route replanning when trips take a long time. This is not the case regarding short trips, probably because there is a cost of changing routes.

4. CONCLUSION AND FUTURE WORK

A summary of the investigations conducted by us is that there are differences among distinct classes of drivers, regarding trip duration, type of knowledge and ability. A macroscopic approach would only conclude that, for the overall population of agents, having full knowledge is advantageous. However, using a microscopic analysis that considers different types of drivers, it is possible to see that having full knowledge seems to be advantageous only if replanning is possible. We remark that the method proposed is general. One needs only to plugin its own scenario and O-D description in ITSUMO in order to generate these kind of data and analysis. Future work regards the decoupling of en-route replan and communication in order to better understand the effects of each.

Acknowledgments

The authors wish to thank CNPq and FAPERGS for their support.

5. REFERENCES

[1] M. Balmer, N. Cetin, K. Nagel, and B. Raney. Towards truly agent-based traffic and mobility simulations. In N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS*, volume 1, pages 60–67, New York, USA, July 2004. New York, IEEE Computer Society.

[2] A. L. C. Bazzan, M. de B. do Amarante, G. G. Azzi, A. J. Benavides, L. S. Buriol, L. Moura, M. P. Ritt, and T. Sommer. Extending traffic simulation based on cellular automata: from particles to autonomous agents. In T. Burczynski, J. Kolodziej, A. Byrski, and M. Carvalho, editors, *Proceedings of the Agent-Based Simulation (ABS) 2011)*, volume 1, pages 91–97, Krakow, June 2011. ECMS.

[3] T. Chmura and T. Pitz. An extended reinforcement algorithm for estimation of human behavior in congestion games. *Journal of Artificial Societies and Social Simulation*, 10(2), 2007.

[4] F. Klügl and A. L. C. Bazzan. Simulated route decision behaviour: Simple heuristics and adaptation. In R. Selten and M. Schreckenberg, editors, *Human Behaviour and Traffic Networks*, pages 285–304. Springer, 2004.

SimAnalyzer : Automated description of groups dynamics in agent-based simulations

(Extended Abstract)

Philippe Caillou
Laboratoire de Recherche en Informatique
Universite Paris Sud – INRIA
Orsay 91405, France
philippe.caillou@lri.fr

Javier Gil-Quijano
CEA, LIST, Laboratoire Information Modeles et Apprentissage
F-91191 Gif-sur-Yvette, France
javier.gil-quijano@cea.fr

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Experimentation, Meausurement, Verification

Keywords

Complex systems simulation, multiagent systems, automated observation, automated description, clustering, value-test

1. INTRODUCTION

Multi-agents based simulations (MABS) have been successfully used to model complex systems in multiple areas. However, a pitfall of MABS is that their complexity increases with the number of agents and behaviors considered in the model. For average and large systems, different phenomena can simultaneously occur at different intermediate levels and influence each other [2]. For instance, groups of agents (flocks of birds, social groups, etc.) following similar state's trajectories may appear, evolve and disappear. To describe and evaluate the evolution of groups, the observation of global and individual variables (like in [1]) is not sufficient anymore. Moreover, because of the emergent properties of complex systems, those groups may be unexpected, or their presence may even be unnoticed because no suited variable or any other adapted observation mechanism is provided in the simulator. The significance and even the existence of groups can then be hidden by the usually huge amount of available data. In this paper we introduce the use of statistical based tools to assist the modeler in discovering, describing and following the evolution of groups of agents, by combining data clustering and value test. Our model can be described within 5 main steps which will be illustrated with a NetLogo library model example.

2. ANALYSIS MODEL

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2.1 Model selection : what do we study?

Our model is generic for a simulation data stream. It may be generated directly from a simulation framework (NetLogo in our application) or extracted from a log file (by simulating an online data stream). We choose here the Bank Reservs model, provided with Netlogo, where financial agents either save money or borrow money via loans. In this model, some global variables give an overview of an experiment: we can observe phenomena as the stabilization of the total *money* when the maximum amount of *loans* is reached, or follow fixed groups of agents that depend on global parameters (for instance 3 groups: negative *wealth*, *wealth* higher than $richThreshold$ and the rest). Even if these informations are interesting, a more precise understanding of the model behavior can not be reached with such global/local observation. For example, *who are the wealthy agents? Do the rich stay rich?* This would be even more true for more complex models, for which variable interactions are much more difficult to deduce than with such very simple toy simulation.

2.2 Data processing : what is the data?

A data matrix is generated every st steps. A line in the matrix represents one agent's state. These raw data are not the only interesting variables for cluster's generation and analysis. From the set V_{ag} of variables that describes the agents, we generate the $\mathbb{V}calc$ and $\mathbb{V}init$ sets. In the $\mathbb{V}calc$ set, several filters/agregators can be considered to enrich the data stream. For now, we use one: for each variable, we add a new variable computed as its moving average. The $\mathbb{V}init$ set contains the initial values of V_{ag} of every agent in the simulation. By default, this *initial* variables are not used in the clustering but used for the later cluster's description. The subset of variables used for the clustering $\mathbb{V}clust$ has to be selected at the beginning of the simulation. By default, $\mathbb{V}clust = \mathbb{V}ag \cup \mathbb{V}calc$.

2.3 Clustering: can we find homogeneous groups?

Clustering is performed on the $\mathbb{V}clust$ data set in order to generate homogeneous agent groups. Our objective is not to propose a new clustering algorithm, and any clustering algorithm may be used (in our application, any algorithm of the Weka framework and the associated parameters). By default, the XMeans clustering algorithm is used with the classical similarity function based on the Euclidean distance.

2.4 Cluster's description: how can we describe them?

Once the clusters are identified, and given $\mathbb{A}$ the entire set of agents and $\mathbb{A}_C$ the set of agents in a cluster C it is possible to obtain easy-to-read descriptions of every cluster by using the value tests VT calculation (equation 1). The VT represents the significance of the average value $E(\mathbb{A}_C^v)$ of a given variable v for the agents in the $\mathbb{A}_C$ set compared to its distribution on the $\mathbb{A}$ set. Roughly, it can be described as the difference between the average of the Cluster ($E(\mathbb{A}_C^v)$) and the global average ($E(\mathbb{A}^v)$) normalized by the standard deviation ($\sigma^2(\mathbb{A}^v)$):

$$VT(v,C) = \frac{(E(\mathbb{A}_C^v) - E(\mathbb{A}^v))}{\sqrt{(\frac{sizeof(\mathbb{A}) - sizeof(\mathbb{A}_C)}{sizeof(\mathbb{A})-1} \times \frac{\sigma^2(\mathbb{A}^v)}{sizeof(\mathbb{A}_C)})}} \quad (1)$$

v is significant for the A_C set if $|VT(v,C)| \geq 2$. If $VT(v,C) \geq 2$, the value of v in A_C is in average higher than in A (and lower if $VT(v,C) \leq -2$).

A global overview of all clusters retrieved in a simulation makes it easy to compare clusters, to describe them and the most significant variables in the simulation. For example :

"who are the wealthy?" We use our analysis tool with the default configuration: $\mathbb{V}clust = \mathbb{V}ag \cup \mathbb{V}calc$. The clustering step is fixed to $st = 200$ simulation steps. At each clustering step, clusters are identified, visualized in NetLogo (with colors), and their extension and description are presented. For example, in $t = 400$, three clusters are identified: a "*poor*" cluster (114 agents, with low *wealth*, $VT = -9.57$), a "*medium*" cluster (20 agents) and a "*wealthy*" cluster: $cluster9$. This cluster is composed of the 66 rich people, with high *wealth* ($VT = 8.91$), *savings* ($VT = 9.68$) and few *loans* ($VT = -3.93$), the results for the associated moving average variables are similar. An interesting result is the significant $T0Wallet$ ($VT = 8.81$) variable, corresponding to the *wallet* value of agents at the beginning of the simulation. The *wealthy* people were significantly richer than the average at the beginning of the simulation.

"what are the rich characteristics at each step?" At the end of the simulation, a description of all the clusters obtained at each step gives a global overview of the simulation. In our experiment, it is always possible to identify a *wealthy* and a *poor* cluster, and sometimes (like in $t = 400$) a *middle* cluster. From their description, it is already possible to observe that the link between the *wealth* and the initial wealth (the $T0Wallet$) is not significant anymore after $t = 400$ ($|VT(T0Wallet, rich)| < 2$ for $t \geq 400$). It may be related to the fact (observed with NetLogo global observation) that bank has reached its loan limit (the total money stops to increase around $t = 230$). However, to compare clusters of different steps in this overview is difficult since they are different both in intension and in extension (see section 2.5). In a more complex model, cluster may have a completely different meaning at different steps.

2.5 Cluster's evolution: how do they evolve?

In order to describe the clusters' evolution, we consider two alternative hypothesis: either the extension or the intension in every cluster is considered as stable.

Fixed extension:*Do the rich stay rich?*

The first possibility is to fix the extension of a cluster (its population), at a given simulation step t. The description of the cluster C is updated at every step. Since the agents in the cluster stay the same, $VT(v_{init}, C_{t'}) = VT(v_{init}, C_t) \forall t' > t$ (unless some agents die or new agents enter the simulation).

If we consider for example the *wealthy* agent of $cluster9$, the initial parameters values ($\mathbb{V}init$) are stable (by definition). But for some other variables, all wealth-related, the differences with the other agents decrease: both *wealth*, *saving* and *loans* (absolute) VT values decrease. This mean that, in average, the wealthy people of $t = 400$ are less and less wealthy. Even if they are still significantly wealthier than the average in $t = 1200$, but their *loans* variable is already not significantly lower anymore from $t = 1000$.

Fixed intension:*Do the rich stay the same people?*

The alternative is to fix the intension of clusters. The intension of a set of clusters is the function that allows to assign an agent to the most relevant cluster. At each new step, the intension function is used to determine which agents to put in each cluster (some of them may be empty) and the new descriptions are computed. For a cluster C_t, for every timestep $t' > t$ a new cluster $C_{t'}$ is built. The description of every $C_{t'}$ can be easily compared with that of C_t. Since the cluster intensions are the same, the v-tests of the variables used in clustering ($\mathbb{V}clust$) are very likely to stay the same. But the v-tests of other variables, especially those of the initial variables ($\mathbb{V}init$) of the agents may evolve (since the population of the cluster changes).

In our example, for clusters of $t = 400$, all the variables considered in clustering ($\mathbb{V}clust$) are by definition roughly similar. However, the other variables evolve (in our example, the initial parameters of the agents $\mathbb{V}init$). $cluster9$ will for example regroup the wealthy agents at each step, but the number and the initial properties of its agents evolves. The number of wealthy agent ($card(cluster9, t)$) stay approximatively constant (66, 71, 56, 58, 65), but the evolution of the initial parameters confirms the observation made with the global overview: after $t = 600$ $|VT(TOWallet, rich)| < 2$ (the initial wealth of the agent ($TOWallet$) is not significant anymore after $t = 600$).

3. CONCLUSIONS

The simulation's observation framework that we present here, provides the modeller with generic tools that allow him/her to get a synthetic descriptive view of simulations. It can be used to understand the dynamics of simulations and to ease their validation. To allow the analysis of a wide number of different type of simulations we are currently adapting our framework to both consider qualitative and network variables and facilitate large simulations analysis. The latter will be done by integrating our framework to GAMA[4] and to the SimExplorer/OpenMole[3] engine.

4. REFERENCES

[1] F. Gaillard, Y. Kubera, P. Mathieu, and S. Picault. A reverse engineering form for multi agent systems. In *ESAW 2008*, pages 137–153, 2008.

[2] J. Gil-Quijano, T. Louail, and G. Hutzler. From biological to urban cells : lessons from three multilevel agent-based models. In *PRACSYS 2010*, Kolkota, India, 2010. LNCS.

[3] M. Leclaire and R. Reuillon. www.openmole.org.

[4] P. Taillandier, A. Drogoul, and D.-A. Vo. Gama: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In *PRIMA 2010*, 2010.

Emergence of Cooperation through Structural Changes and Incentives in Service-Oriented MAS

(Extended Abstract)

E. del Val, M. Rebollo, V. Botti
Universitat Politècnica de València
{edelval, mrebollo, vbotti}@dsic.upv.es

ABSTRACT

In distributed environments where entities only have a partial view of the system collaboration plays a key issue. In the case of decentralized service discovery in Service-Oriented MAS (SOMAS), agents only know about the services they provide and their direct neighbors. Therefore, they need the collaboration of their neighbors in order to locate the required services. However, collaboration is not always present in open and distributed systems. Non-collaborating agents pursuing their own goals could reject forwarding queries from other agents; therefore, the efficiency of the decentralized service discovery could be seriously damaged. In this paper we propose the combination of structural changes and incentives based on utility in order to promote the collaboration in the service discovery process.

Categories and Subject Descriptors

H. [**Information Systems**]

General Terms

Management, Performance

Keywords

Incentives, collaboration, service discovery, complex networks

1. SERVICE DISCOVERY SYSTEM

SOMAS are characterized by a finite set of agents $A = \{a_1, ..., a_n\}$, which offer their functionalities through services, and a set of links $L \subseteq A \times A$, which indicates the existence of a direct relationship between two agents. It is assumed that the knowledge relationship among agents is symmetric, so the network is an undirected graph.

In our model, agents are characterized by a tuple of five elements $(S_i, N_i(t), st_i(t), \Omega_i(t))$ where:

- S_i is the set of services provided by the agent
- N_i is the set of neighbors of the agent, $N_i \subseteq A - \{a_i\}$: $\forall a_j \in N_i, \exists(a_i, a_j) \in L$, and $|N_i| > 0$. It is assumed that $|N_i| \ll |A|$. Links between agents are established based on a social feature called homophily which measures the similarity between agents considering the services that the agents offer. For a detailed mathematical treatment of how homophily between agents is calculated, we refer the reader to [4];
- $st_i(t)$ is the internal state of the agent at a given time t. It is defined by a set of $(q, \#fw(t), \#sfw(t), \#rq(t), \#q(t), \#r(t), \varepsilon)$:
  - q represents the query that the agent receives asking for a service,
  - $\#fw(t)$ is the number of queries that the agent forwarded until time t,
  - $\#sfw(t)$ is the number of queries that the agent forwarded in a successful discovery processes until time t,
  - $\#rq(t)$ is the number of queries that the agent refuses to forward until time t,
  - $\#q(t)$ is the number of service requests attended by the agent until a given time t,
  - $\#r(t)$ is the number of service requests sent by an agent until a given time t,
  - ε is the threshold established by the agent to consider a service similar enough to a query.
- $\Omega_i = \{\omega_i(t), \omega_j(t+1), ...\}$: is the set of strategies used by the agent. Each strategy defines its behavior at a given time t.

Service discovery process in our system relies on the collaboration of the agents. The process starts when an agent a_i is looking for an agent a_t that provides a service s_t. The agent redirects the query to the most promising agent in its neighborhood. The most promising neighbor, $a_j \in N_i$, is the neighbor that is most similar to the target agent a_t (higher degree of homophily) that has the highest degree of connection. The selection function that calculates the most promising neighbor a_j of an agent a_i to reach the agent a_t is described with detail in [4].

If a_j does not offer a service that is similar enough, it chooses between two options: to forward the query or to not forward the query. If a_j does not forward the query, it sends a reject message to a_i, and a_i looks for another promising agent in its neighborhood to redirect the query. If a_j accepts forwarding the query, the query is sent to the most promising agent in the neighborhood of a_j. This process is repeated until the agent that offers a service that is 'similar enough' is found or when the TTL (Time To Live) of the query ends.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. STRUCTURAL CHANGES AND INCENTIVES

Distributed systems rely on the collaboration of the entities that participate in them. However, in open and heterogeneous environments, a common and more realistic situation is that selfish agents appear [1]. It is important to provide mechanisms to be able to confront the situation where agents that are pursuing their own goals without collaborating are damaging the performance of the overall system [3].

2.1 Structural Mechanism

Through interactions, agents should be able to change their relations taking into account which neighbors provide profitable relationships and which do not. This feature is called social plasticity [2]. In order to evaluate a link's utility, an agent uses a decay function that evaluates the probability of maintaining a link considering the number of queries rejected. This function is a sigmoid that ranges between [0,1],

$$D_{(a_i,a_j)}(\#rq,t) = 1 - \frac{1}{1 + b \cdot e^{\frac{-(\#rq-m)}{n}}}, \qquad (1)$$

where $\#rq$ is the number of queries that have arrived to neighbor a_j from agent a_i and a_j decides not to forward at a given time t. The parameters b and m are the displacement, and n is the steepness. These parameters are adjusted by the agent. If a query is forwarded through the link (a_i, a_j), $\#rq$ is updated to 0. Otherwise, the $\#rq$ is increased by one unit.

In the case that the agent a_i decides to break the link with neighbor a_j, a_i looks for another agent to establish a new link in order to maintain its degree of connectivity. We assume that any alternative agent always accepts a new partner. There are different criteria for establishing a new link with another agent in the network: establish a link with a neighbor's neighbor [2], look for a similar agent to me in order to keep the homophily of the system, look for an agent similar to the previous neighbor.

2.2 Incentive Mechanism

In our model, the strategies that an agent can choose at a given time $\omega_i(t)$ are : to collaborate or to not collaborate. Collaborating in the service discovery scenario implies that the agent is going to: forward queries, request services, and attend requests about its services. If the agent decides not to collaborate, it means that the agent is going to: request services and offer its services, but it is not going to forward the queries of neighbors. Considering the possible strategies and the actions involved in each strategy, the following utility function is defined:

$$u_i(\omega_i,t) = \begin{cases} \#q(t) \cdot PS - \#r(t) + RS & \text{if } \omega_i = \text{not coll.} \\ -\#fw(t) \cdot Q + \#sfw(t) \cdot SQ + & \text{if } \omega_i = \text{coll.} \\ \#q(t) \cdot PS - \#r(t) + RS & \end{cases} \qquad (2)$$

where ω_i is the strategy used by the agent at a given time t, and $\#q(t), \#sfw(t), \#fw(t), and\#r(t)$ is information of the internal state of the agent at a given time t (see Definition 1). In this function each action in the model implies a cost (forwarding queries (Q), and requesting a service (RS)) or a benefit (forwarding queries in a service discovery process that ends successfully (SQ), and providing a service (PS)).

We assume that all the agents have the same payoffs. Agents are rational entities that update their own behavior to maximize their own benefit. They also take into account the utility of their direct neighbors, and update their strategy. If the agent has a neighbor that has obtained a higher payoff in the previous iteration, the agent changes its strategy to the neighbors' strategy.

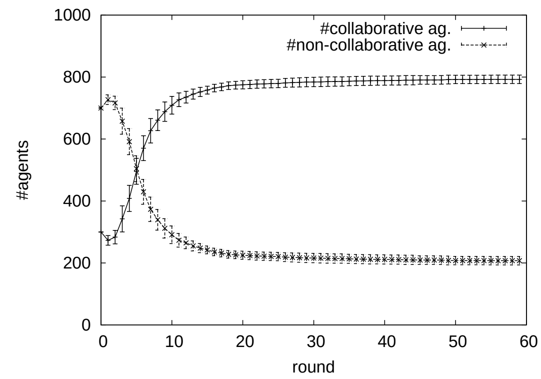


Figure 1: Evolution of collaboration in networks of 1000 agents (300 C and 700 NC). Agents consider utility and plasticity.

2.3 Structural Changes and Incentives

The use of structural mechanisms such as social plasticity or incentives promote the emergence of cooperation. Nevertheless, in scenarios where the predominant behavior is not to collaborate, the separate use of these mechanisms is not enough. Social plasticity could break the network and incentives cannot change the behavior due to the high number of non-collaborators. Therefore, we propose the integration of both mechanisms in order to facilitate the emergence of collaboration.

Basically, each agent evaluates its links considering whether or not its neighbors are collaborating in the forwarding process. This evaluation is done each time an agent receives or generates a query (see Eq. 1). With the result of this evaluation, the agent decides whether or not change its links. Moreover, in each iteration, each agent updates its utility and compares it with the rest of its direct neighbors. Based on this comparison, the agent decides whether or not to change its behavior in order to improve its payoff in future interactions. The results show that, even in scenarios where the predominant behavior is to not collaborate the collaboration emerges.

3. ACKNOWLEDGEMENTS

TIN2009-13839-C03-01 project, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, FPU grant AP-2008-00601 awarded to E. del Val.

4. REFERENCES

[1] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz. Characterizing selfishly constructed overlay routing networks. In *INFOCOM*, 2004.

[2] V. M. Eguiluz, M. G. Zimmermann, C. J. Cela-Conde, and M. S. Miguel. Cooperation and emergence of role differentiation in the dynamics of social networks. *Am. J. Sociol.*, 2006.

[3] J. Kleinberg. Social networks, incentives, and search. In *Proceedings of the 29th ACM SIGIR*, 2006.

[4] E. D. Val, M. Rebollo, and V. Botti. Self-Organized Service Management in Social Systems. In *45 HICCS*, 2012.

Disagreement for control of rational cheating in peer review: a simulation

(Extended Abstract)

Mario Paolucci
Institute of Cognitive Sciences and Technologies
Italian National Research Council
Via Palestro 32, Roma, Italy, 00185
mario.paolucci@istc.cnr.it

Francisco Grimaldo
Departament d'Informàtica
Universitat de València
Av. Universitat, s/n, Burjassot, Spain, 46100
francisco.grimaldo@uv.es

ABSTRACT

Understanding the peer review process could help research and shed light on mechanisms that underlie crowdsourcing. We present an agent-based model of peer review built on three entities - the paper, the scientist and the conference. The model allows us to define a rich model of scoring, evaluating and selecting papers for conferences. Some of the reviewers apply a strategy (called "rational cheating") aimed to prevent papers better than their own to be accepted. We show how programme committee update, based on disagreement control, can remove rational cheaters.

Categories and Subject Descriptors

I.2.0 [**ARTIFICIAL INTELLIGENCE**]: General—*Cognitive simulation*

General Terms

Algorithms, Human Factors, Design

Keywords

Artificial social systems, Peer Review, Agent-based simulation, Trust, Reliability, Reputation, Cognitive Modeling, Rational Cheating

1. INTRODUCTION

Peer review, the process that scrutinizes scientific contributions before they are made available to the community, lies at the core of the social organization of science. Curiously, while the measurement of scientific production, that is, the process that concerns the citation of papers - scientometrics - has been an extremely hot research issue in the last years, we can't say the same for what concerns the process of selection of papers, although some attention has been focused on its shortcomings [3, 2].

Although being extremely important, the actual effectiveness of peer review in ensuring quality has yet to be fully investigated. While the heterogeneous review approach to a decision between two options is supported by Condorcet's jury theorem, if we move beyond simple accept/reject decisions by considering scoring and ranking, we find several kinds of potential failures that are not waived by the theorem. To understand and possibly to apply policies to peer review, we need more evidence coming from both the analysis and review of the process as it is, as well as from the creation of numerical, agent-based models, that could be validated both on the micro and the macro level.

The literature of simulation models about peer review is scarce. In [5], the authors focus on an optimizing view of the reviewer for his/her own advantage. To this purpose, they define a submission/review process that can be exploited by a *rational cheater* [1] strategy in which the cheaters, acting as reviewers, reject papers whose quality would be better than their own. They find out that a small number of rational cheaters reduces rather quickly the process to random selection. In this paper, we propose an more complete agent-based model of peer review and we test how a simple mechanism based on disagreement control could help controlling this kind of cheating.

2. THE PEER REVIEW MODEL

The key entities we identify within the peer review process are: the *paper*, the *scientist* and the *conference*. Thus, the proposed model represents the peer review problem by a tuple $\langle S, P, C\rangle$, where S is the set of *scientists* playing both the role of authors that write *papers* and the role of reviewers that participate in the PC of a set of *conferences* C. Papers produced by scientists have an associated value representing their intrinsic value, and receive a review value from each reviewer. These values are expressed as integers in an N-values ordered scale, ranging from strong reject (i.e. value 1) to strong accept scores (i.e. value N).

Every scientist $s \in S$ is represented by a tuple of the form $s = \langle ap, aq, as, cd, rs, rt\rangle$. Regarding paper production, each scientist has an associated author productivity ap, meaning the number of papers uniformly written per year. The intrinsic value of each paper is calculated considering the author quality $aq \in [1, N]$ and the author skill value $as \in [0, 1]$. The latter represents production reliability, so that scientists write papers of value aq with probability as, and of random value with probability $(1 - as)$. Each scientist also has an associated reviewer skill value $rs \in [0, 1]$ and a reviewing strategy $rt \in \{\texttt{normal}, \texttt{rational}\}$. We model a noisy evaluation of papers, where the result of reviewing is accurate with probability rs, and completely random with probabil-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

ity $(1 - rs)$. Furthermore, *rational* cheaters punish those papers whose intrinsic value is greater than his own author quality (by scoring them with the lowest reviewing value), thus trying to clear the way for their own papers.

Finally, conferences $c \in C$ are represented by a tuple of the form $c = \langle PC, av, I, pu \rangle$. Conferences employ a subset of scientists $PC \subseteq S$ as their programme committee, who accept those papers whose average review value is greater than the acceptance value av. Additionally, conferences maintain an image of each scientist that has ever been a PC member (I), accounting for the disagreements with the other reviewers. Disagreements are calculated on a paper basis as the difference between the review value given by the reviewer and the average review value for that paper. Thus, reviewer images are used to update the PC by discarding the pu% of reviewers with a higher ratio of disagreement. As a response to each call for papers, scientists decide to submit papers provided the distance between the estimated paper value (authors perform on their paper the same noisy evaluation seen before) and the conference acceptance value av is less than or equal to the cautiousness degree of the author, expressed by the integer value cd.

3. RESULTS

Here, we present the results of a set of simulations of the proposed model involving 1000 scientists and 10 conferences across 50 years. Each scientist writes 2 papers uniformly distributed over the year ($ap = 2$). Paper intrinsic values and review values are expressed in a 10-values ordered scale from 1 to 10 ($N = 10$). Authors' qualities (aq) follow a discretized bell shaped curve with mean 5.5 and symmetrically distributed between 1 and 10, in the hypothesis that average papers are more common than either excellent or bogus papers. Authors' skills (as) and reviewers' skills (rs) follow a Uniform distribution in [0.5,1], that we consider a moderate level of noise. With respect to the reviewing type (rt), we only show results with rational cheaters up to 30% since greater ratios reverse the system, ending up with no papers accepted at all. For the conferences, parameters have been set in order to reproduce two different experimental scenarios that we call *homogeneous condition* (i.e. all av are equal to 5.5) and *heterogenous condition* (i.e. av range from 1 to 10). The percentage of PC update is $pu = 10\%$.

Our research hypothesis is that the PC update mechanism proposed will effectively find out and expel the rational cheater scientists. The argument that rational cheaters will find themselves in disagreement with others every time they act strategically makes sense and, in fact, in figure 1 we can observe how rationals cheaters decrease substantially in the conditions where they are more abundant. The PC update mechanism results more effective in the homogeneous condition than in the heterogeneous one (two-sided t test with p-value of 0.036, comparing MR-30 and SR-30), since the PC update mechanism fails in moving rational cheaters away from the PC when the quality of the conference is low.

A deep analysis of the results has been conducted in order to elucidate the effects of reducing the presence of rational cheaters. Though, due to space reasons, we can barely mention some of its main conclusions. The simulations show a significant decrease of the number of papers that should be accepted, but end up being rejected. In turn, as rational cheaters are expelled, the number of accepted papers grows to approach that of conditions without rational cheaters.

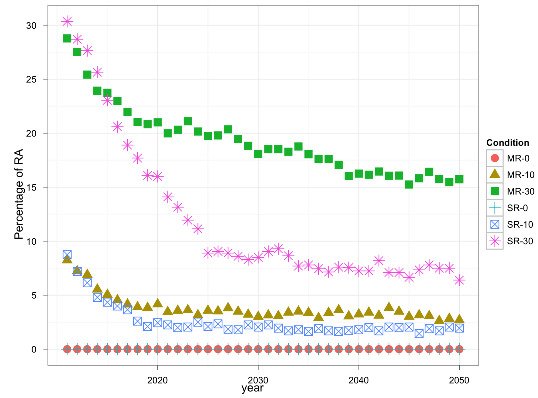


Figure 1: Percentage of Rational Scientists (rational cheaters) under different conditions: homogeneous (SR) and heterogeneous (MR) conditions with initial percentages from none to 30%.

4. CONCLUSIONS AND FUTURE WORK

Results from our simulations show how the mechanism we introduced to control disagreement in the PCs is also effective in removing most of the rational cheaters from the process. The benefit for the system can be measured in terms of the growing number of accepted papers and the decrease in the number of mistakes (good papers rejected).

A next step in this research would be to ground our model against data extracted from one of the several automated conference review systems. However, this data has proven surprising difficult to obtain. Not only the authors' queries to the owners of those systems went unanswered, but we have come to learn that other researchers had the same situation (none of [4, 5] manages to ground their assumption either). The difference between the immediate availability of publication and citation data is especially striking.

5. ACKNOWLEDGMENTS

Work supported by the Spanish MICINN, Consolider Programme and Plan E funds, European Commission FEDER funds and Universitat de València, under Grants CSD2006-00046, TIN2009-14475-C04-04, UV-INV-AE11-40990.62, as well as under the FuturICT coordination action.

6. REFERENCES

[1] D. Callahan. Rational Cheating: Everyone's Doing It. *Journal of Forensic Accounting*, pages 575+, Oct. 2004.

[2] F. Casati, M. Marchese, A. Ragone, and M. Turrini. Is peer review any good? A quantitative analysis of peer review. Technical report, Ingegneria e Scienza dell'Informazione, University of Trento, 2009.

[3] B. D. Neff and J. D. Olden. Is Peer Review a Game of Chance? *BioScience*, 56(4):333–340, Apr. 2006.

[4] F. Squazzoni and C. Gandelli. San Matthew Is Always There. A Study on Peer Review and the Scientific Community Structure. 2011.

[5] S. Thurner and R. Hanel. Peer-review in a world with rational scientists: Toward selection of the average. Aug. 2010.

Sub-delegation and Trust

(Extended Abstract)

Chris Burnett
University of Aberdeen
King's College
Aberdeen, UK
cburnett@abdn.ac.uk

Nir Oren
University of Aberdeen
King's College
Aberdeen, UK
n.oren@abdn.ac.uk

ABSTRACT

Trust mechanisms can allow an agent to identify the most trustworthy entity to which a task should be delegated. Now this entity may further delegate the task, ultimately resulting in a *delegation chain* representing the sub-delegation process. Such *delegation chains* present a problem for current trust evaluation mechanisms, as they typically which reward or penalise a single agent rather than sharing responsibility among all members of the delegation chain. As a result, decisions made on such incorrect trust values would not be optimal, leading to degraded system performance. In this paper we investigate the effects of sub-delegation on a probabilistic trust model and propose a model of weighting trust updates based on shared responsibility. We evaluate this model in the context of a simulated multi-agent system and describe how different weighting strategies can affect probabilistic trust updates when sub-delegation is possible.

Categories and Subject Descriptors

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

General Terms

Theory

Keywords

Trust, Delegation Chains

1. INTRODUCTION

Marsh's seminal thesis [1] identified the existence of an implicit trust relationship in multi-agent systems (MASs), and since then, researchers have investigated mechanisms for computing — and acting based on — different trust levels between agents [3, 4, 6]. Such systems have consistently been shown to improve the overall utility of the MAS, with poorly performing agents quickly garnering a low trust rating, which leads to others minimising their interactions with them, thereby reducing the potential harm such untrustworthy agents can cause to the system.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Trust is critical for the successful *delegation* of tasks in open MASs, whereby one agent requests that some other agent execute a task on the first agent's behalf. Most existing trust mechanisms assume interactions which affect only the truster and trustee, and ignore the possibility of a task being repeatedly delegated from one agent to another. We refer to this sequence of delegations as a *delegation chain*, with the agent originally desiring the execution of the task at the head of the chain, the agent executing the task at its tail, and other, intermediate delegators between them. Such chains appear in a variety of applications (e.g. virtual organisations [2]). The core question we seek to address in this paper is *how the process of delegation (and sub-delegation) should affect trust measures.*

For example, if the agent at the end of a delegation chain fails to achieve the delegated task, all agents in the chain *should* share some of the blame. However, several intuitive ways of apportioning this blame exist, and we seek to investigate the effects of each of these approaches on the system as a whole. In seeking to answer this question our main contribution is to describe and evaluate a model for updating trust in the presence of delegation. Our approach consists of a weighting scheme which discounts the change in trust placed in an agent based on the outcome of a delegated task and the agent's position in the delegation chain.

As an example, consider the situation where Alice asks Bob to book a hotel for her. Bob, being unfamiliar with hotels, asks Charlie to perform the booking. Charlie delegates this request to Debbie, who books a bad hotel, upsetting Alice. Should Alice ever ask Bob to book a hotel for her again? Intuitively, Bob has done nothing wrong; the delegation means that Alice's trust in Bob should be affected to a lesser degree than Bob and Alice's trust in Charlie, and in turn, by Alice, Bob and Charlie's trust in Debbie.

2. APPROACH

We evaluate different weighting measures over a simple model of delegation. Our system consists of a set of agents, each of which are capable of performing some tasks. Agents also have communication links to other agents, and, if they choose not to perform a task, can request that some agent with which they can communicate, perform the task. Now in order to encourage delegation, we assume that agents have different levels of competence in performing different classes of tasks. When an agent must perform some task, they can deal with it in one of three ways, namely 1) decline the task, 2) delegate the task to another agent, or 3) perform the task. Deciding between these three courses of action is done

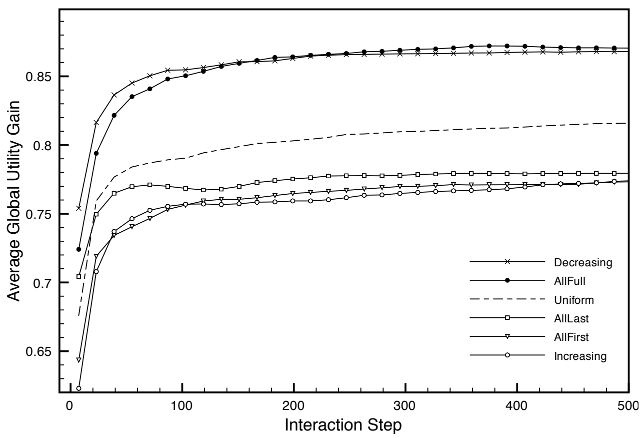


Figure 1: Average global utility gain

based on the agent's own capabilities, and in the trust it has in the agents it communicates with. The same task can be repeatedly delegated, creating a path of agents responsible for its fulfilment, which we refer to as a *delegation chain*.

We use Jøsang's Subjective Logic based trust model [5]. While more complex trust models exist, the use of a relatively straight-forward model simplifies our experiments and allows us to highlight our contribution. Note that we leave the repetitional dimension for future work – agents do not obtain third-party opinions through communication.

Existing approaches to trust apportion blame without taking delegation into account. That is, the trust of the delegator in the delegatee would be updated without taking into consideration any other agent in the delegation chain. In such cases, we argue that it is appropriate to update our trust in the various agents in a delegation chain to different degrees, to reflect the fact that a particular outcome should not reflect equally on all the agents' responsibilities for this outcome, and therefore on their trustworthiness.

We evaluated several different trust update mechanisms, which allow an agent to update its trust rating in all subsequent agents in the chain. These mechanisms weigh trust based on position in the delegation chain, and are as follows: *Uniform Weighting*, where a responsibility is distributed evenly among all in-chain agents; *All-First/-Last Weighting*, where all weight is applied to the first/last agent; *Increasing/Decreasing Weighting*, where increasing/decreasing) proportions of weight are applied applied to consecutive agents in the chain; and *Full Weighting*, where all agents receive full weight. Intuitively, we seek to weigh trust according to an agent's *responsibility for the final outcome*, and evaluate the performance of different models of responsibility.

3. EVALUATION AND DISCUSSION

The different trust update mechanisms were evaluated via simulation. Agents interact over a number of rounds with partners of varying trustworthiness. Figure 1 shows the performance of the system (with respect to global utility) when using different weighting functions.The *Decreasing* and *AllFull* weighting functions appear to perform best.

While it appears attractive, the *AllFull* model is problematic as it is inherently unfair; intermediate agents are penalised (or rewarded) as if they performed the task alone. While this leads to a rapid convergence of performance, strategically minded agents could collude to abuse this approach. For example, agents pass a task around unnecessarily within a group, before finally delegating to a highly trusted individual, so that each agent in the group receives a full positive trust update without having to perform the task. Using the *Decreasing* mode prevents this possibility, as each sub-delegation reduces the weight applied to each agent in the chain. Investigating such strategic aspects of each weighting function will form one area of future work.

An important feature of our approach is that it places few constraints on the particular trust model used, requiring only that the model permits discounted or weighted update. This is already an important feature of many prominent trust models [4], which use discounting to reduce the impact of older experiences on trust assessments, allowing trust models to cope with dynamic behaviour.

Apart from investigating how repetitional information can be included in our model, as future work we intend to examine how partial observability of the delegation chain can be dealt with. We also intend to investigate more complex delegation behaviours between agents (e.g. task splitting).

4. CONCLUSIONS

This research addresses a new and exciting aspect of trust in multi-agent systems, namely how trust should be updated in the context of delegation. Such an approach has many practical applications. In both human and computational domains, one could ask how contractors should trust each other when tasks may be "outsourced" to other parties, and when trustors may be unable to control or observe this outsourcing process. In such situations, our model allows one to apportion responsibility between individuals in a fine grained manner, leading to improved overall system behaviour. We have shown that the choice of weighting function significantly affects the utility of the system. While our approach goes some way towards addressing delegation, it forms only a first step in investigating this aspect of trust, and many exciting avenues of future research remain open.

5. REFERENCES

[1] S. Marsh. Formalizing Trust as a Computational Concept. PhD thesis, University of Stirling, 1994.

[2] J. Patel, W. T. L. Teacy, N. Jennings, M. Luck, S. Chalmers, N. Oren, T. J. Norman,A. Preece, P. Gray, G. Shercliff, P. Stockreisser, J. Shao, W. Gray, N. Fiddian and S. Thompson. Agent-Based Virtual Organisations for the Grid. *Int. Jnl. of Multi-Agent & Grid Systems*, 1 (4). 237-249, 2005.

[3] J. Sabater and C. Sierra. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[4] W. Teacy, J. Patel, N. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[5] A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *Proc. of the 29th Australasian Comp. Sci. Conf.*, pages 85–94, 2006.

[6] Y. Wang and M. P. Singh. Formal trust model for multiagent systems. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence*, pages 1551–1556, 2007.

A Dempster-Shafer Theory Based Witness Trustworthiness Model

(Extended Abstract)

Siyuan Liu[1] Alex C. Kot[1] Chunyan Miao[2] Yin-Leng Theng[3]
[1]School of Electrical and Electronic Engineering, [2]School of Computer Engineering,
[3] Wee Kim Wee School of Communication and Information,
Nanyang Technological University, Singapore, lius0036@ntu.edu.sg

ABSTRACT

The problem of unfair testimonies remains to be a big concern in reputation systems. To address this problem, we propose a witness trustworthiness model based on Dempster-Shafer theory for reputation systems using multi-nominal testimonies. The proposed approach uses Dempster-Shafer theory to model a witness's trustworthiness from both personal and public aspects. Experimental evaluation demonstrates promising results of the proposed approach in modeling witnesses' trustworthiness and adapting to the buyer specified subjective difference tolerance level.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence – Intelligent agents, Multiagent systems

General Terms

Design, Measurement

Keywords

Reputation System, Unfair Testimony, Dempster-Shafer Theory

1. INTRODUCTION

The problem of "*unfair testimonies*" remains to be a big concern in reputation systems. In our previous work [2] [3], we proposed to use clustering to filter unfair testimonies. But the previous approaches cannot exactly indicate how trustworthy the testimonies can be. In this paper, we propose a novel approach based on Dempster-Shafer theory [4] to address the problem of unfair testimonies. The proposed approach uses Dempster-Shafer theory to model a witness's trustworthiness to indicate how trustworthy a witness is by adapting to the subjective difference tolerance level specified by the buyer.

2. THE PROPOSED WITNESS TRUSTWORTHINESS MODEL

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Suppose that there are N sellers $\{S_1, S_2, \ldots, S_N\}$ in a reputation system. Now a buyer B is evaluating a seller S_i's $(1 \leq i \leq N)$ reputation. To facilitate B's evaluation regarding S's reputation, B may request ratings from other buyers who had transactions with S_i before. From B's point of view, these buyers providing ratings regarding S_i are called witnesses, and the ratings provided are called testimonies. Now a new problem arises – how does B know a witness W's testimonies are trustworthy? To address this problem, we propose using Dempster-Shafer theory to model a witness's trustworthiness from both personal and public aspects.

The witness W's personal trustworthiness is evaluated through comparing W's testimonies with B's personal ratings regarding all the sellers. Suppose a transaction between B (or W) and a seller S_i happens at time t. After the transaction is completed, the rating from B (or W) is r^t_{B,S_i} (or $r^t_{W,Si}$), which is a value k from the integer set of $\{1, 2, ...K\}$ (K is the number of rating levels the reputation system adopts). Suppose in a time period $[\mu, \mu + \varepsilon]$, B has a rating vector $R^{\mu,\mu+\varepsilon}_{B,S_i}$ and W has a rating vector $R^{\mu,\mu+\varepsilon}_{W,S_i}$. Then $[\mu, \mu + \varepsilon]$ is partitioned into some consecutive elemental time windows [5]. For each rating r^t_{B,S_i} in $R^{\mu,\mu+\varepsilon}_{B,S_i}$, we find a mapped rating $r^{t'}_{W,S_i}$ in $R^{\mu,\mu+\varepsilon}_{W,S_i}$. The mapped rating should be the rating provided by W at time t' which is closest to time t and in the same elemental window. Then $< r^t_{B,S_i}, r^{t'}_{W,S_i} >$ is called a rating pair. We calculate the difference d as $r^t_{B,S_i} - r^{t'}_{W,S_i}$ for the rating pair. d has a total of $2K-1$ possible values and $-(K-1) \leq d \leq K-1$. We count the number of d happenings as α_d in all elemental windows. According to subjective logic [1], we assign the *B*asic *B*elief *A*ssignment function (BBA) [4] for S_i. After we get the BBAs for the N sellers, we use the Dempster-Shafer combination rule [4] to combine the N BBAs together. Denote the combined BBA as m and corresponding belief function [4] as Bel. The witness W's personal trustworthiness T^{Per}_W is calculated as:

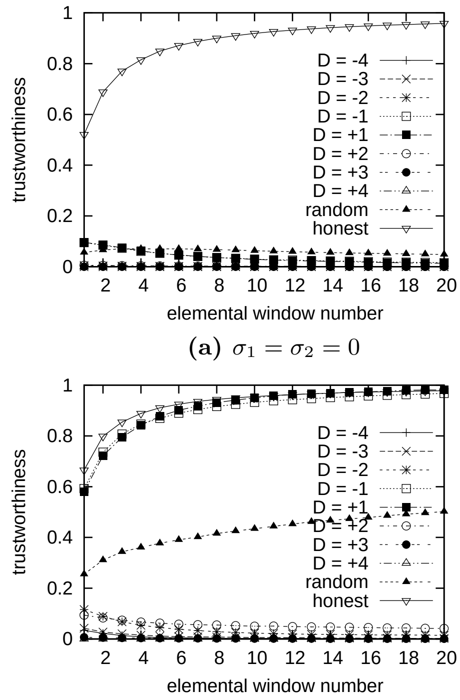
$$T^{Per}_W = Bel(\{d|\sigma_1 \leq d \leq \sigma_2\}) = \sum_{\sigma=\sigma_1}^{\sigma_2} m(\{d\}) \qquad (1)$$

where $-(K-1) \leq \sigma_1 \leq \sigma_2 \leq K-1$. We call $\sigma_1 \sim \sigma_2$ as the buyer's subjective difference tolerance level, meaning the extent of the subjective difference the buyer can tolerate. For example, if $\sigma_1 = -1$ and $\sigma_2 = 1$, it means that the buyer considers the witnesses whose testimonies have -1, 0, and $+1$ difference from the buyer's personal opinions as acceptable and trustworthy.

The witness W's public trustworthiness value is calculated through comparing W's ratings with other witnesses' ratings regarding all the sellers. Suppose there are other L witnesses, $W_1, W_2, \ldots, W_L$, for a seller S_i for whom W provides testimonies. We still partition the time period into some elemental windows. Now we only consider the last rating provided by each witness regarding S_i in each elemental window. Suppose in an elemental window, the last rating provided by the majority witnesses is $r^{last}_{majority,S_i}$ and the last rating provided by W is r^{last}_{W,S_i}. We calculate the difference d' as $r^{last}_{majority,S_i} - r^{last}_{W,S_i}$. d' still has a total of $2K-1$ possible values. By counting the number of d' happenings as $\alpha_{d'}$ in all elemental windows, we have the BBA assignment for S_i. Then we can get the combined BBA and belief function after combining the BBAs for the N sellers. The public trustworthiness T^{Pub}_W is calculated using the similar equation as Eq.(1) in the personal trustworthiness part.

Finally, we calculate the weighted sum of personal trustworthiness and public trustworthiness as the estimation regarding W's trustworthiness. The weights of the personal trustworthiness ω_{per} and public trustworthiness ω_{pub} are assigned based on the uncertainty in the personal trustworthiness part. The more uncertainty in the personal trustworthiness part, the more public trustworthiness is required to be considered. As the last step, the witness W's trustworthiness T_W is calculated as:

$$T_W = \omega_{per} \times T^{per}_W + \omega_{pub} \times T^{pub}_W \quad (2)$$



(a) $\sigma_1 = \sigma_2 = 0$

(b) $\sigma_1 = -1$, $\sigma_2 = 1$

Figure 1: Trustworthiness changes with the number of elemental windows

3. EXPERIMENTAL STUDIES

We simulate an e-commerce environment to investigate the witnesses' trustworthiness using the proposed model. Five rating levels are adopted. We simulate 20 sellers and 51 buyers. From the last buyer's point of view, the first 50 buyers are witnesses. We simulate two types of unfair witnesses. The first type is D-shifting witnesses who report real rating adding D rating level, where D is from the value set $\{-4, -3, -2, -1, 1, 2, 3, 4\}$. The second type is random witnesses who report a randomly selected rating level except the real rating. We simulate 2000 time units (a time unit can be a minute, an hour, a day..., depending on different reputation systems) and run 100 rounds for each simulation scenario to achieve a statistical accuracy. For each buyer's transaction, a seller is randomly selected. The rating for each transaction is simulated from a normal distribution.

Figure 1 shows the witnesses' trustworthiness changes with the number of elemental windows when the length of an elemental window is 100 time units and there are 30% unfair witnesses. Figure 1(a) and (b) show the results when the subjective difference tolerance level is set as $\sigma_1 = \sigma_2 = 0$ and $\sigma_1 = -1$ and $\sigma_2 = 1$, respectively. According to the results, the witnesses' trustworthiness value will stabilize after about 10 elemental windows which are 1000 time units. When $\sigma_1 = \sigma_2 = 0$, only the 0-shifting witnesses can get a high trustworthiness value. When $\sigma_1 = -1$ and $\sigma_2 = 1$, the -1-shifting and 1-shifting witnesses can also get a high trustworthiness value. Therefore, the buyer can use σ_1 and σ_2 to indicate the subjective difference tolerance level that is acceptable.

4. CONCLUSIONS

In this abstract, we proposed a witness trustworthiness model based on Dempster-Shafer theory to address the problem of unfair testimonies in reputation systems. Our approach models a witness's trustworthiness from both personal and public aspects. It supports reputation systems using multi-nominal rating levels, and provides buyers a great extent of flexibility to identify the trustworthy witnesses by specifying their own subjective difference tolerance level. Experimental results show that the proposed approach can effectively model witnesses' trustworthiness and adapt to the buyer's specified subjective difference tolerance level.

5. REFERENCES

[1] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, 2001.

[2] S. Liu, C. Miao, Y.-L. Theng, and A. C. Kot. A clustering approach to filtering unfair testimonies for reputation systems (extended abstract). In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 1577–1578, 2010.

[3] S. Liu, J. Zhang, C. Miao, Y.-L. Theng, and A. C. Kot. iclub: An integrated clustering-based approach to improve the robustness of reputation systems (extended abstract). In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 1151–1152, 2011.

[4] G. Shafter. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[5] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: The problem of unfair ratings. In *Proceedings of the Eighth International Conference on Electronic Commerce (ICEC'06)*, 2006.

Detecting and Identifying Coalitions

(Extended Abstract)

Reid Kerr
University of Waterloo
Waterloo, Ontario, Canada
rckerr@cs.uwaterloo.ca

Robin Cohen
University of Waterloo
Waterloo, Ontario, Canada
rcohen@cs.uwaterloo.ca

ABSTRACT

In multiagent scenarios, subsets of a population (*coalitions*) may attempt to cooperate, for mutual benefit. We present a technique for detecting the presence of coalitions (malicious or otherwise) and identifying their members, and demonstrate its effectiveness.

Categories and Subject Descriptors

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

General Terms

Experimentation, Security

Keywords

Coalitions, Collusion, Trust and Reputation, Multiagent Systems

1. INTRODUCTION/RELATED WORK

In multiagent systems, groups of agents (*coalitions*) may seek to coordinate their activities in some way, to further their goals; where such activity is unwelcome, it may be called *collusion*. Coalitions represent a persistent and pervasive problem for many multiagent systems. Despite this, there has been little progress towards a solution. Here, we present a technique for detecting coalitions in an environment, and for identifying coalition members. Detection might, e.g., allow remediation, or might serve as a deterrent.

Because our approach is based on the concept of benefit rather than on domain-specific features, and because it requires no knowledge of the plans in use, we believe it to be applicable to a wide variety of domains: e.g., cheating in games, 'shilling' or 'astroturfing', or insurgent activity. Here, we apply our technique to trust and reputation systems for marketplaces, where two forms of collusion are well-known problems: *ballot-stuffing* (false positive reviews, to inflate teammates' reputations), and *bad-mouthing* (false negative reviews, to damage competitors' reputations). Both attacks seek to improve team members' chances of being selected by other agents. We demonstrate strong detection performance, with excellent resistance to false positives. As such, this work represents an important step towards addressing the challenges posed by coalitions.

Key characteristics of the scenarios of interest should be noted. First, we have no knowledge of communication or sharing of resources by coalition members outside the system. Second, and importantly, we assume no knowledge of the plans in use.

While several areas of research share some relation to our problem, the work in each targets fundamentally different scenarios than our own. Coalition formation and stability (e.g., [4]) assumes, for example, that the capabilities of agents, and payouts, are known. Similarly, work in multiagent plan/behavior recognition (e.g., [5]) assumes known plan libraries. Community finding, in social networks (e.g., [3]), typically uses metrics (e.g., connectivity, frequency of interaction) that are of limited value for our problem.

2. METHOD

Because we have no access to a plan library, our method must rely on fundamental properties of the observable actions themselves. In particular, self-interested agents belong to coalitions because they expect to improve their benefit (or reduce harm done to them). We might expect that coalition members are more likely to help one another than to help outsiders, and/or more likely to harm outsiders than to harm one another. The important insight is that because coalition members favor the same set of agents (each other), there is likely similarity in terms of the agents they benefit, and harm.

Our technique is a two step process. First, we identify 'candidate' sets of agents; second, we characterize each candidate group as either a coalition, or not.

We define the *benefit space* as a high-dimensional space reflecting the degree of benefit (and harm) rendered to each agent in the system. This is a key insight—the benefit space formulation allows possible coalitions to be detected using existing tools such as clustering. Specifically, given N total entities in the system, the benefit space $\mathcal{B}$ is a space $\mathbb{R}^N$, where the value in each dimension β_i represents an amount of net benefit (i.e., total benefit minus total harm) to entity i. Each entity maps to a point in the benefit space, reflecting the amount of (observable) net benefit it has rendered to each entity in the system. Because members of a coalition are likely to be similar in terms of the sets of agents that they favor, we would expect them to be close in this benefit space. Using Euclidian distance as our dissimilarity measure, we have used k-means clustering to partition the population P into a set of clusters $\{C_1, C_2, ..., C_n\}$, each of which is a *candidate coalition*.

Similarity does not *necessarily* imply that a set of agents is a coalition; for example, agents may simply have similar preferences, so they select the same sellers. Thus, we must characterize each candidate cluster to determine if it is, in fact, a coalition. We might expect a true coalition T to be more 'self-serving' (i.e., benefiting each other more than outsiders) than a 'non-coalition' group G. In this case, we would expect the benefit flowing *from* members of T *to* members of T to be greater than the benefit flowing from members of G to members of G. (Similarly, we might expect a coalition

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

to damage outsiders more than a 'normal' group would. The discussion of this is omitted, for brevity.) Consider any given set of agents S, where $m = |S|$. There are $m(m-1)$ (directed) relationships between agents in S. The average benefit (per relationship) flowing *from* agents in S, *to* agents in S, then, is:

$$\bar{\beta}_S = \frac{\sum_{i \in S} \sum_{j \in S, j \neq i} \beta_j(i)}{m(m-1)} \tag{1}$$

Using Formula 1, we can find $\bar{\beta}_C$, the average benefit within C. To know whether the computed value is abnormally high, we need a benchmark to which to compare it. For this, we take random samples of m agents (drawn from the entire population P, including agents in C). For each sample G, we compute $\bar{\beta}_G$, using Formula 1. Doing so over a large number of samples, we estimate the mean and standard deviation over $\bar{\beta}_G$. With this, we can estimate the probability of obtaining a measure as high as $\bar{\beta}_C$ by chance, using the normal distribution. If this probability is too low (i.e., below α, a parameter), we conclude that members of C abnormally benefit one another; we label all agents in C as coalition members.

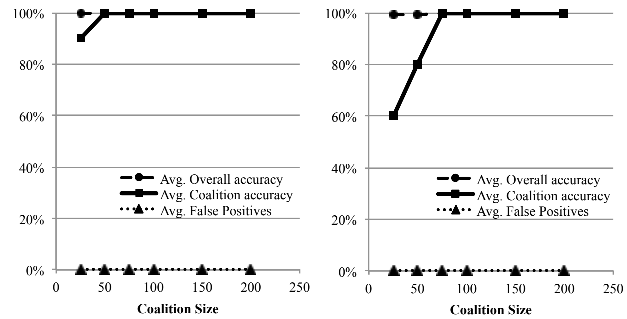
3. EXPERIMENTAL SCENARIO/RESULTS

Real-world colluders do not willingly reveal themselves as such, making it problematic to obtain real-world, labelled data that might be used for validation. Thus, the TREET marketplace testbed [2], populated by buying and selling agents, was used to validate our technique. Populations of 1000 agents made use of the Beta Reputation System [1]. Coalitions attempted to improve profits by bad-mouthing or ballot-stuffing. For each combination of parameter values, 10 trials were run (except where noted); the figures reported reflect the aggregate results across trials. The measure of benefit used to detect coalitions was the net sum of the review values given (counting a positive review as $+1$ and a negative review as -1), weighted by the dollar value of the transaction. After applying our technique, our classifications were compared to the true, hidden class of each agent to determine accuracy.

In the first set of tests, we evaluate the technique where exactly one coalition is present in the population. First, we consider coalition members engaged in bad-mouthing. These results are shown in Figure 1a, which contains three series. 'Avg. Overall accuracy', shows the percentage (across all trials) of agents that were accurately labelled as either coalition members or non-members. This metric can be misleadingly high, however, especially when the number of colluders is low. The second series, 'Avg. Coalition accuracy', depicts the fraction of coalition members that were accurately labelled as such. (This is equivalent to *recall*.) This shows that some colluders were missed for the smallest coalition size, but in general, performance is excellent. The third series, 'Avg. False Positives' shows the number of non-coalition members that were mistakenly identified as coalition members. (This value is equal to $1-$ *precision*.) This was zero, in all trials. Results for the ballot-stuffing case are depicted in Figure 1b; performance is slightly weaker, but very strong.

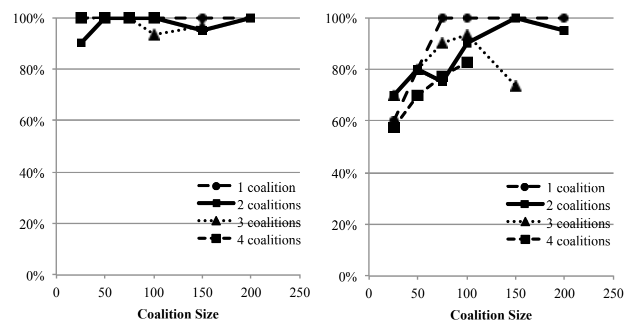
While performance is strong with exactly one coalition, it may be the case that there is no coalition present in a given population. Such situations provide a good test of the algorithm's resistance to false positives. We ran 120 trials with zero coalitions. In total, 3 agents were wrongly labelled as coalition members (a rate of 0.000025).

Just as a population might contain no coalitions, it might also contain multiple coalitions. We ran trials with up to 4 coalitions. The results for bad-mouthing are displayed in Figure 2a; those for ballot-stuffing are shown in Figure 2b. For clarity and brevity, false



(a) Bad-mouthing. (b) Ballot-stuffing.

Figure 1: Coalition detection accuracy, single coalition.



(a) Bad-mouthing. (b) Ballot-stuffing.

Figure 2: Coalition detection accuracy, multiple coalitions.

positive rates have been omitted from these charts. Again, they were zero in the vast majority of cases, and very low in the others.

Overall performance is quite strong, in all cases. As in the single-coalition cases, performance is somewhat better for bad-mouthing than for ballot-stuffing; similarly, the general pattern of weaker performance on smaller coalitions is again evident in the ballot-stuffing data. Perhaps most importantly, note that there is no clear correlation between number of coalitions and performance: increasing the number of coalitions does not have the detrimental impact on performance that one might expect.

4. REFERENCES

[1] A. Jøsang and R. Ismail. The beta reputation system. 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, June 2002.

[2] R. Kerr and R. Cohen. TREET: The Trust and Reputation Experimentation and Evaluation Testbed. *Electronic Commerce Research*, 10(3):271–290, 2010.

[3] B. Mehta and W. Nejdl. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction*, 19(1-2):65–97, 2009.

[4] O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents in nonsuperadditive environments. *Computational Intelligence*, 15(3):218–251, 1999.

[5] G. Sukthankar and K. Sycara. Robust recognition of physical team behaviors using spatio-temporal models. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 638–645, New York, NY, USA, 2006. ACM.

SARC: Subjectivity Alignment for Reputation Computation (Extended Abstract)

Hui Fang Jie Zhang Murat Şensoy* Nadia M. Thalmann

School of Computer Engineering, Nanyang Technological University, Singapore, {hfang1}@e.ntu.edu.sg
*Department of Computing Science, University of Aberdeen, United Kingdom

ABSTRACT

Current deployed reputation systems simply aggregate numerical ratings provided by buyers, but overlook the buyers' subjectivity difference in evaluating the transactions with a seller. To address this problem, we propose a subjectivity alignment approach for reputation computation (SARC).

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents; K.4.4 [**Electronic Commerce**]: Trust, Reputation

General Terms

Algorithms; Design

Keywords

Subjectivity Alignment; Reputation System; Bayesian Learning; Intelligent Buying Agent

1. INTRODUCTION

Reputation systems [3] have been proposed to model the trustworthiness of sellers in e-marketplaces where buyers who previously bought products from a seller share their experience, normally in the form of a numerical rating. These ratings are aggregated to represent the seller's reputation. However, a rating is subjective evaluation of a seller by a buyer within the context of a specific transaction. Different ratings could be given for the same transactions by different buyers. Two aspects contribute to the subjectivity difference among buyers: 1) *intra-attribute subjectivity*, the subjectivity in evaluating the same attribute of a transaction; 2) *extra-attribute subjectivity*, the subjectivity in evaluating different attributes of a transaction.

To address the subjectivity difference issue, we propose a subjectivity alignment approach for reputation computation (SARC). In SARC, buyers' subjectivity is learned based on the ratings and detailed reviews they provide about the objective attributes of their transactions with sellers. More specifically, SARC separately learns the *intra-attribute subjectivity* and *extra-attribute subjectivity* of buyers. Buyers' *intra-attribute subjectivity* is modeled using Bayesian learning. Their *extra-attribute subjectivity* is learned using a regression analysis model. Ratings provided by one buyer can then be aligned (converted) for another buyer according to the two buyers' subjectivity.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. THE SARC APPROACH

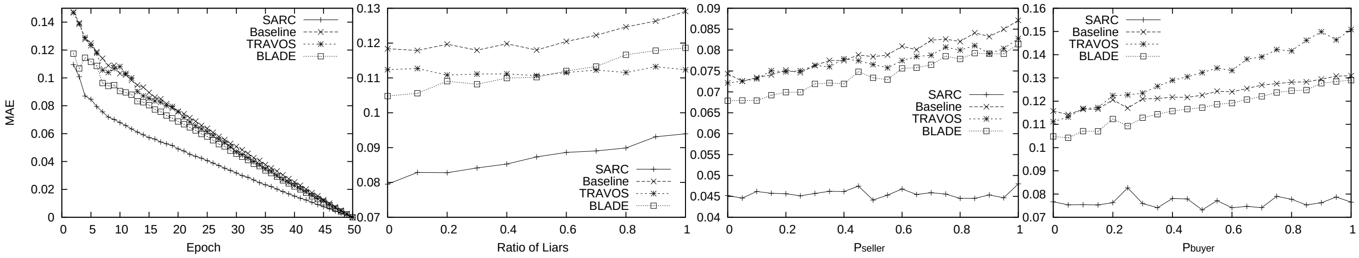
In an e-marketplace, each buyer is equipped with an intelligent (buying) agent. We denote the set of buyers by $\mathcal{B} = \{b_1, b_2, \ldots\}$. The set of agents equipped by corresponding buyers is denoted by $\mathcal{A} = \{a_1, a_2, \ldots\}$, and the set of sellers are referred to as $\mathcal{S} = \{s_1, s_2, \ldots\}$. The set of objective attributes for describing a transaction between a buyer and a seller is denoted as $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$. Each rating provided by a buyer for a seller is from a set of predefined discrete rating levels $\mathcal{L} = \{r_1, r_2, \ldots, r_n\}$. For a buyer $b_i \in \mathcal{B}$, the goal of her buying agent $a_i \in \mathcal{A}$ is to accurately compute the reputation value of a target seller $s_j \in \mathcal{S}$, according to b_i's subjectivity. To achieve the goal, a_i needs to consider the ratings of other buyers (advisors) that evaluate the satisfaction levels about their past transactions with seller s_j. Due to the possible subjectivity difference between buyer b_i and the advisors, agent a_i also needs to align/convert ratings of each advisor (for example b_k) using our SARC approach.

More specifically, at the beginning of buyer b_i's interactions with the system, agent a_i asks b_i to provide a rating for each of her transactions with a seller (which can be any seller in $\mathcal{S}$). Buying agent a_i also asks b_i to provide detailed review information about each transaction containing the values of the set of objective attributes in $\mathcal{F}$. Based on the provided information (rating-review pairs), agent a_i models a set of correlation evaluation functions (CEFs) for buyer b_i, capturing b_i's *intra-attribute subjectivity*. Each correlation evaluation function is represented by a *Bayesian conditional probability density function* that models the correlation between each rating level and each objective attribute:

$$\mathrm{CEF}_{u,v}^{b_i} = p^{b_i}(f_u \mid r_v) = \frac{p^{b_i}(r_v \mid f_u) \times p^{b_i}(f_u)}{p^{b_i}(r_v)} \qquad (1)$$

where $\mathrm{CEF}_{u,v}^{b_i}$ is the correlation function between attribute $f_u \in \mathcal{F}$ and rating level $r_v \in \mathcal{L}$ for buyer b_i; $p^{b_i}(r_v)$ refers to the probability that buyer b_i provides a rating r_v; $p^{b_i}(f_u)$ is the probability distribution of the values for attribute f_u, and $p^{b_i}(r_v \mid f_u)$ is the conditional probability of rating level r_v given the distribution of the values for attribute f_u.

The learned CEFs of buyers will be shared with each other buyer's agent. For a rating provided by the buyer (advisor) b_k, agent a_i can then derive a rating for each attribute $f_u \in \mathcal{F}$, based on the CEFs shared by b_k's agent a_k and those of buyer b_i's own. We use a Naïve Bayesian Network model to learn the mapping from r^{b_k} of buyer b_k to the ratings of b_i for the attributes. Take any $f_u \in \mathcal{F}$ as an example attribute, agent a_i first estimates the conditional probability of a rating level in $\mathcal{L}$ for attribute f_u, given rating r^{b_k} provided by buyer



(a) **(b)** **(c)** **(d)**

Figure 1: (a) Performance Comparison in the Basic Environment; (b) Performance When Varying Ratio of Lying Buyers; (c, d) Performance for Sellers' Changing Behavior and Buyers' Changing Subjectivity

b_k. Take any rating level r_v as an example, a_i computes $p^{b_i}(r_{v,f_u}|r^{b_k})$, the conditional probability that buyer b_i will assign the rating level r_{v,f_u} to attribute f_u given the rating r^{b_k} of buyer b_k:

$$p^{b_i}(r_{v,f_u}|r^{b_k}) = \frac{p^{b_i}(r_v \mid f_u) \times p^{b_k}(f_u \mid r^{b_k})}{p^{b_i}(f_u \mid r_v)} \tag{2}$$

where $p^{b_k}(f_u \mid r^{b_k})$ is learned by agent a_k of buyer b_k using Equation 1 and shared by agent a_k to agent a_i, $p^{b_i}(f_u \mid r_v)$ is learned by a_i itself using Equation 1, and $p^{b_i}(r_v \mid f_u)$ is obtained by agent a_i from the rating-review pairs provided by its buyer b_i. What is derived for f_u is a set of probability values, each of which corresponds to a rating level in $\mathcal{L}$. The rating level with the highest probability will be chosen as the rating for f_u, $r^{b_i}_{u,k}$.

Based on the provided rating-review pairs by b_i, agent a_i also learns the *extra-attribute subjectivity* of buyer b_i, which is represented by a set of weights for corresponding attributes in $\mathcal{F}$. The weight of f_u is determined by two factors: 1) the probability value of the rating derived earlier, C_u; and 2) the importance of the attribute learned using a regression analysis model, I_u. These weights will not be shared with other buyers. Once they are learned, the aligned rating ($r^{b_i}_k$) from that of advisor b_k can be computed as the weighted average of the derived ratings for the attributes:

$$r^{b_i}_k = \frac{\sum_{u=1}^{m} r^{b_i}_{u,k} \times C_u \times I_u}{\sum_{u=1}^{m} C_u \times I_u} \tag{3}$$

3. EVALUATION

We simulate an e-marketplace involving 50 sellers and 200 buyers. Sellers may provide different products with different attribute values. Buyers may have different subjectivity in evaluating their transactions with (the products of) sellers. We also set several important parameters for our simulations, including *information availability*, *dynamic behavior of sellers*, *dynamic subjectivity of buyers*, *ratio of liars* (dishonest buyers), and *granularity of rating scale*. We vary the values of these parameters to simulate basic, deceptive and dynamic environments, respectively. In the experiments, we compare our approach with some representative competing approaches: a baseline approach without subjectivity alignment, TRAVOS [2] and BLADE [1].

In the basic environments without deception, seller dynamic behavior or buyer dynamic subjectivity, SARC can more accurately model sellers' reputation than the other three approaches (Figure 1(a)). We also test some parameters including the ratio of objective attributes, the number of detailed reviews, the granularity of rating scale, and the ratio of shared interactions. We find that in different settings, SARC still has better performance than BLADE. In the deceptive environments where some buyers may intentionally lie about their past experience with sellers (Figure 1(b)), SARC still performs much better than the other approaches. It is not dramatically affected by buyers' deception because it treats deceptive buyers as the ones with different subjectivity, and aligns the ratings from them effectively. In the dynamic environments where sellers may change their provided products (Figure 1(c)), SARC performs consistently and is independent of sellers' behavior change. The performance of other three approaches gets worse as sellers become more probably to change their behavior. When buyers may vary their subjectivity during a certain period of their interactions with sellers, Figure 1(d) shows that SARC continues to perform positively, while the performance of BLADE gets closer to the baseline approach, and TRAVOS performs worse than the baseline approach as P_{buyer} increases.

4. CONCLUSIONS

We proposed a subjectivity alignment approach for reputation computation, SARC, to address the subjectivity difference problem. It performs better than the other three approaches, and can more accurately and stably model sellers' reputation. It is capable of coping with environments with deception and dynamic buyer and seller behavior. The requirement of detailed reviews and objective attributes is not very restrictive. For future work, we will conduct experiments on real data to further verify the robustness and efficiency of SRAC in addressing the subjectivity difference problem for reputation computation.

5. ACKNOWLEDGEMENT

This work has been made possible thank to the Institute for Media Innovation at Nanyang Technological University who has given a scholarship to the first author.

6. REFERENCES

[1] K. Regan, P. Poupart, and R. Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2006.

[2] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[3] J. Zhang and R. Cohen. Evaluating the trustworthiness of advice about selling agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, 7(3):330–340, 2008.

The Impact of Social Placement of Non-Learning Agents on Convention Emergence

(Extended Abstract)

Nathan Griffiths
Department of Computer Science
University of Warwick
Coventry, CV4 7AL, UK
nathan.griffiths@warwick.ac.uk

Sarabjot Singh Anand
Department of Computer Science
University of Warwick
Coventry, CV4 7AL, UK
s.s.anand@dcs.warwick.ac.uk

ABSTRACT

Social conventions are important for establishing and maintaining coordination in groups of agents, especially where there is no centralised control. As individuals interact, learn, and update their strategies, effective coordination can be achieved through the emergence of suitable conventions. In this paper we (i) show how the structure of a population affects convention emergence, (ii) demonstrate how fixed strategy agents can manipulate emergence, and (iii) evaluate strategies for inserting fixed strategy agents.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Conventions, Norms, Emergence, Social Influence

1. INTRODUCTION

Social conventions are behaviours or strategies that are generally accepted in a society as describing how to act in a particular situation, and effective conventions can facilitate effective coordinated action. Where centralised control is lacking, conventions can emerge from the local interactions and observations of self-interested individuals [1]. This is a form of *social learning* in which individuals learn from repeated interactions with multiple agents in the population. Many previous investigations assume that agents can perceive the actions, strategy and payoffs of those with whom they interact. Although sometimes possible, in general we cannot make such an assumption, and so we *limit an agent's perception to knowledge of its own payoff.* There has been little exploration of settings in which observations are restricted in this way, with some notable exceptions such as the work of Sen *et al.* [2, 3] and Villatoro *et al.* [4]. Previous work on convention emergence has also typically given little consideration to the importance of the network topology that constrains interactions, the size of the action space (i.e. the number of possible actions, or candidate conventions), the effect of previous interactions on the rewards received, and the effect of fixed strategy non-learning agents.

Where agents in a population learn and adapt based on interactions with others, inserting a small number of non-learning individuals can influence the direction in which the population evolves [2]. In this paper we investigate the effect of fixed strategy (FS) agents on convention emergence, while addressing some of the limitations of previous work.

2. THE SOCIAL LEARNING MODEL

We consider agents that are situated in a network topology, with agents' interactions being restricted to their neighbours. Many previous investigations have considered completely connected or regular networks. However, in most social networks the degree distribution of nodes is typically highly skewed, with a few nodes having an unusually high degree. In this paper we explore topologies that represent properties observed in real-world environments, namely *scale-free* and *small-world* networks, along with *random* networks as a base case for comparison[1].

We modify the interaction game defined by Villatoro *et al.* [4] to support m actions ($m > 2$). The reward for an interaction depends on the current and previous choices, modelling the social pressure that arises from the history of interactions. Each agent x has a fixed length FIFO memory M_x recording the most recent l actions that it has selected. Each time step each agent randomly selects one of its neighbours, and both agents choose which of the m actions they will take. If an agent selects the majority action, as represented in the combination of the two memories, then its payoff is equal to the proportion of the majority actions that it was responsible for, otherwise it receives nothing. Specifically, when an agent x interacts with another y, the reward r_x it receives for action a_x is given by:

$$r_x = \begin{cases} \frac{M_x^{a'}}{M_x^{a'}+M_y^{a'}}, & \text{if } a_x = a' \\ 0, & \text{otherwise} \end{cases}$$

where $M_x^{a'}$ is the number of times action a' appears in agent x's memory and a' is the majority action. An agent's per-

[1]We use the generator implementations provided by JUNG (v2.0.1): http://jung.sourceforge.net/

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

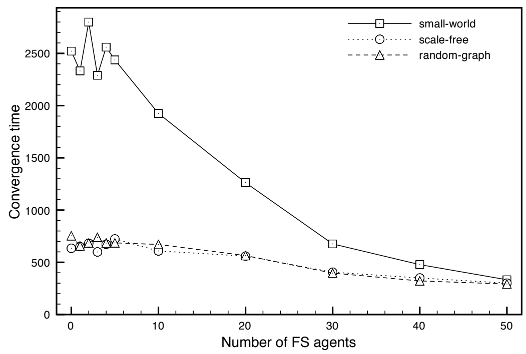


Figure 1: Time for convergence with random, scale-free and small-world topologies using an action space of size $m = 2$ and random placement of FS agents.

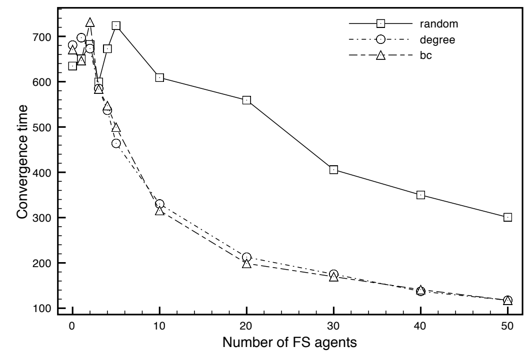


Figure 2: Time for convergence using FS agents placed according to random, degree, and bc with $m = 2$ in a scale-free topology.

ception is restricted to the payoff received in an interaction; agents cannot observe others' actions, memories, or payoffs.

In order to select an action agents use a learning algorithm to estimate the desirability of each possible action. We adopt the approach taken by Villatoro *et al.* [4] of using a simplified Q-Learning algorithm. For each action $a \in A$ each agent maintains an estimate of the utility of choosing that action (a Q-value), which is updated according to: $Q^t(a) = (1 - \alpha) \times Q^{t-1}(a) + \alpha \times reward$ where $Q^t(a)$ is the estimated utility of action a after selecting it t times, α is the learning rate, and $reward$ is the payoff received from the current interaction. With some probability $p_{explore}$ an agent will explore by selecting an action at random, otherwise it selects the action that has the highest Q-value.

In this setting we consider the effect of non-learning *fixed strategy* FS agents, which are each given one of the possible actions as a fixed strategy. We explore two alternatives: (i) all FS agents have the same strategy, with the motivation of reducing the convergence time and (ii) each of the m-actions are uniformly distributed among the FS agents, with the motivation of slowing convergence and maintaining diversity.

3. EXPERIMENTAL RESULTS

In the simulations described below we use a learning rate of $\alpha = 0.5$ and an exploration probability of $p_{explore} = 0.25$. The Q-values for each action are initialised to zero, and each agent's memory is of length $l = 5$ and is initially empty. We use a population of $N = 500$ agents (we see similar trends for $N = \{100, 1000\}$). Each topology was generated to have approximately the same number of edges (1500), using the following parameters: (i) random-graph: $p = 0.012$, (ii) scale-free: $v = 25$ and $e = 3$, and (iii) small-world: $c = 1$ and $\alpha = 2.0$. We adopt Kittock's convergence criteria [1], considering the population to have converged when 90% of the regular agents (non-FS agents), when not exploring, select the same action. Simulations are run for 10000 learning steps, and results are averaged over 100 simulation runs.

Figure 1 shows the effect of the network topology on convergence time as the number of fixed strategy agents increases. In all cases, convergence time reduces as the number of fixed strategy agents increases. Interestingly agents in a small-world network converge to a single convention at a much slower rate than those in scale-free or random graphs. While in the absence of fixed strategy agents the difference in convergence time between small-world and other networks is largest, the convergence times tend to become more similar as the number of fixed strategy agents increases. The difference is insignificant once the number of fixed strategy agents reaches 50, while in the absence of FS agents small-world networks take approximately four times as long to converge when compared with random and scale-free topologies.

Figure 2 shows the results, for a scale-free topology, of placing FS agents by degree and betweenness centrality (bc), along with the baseline random placement strategy as used in Figure 1. As with random placement, increasing the number of fixed strategy agents decreases the convergence time when using degree and bc for placement. Once the number of FS agents is greater than 5 or 6, the degree and bc strategies outperform random placement. The difference in performance between degree and bc is insignificant for random and scale-free topologies, while for small-world degree outperforms bc once the number of FS agents is greater than 5 (graphs for random and small-world topologies are omitted due to space). This is explained by the values for Pearson's correlation between degree and bc for agents within random, scale-free and small-world networks of 0.95, 0.95 and 0.79 respectively, meaning that the same agents are typically selected by degree and bc in random and scale-free networks.

We have performed further experiments that show increasing the size of the action space m increases the time for convergence, and that this increase is fairly consistent across topologies. We have also explored the impact of FS agents having different fixed strategies, and our results show that giving FS agents different strategies can be effective in delaying convergence, with scale-free and random topologies being more manipulable than small-world.

4. REFERENCES

[1] J. E. Kittock. Emergent conventions and the structure of multi-agent systems. In *Lectures in Complex systems*, pages 507–521. Addison-Wesley, 1993.

[2] P. Mukherjee, S. Sen, and S. Airiau. Norm emergence under constrained interactions in diverse societies. In *AAMAS-08*, pages 575–582, 2008.

[3] S. Sen and S. Airiau. Emergence of norms through social learning. In *IJCAI-07*, pages 1507–1512, 2007.

[4] D. Villatoro, S. Sen, and J. Sabater-Mir. Topology and memory effect on convention emergence. In *WI-IAT-09*, pages 233–240, 2009.

Handling Change in Normative Specifications

(Extended Abstract)

Duangtida Athakravi, Domenico Corapi, Alessandra Russo
Department of Computing,Imperial College London
{da407, d.corapi, a.russo}@imperial.ac.uk

Marina De Vos, Julian Padget
Dept. of Computer Science,
University of Bath
{mdv,jap}@cs.bath.ac.uk

Ken Satoh
National Institute of
Informatics
ksatoh@nii.ac.jp

ABSTRACT

Normative frameworks provide a means to address the governance of open systems, by offering a mechanism to express responsibilities and permissions of the individual participants with respect to the entire system without compromising their autonomy. Careful design is crucial if it is to meet its requirements. Tools that support the design process can be of great benefit. In this paper, we describe a method for choosing the appropriate change in the normative specification, using impact analysis of the critical consequences being preserved or rejected by the change.

Categories and Subject Descriptors

H.4 [**Programming Techniques**]: Logic Programming

General Terms

Theory, Verification, Algorithms, Design

Keywords

Design, Normative systems, Inductive Logic Programming

1. INTRODUCTION

Normative frameworks provide a powerful tool for governing open systems by providing guidelines for the behaviour of the individual components without regimentation [4]. Using a formal declarative language to specify the behaviour of a normative system gives the system's designer a means to verify the compliance of the system with respect to desirable behaviours or properties [2, 1]. However, when errors are detected, the identification of what changes to make is often a difficult and error-prone manual process. Corapi et al [3] have shown how Inductive Logic Programming (ILP) can support the elaboration of normative specifications, modelled in Answer Set Programming (ASP), by learning possible changes that would make partial normative specification consistently compliant with given use-cases.

This paper addresses the problem of how to choose between alternative changes by analysing their impact on the specification. We use a notion of *relevant literals*, i.e. elements of the domain that are critical for discriminating between the suggested changes. We describe how these literals are computed using test generation [5]. Ranking them is based on the number of changes that they would discard. This ranking ensures that the revisions are discriminated against in the most effective way.

2. HANDLING CHANGE

Our framework for handling change during the elaboration of normative specifications addresses the limitation of our previous work [3], where it is the designer's responsibility to choose the most appropriate revision from multiple possible revisions computed by a learner. In real applications the number of suggested changes can be large, making an automated criteria essential for selecting the most effective change. As shown in Figure 1, our framework combines the approach in [3], with two additional steps for computing and scoring relevant literals. The most highly ranked literal is then queried to the designer, who can then specify its truth value. Based on the answer of the designer, those changes that are refuted by the relevant literals are discarded. Literals that are dependent on the highly ranked one could be used to further reduce the hypothesis space.

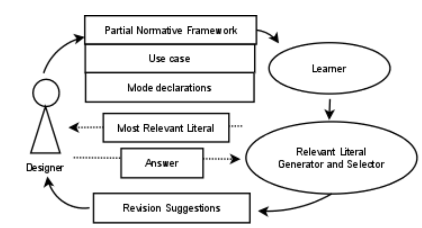


Figure 1: Framework for handling change

2.1 Abducing Relevant Literals

Our definition of relevant literal is based on the notion of relevant test given in [5].

Definition 1. (*Relevant Literal*) Let $\langle T, O\rangle$ be a use-case consisting of a trace T and desired outcome O, given the existing (partial) normative specification Σ, and the set of hypothesis representing the suggested revisions HYP. A literal l is relevant if:

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1. $\Sigma \wedge T \wedge O \wedge H$ is satisfiable for all $H_i \in HYP$
2. $T \wedge O \wedge l$ is an abductive explanation for $\bigvee_{H_i \in HYP} \neg H_i$
3. $\Sigma \wedge T \wedge O \not\models \bigvee_{H_i \in HYP} \neg H_i$
4. $T \wedge O \wedge l$ is not an abductive explanation for $\neg H_i, \forall H_i \in HYP$

Conditions (1) and (3) state that all suggested revisions satisfy the use-case and are consistent with the normative specification. These are guaranteed by the correctness of the learner. Conditions (2) and (4) ensure that some but not all suggested revisions are rejected by the relevant literal. These are captured by integrity constraints used as goals of the following abductive program.

Let N_B be the static part of the normative specification, N_T the part containing rules that are revisable, $\langle T, O\rangle$ the use-cases applied for learning suggested revisions S, and let $C_H/2$ be the function that combines N_T with suggested revisions by representing them as hypotheses. The relevant literals are solutions of the abductive task $\langle B, Ab, G\rangle$ where:

$$B = N_B \cup T \cup C_H(N_T, S)$$
$$G = O \cup \neg(\bigwedge_{H_i \in C_H(N_T,S)} \neg H_i) \cup \neg(\bigwedge_{H_i \in C_H(N_T,S)} H_i)$$

and Ab is the set of ground instances of (possible) outcomes. The set E of relevant literals is a subset of Ab such that $B \cup E \models G$.

2.2 Ranking Relevant Literals

Ideally we want to be able to dismiss as many suggested revisions as possible, based on the truth value of relevant literals. We use the number of minimum hypotheses (or revisions) that a relevant literal may reject, in order to compare it against other relevant literals (we give a fractional score when a conjunction of literals is required to dismiss a hypothesis). Thus, the score of a relevant literal l is $s = minimum(n, m)$, where n is the number of suggested revisions that l rejects when true, and m is the number of suggested revisions that l rejects when false. The literals with maximum score are the most relevant literal. Literals with equal score could be further ranked according to the maximum number of hypotheses each one falsifies.

3. CASE STUDY

We have applied our approach to the file sharing agents example [3], where only VIP agents have permission to download a block of data without having previously shared its own block. We have used the same use-case for which the learned has produced many alternative changes. We have considered the following four of these changes, which were similar to each other:

1. $occurred(myDownload(X, B), I) \leftarrow$
   $occurred(download(X, Y, B), I), holdsat(hasblock(Y, B), I).$
2. $occurred(myDownload(X, B), I) \leftarrow$
   $occurred(download(X, Y, B), I), holdsat(hasblock(Y, B), I).$
   $occurred(myDownload(X, B), I) \leftarrow$
   $occurred(viol(myDownload(Y, B2)), I),$
   $holdsat(hasblock(Y, B), I).$
3. $occurred(myDownload(X, B), I) \leftarrow$
   $occurred(download(Y, Y, B), I).$
4. $occurred(myDownload(X, B), I) \leftarrow$
   $occurred(download(X, Y, B), I), holdsat(hasblock(Y, B), I).$
   $occurred(myDownload(X, B), I) \leftarrow$
   $occurred(viol(myDownload(X, B2)), I),$
   $holdsat(hasblock(Y, B), I).$

Using our method and forming an abductive problem with background knowledge given by the above revisions and the original specification, we were able to identify the relevant literals and score them according to how many suggestions they could discard. Table 1 contains literals that could potentially dismiss suggestions (2) or (4). Using these values, we were able to identify the following most relevant literals:

Relevant literal	Truth value	
	True	False
$occurred(viol(myDownload(alice, x1)), i06)$	0.0	2.0
$occurred(viol(myDownload(alice, x2)), i06)$	0.0	2.0
$occurred(viol(myDownload(alice, x3)), i06)$	0.0	2.0
$occurred(viol(myDownload(alice, x5)), i06)$	0.0	2.0
$occurred(viol(myDownload(bob, x1)), i06)$	0.5	2.0
$occurred(viol(myDownload(bob, x2)), i06)$	0.5	2.0
$occurred(viol(myDownload(bob, x3)), i06)$	0.5	2.0
$occurred(viol(myDownload(bob, x4)), i06)$	0.5	2.0
$occurred(viol(myDownload(bob, x5)), i06)$	0.5	2.0
$occurred(misuse(bob), i06)$	0.0	0.5

Table 1: Scoring of relevant literals

$occurred(viol(myDownload(bob, x1)), i06)$
$occurred(viol(myDownload(bob, x2)), i06)$
$occurred(viol(myDownload(bob, x3)), i06)$
$occurred(viol(myDownload(bob, x4)), i06)$
$occurred(viol(myDownload(bob, x5)), i06)$

Any of the above literals would dismiss suggestions (2) and (4) when false, or only one of them when true, provided that $occurred(misuse(bob), i06)$ was false.

4. DISCUSSION

In this paper, we have tackled the problem of distinguishing between revisions over normative specifications through the use of test generation. By identifying comparable consequences of the suggested revisions, we are able to use them as a rationale for rejecting possible changes. Our case study provides an example of what outputs can be acquired by the proposed approach. However, as well as showing the outputs that could be used for selection criteria, it also demonstrates that the approach is unable to discriminate between all suggested revisions (no relevant literals that could dismiss the suggestion (1) or (3) were found).

For the future, the problem of the inability to distinguish some hypotheses (possibly by extending the use-case's trace) needs be adressed. An implementation of an automated system for normative revision for further evaluation of the technique is required. We also plan a modification to the scoring method that was used to take into account additional factors such as the length of the revision suggestion.

5. REFERENCES

[1] Alexander Artikis, Marek Sergot, and Jeremy Pitt. An executable specification of an argumentation protocol. In *Proceedings of Conference on Artificial Intelligence and Law (ICAIL)*, pages 1–11. ACM Press, 2003.

[2] Owen Cliffe, Marina De Vos, and Julian Padget. Answer set programming for representing and reasoning about virtual institutions. In Katsumi Inoue, Ken Satoh, and Francesca Toni, editors, *Computational Logic in Multi-Agent Systems*, volume 4371 of *Lecture Notes in Computer Science*, pages 60–79. Springer Berlin / Heidelberg, 2007.

[3] Domenico Corapi, Alessandra Russo, Marina De Vos, Julian A. Padget, and Ken Satoh. Normative design using inductive learning. *TPLP*, 11(4-5):783–799, 2011.

[4] D. Grossi, H. M. Aldewereld, and F. Dignum. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, M.V. Dignum, N. Fornara, and E. Matson, editors, *COIN II*, pages 101–114. Springer, 2007.

[5] Sheila Mcilraith. Generating tests using abduction. In *Proc. Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 449–460, 1994.

A Context-aware Normative Structure in MAS

(Extended Abstract)

Jie Jiang, Huib Aldewereld, Virginia Dignum and Yao-Hua Tan
TPM, Delft University of Technology
Delft, The Netherlands
{jie.jiang, h.m.aldewereld, m.v.dignum, y.tan}@tudelft.nl

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Languages and structures; D.2.2 [**Design Tools and Techniques**]: Petri nets; D.2.4 [**Software/Program Verification**]: Validation

General Terms

Design, Theory, Verification

Keywords

Norm, Multi-agent systems, Context, and Colored Petri Nets

1. INTRODUCTION

Many domains are characterized by agents that interact with each other in accordance with common rules or norms. In international trade, a trading network may include a variety of entities (e.g., software, organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, culture, social capital, and goals. In this context, agents represent real interests and real entities, i.e., different agents have different owners, goals, interests, and preconditions for collaboration. For example, importers are motivated by profit and quality of products, while customs authorities are motivated by safety and security concerns. At any given moment, most agents will be conditioned by different regulations and norms, originating from different institutional contexts.

In this paper, we propose an approach to represent and analyze sets of norms that takes into consideration both the interrelationships between different norms and the context of their application. This extends current approaches where dependence between norms is not explicitly considered. The representation of the influence of institutional contexts on norms facilitates a contextual refinement normative structure, which supports checking inconsistencies between norms. Our approach is different from those based on deontic reasoning, as we do not aim at identifying the deontic consequences of actions. In short, our framework will enable, given a set of norms represented as a graph or net, to check whether there is a possible way to comply with those norms, i.e., a path through the graph which indicates norm compliant at all steps.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. NORMATIVE STRUCTURE

In the analysis of institutional statements, E. Ostrom [1] introduces the ADICO syntax which describes who (Attribute) is obliged/forbidden/permitted (Deontic) to do or achieve what (aIm), when and where (Condition), otherwise (Or else) leading to consequences of violation. In this paper, to model the possible relationships between norms in agent societies, we introduce three logical operators AND, OR, and OE (representing Or else) and define the norms as a composite entity which not only describes the components in ADICO syntax but also represents the relations among different dos and don'ts in a specific institutional context.

Definition 1 (Norm Net). *A Norm Net NN = (context, NS), where context describes the institution within which a set of related norms NS exist.*

Each norm net is associated with an institutional context which describes the environment of the institution where the norm net exists. Making the context explicit enables us to control the evolution of the norm net and to accommodate compliance and resolution of conflicts. A norm set NS is a nested structure composed of a set of hierarchically connected norms in a certain context. In a norm net, obligations and prohibitions may have corresponding sanctions while permissions usually do not. The norms and their sanctions are exclusive and conditional, i.e., either conform to the norms or accept the sanctions when violating the norms, which is in accordance with the semantic of OE operator.

For example, in the EU international trade regulations concerning the issue of *origin of goods*, a norm net can be constructed as $NN_1 = (context_1, NS_1)$ where

- $context_1$ = "non-preferential origin in the EU",
- $NS_1 = OE(\ AND(AND(AND(n_{a1}, n_{a2}), AND(n_{a3}, n_{a4})), OR(n_{b11}, n_{b12})), n_{b2})$, where
  - n_{a1}: The certificate of origin shall measure 210×297 mm.
  - n_{a2}: A tolerance of up to minus 5 mm or plus 8 mm in the length shall be allowed.
  - n_{a3}: The paper used shall be white, free of mechanical pulp, dressed for writing purposes and weigh at least 64 g/m^2 or between 25 and 30 g/m^2 where air-mail paper is used.

– n_{a4}: The certificate of origin shall have a printed guilloche pattern background in sepia such as to reveal any falsification by mechanical or chemical means.

– n_{b11}: The certificate of origin shall be printed in one or more of the official languages of the Community,

– n_{b12}: depending on the practice and requirements of trade, in any other language.

– n_{b2}: The certificate of origin shall not be approved when it is not in the prescribed format.

Figure 1 gives the graphical illustration of the norm net NN_1 represented as an oval. NS_1, represented as a rectangle, is an OE connection of two norm sets NS_2 and NS_3, NS_3 being the consequence of violating NS_2. Specifically, we use a dashed line to indicate the consequence NS_3. NS_2 is composed of two sub norm sets NS_4 and NS_5 connected by AND. Following the same rules, we finally come to the rightmost norms that construct the norm net.

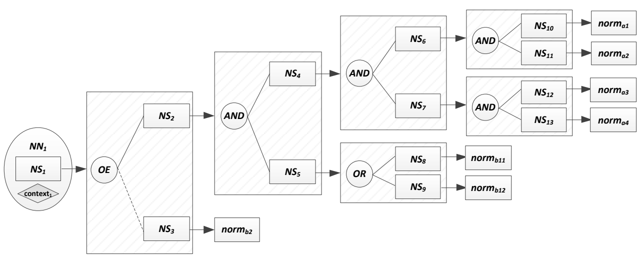


Figure 1: Graphical expression of NN_1.

3. CONTEXTUALIZATION

Laws and regulations are a system of textual rules and guidelines that are enforced through social institutions to govern behavior. They are specified as a normative structure, which describes the expectations and boundaries for agent behavior. We have already presented the representation of norms using *norm net* in Definition 1 to capture the declarative meaning of the law/regulation and also the relations between them. However, in real world domains, norms are not specified at a single level of abstraction. An abstract norm net, resulting from the formalization of law/regulation, may have different extensions according to different contexts. Usually, laws are first issued at a higher abstraction level stating the dos, don'ts and sanctions to regulate actors' behavior. Based on this set of abstract norms, elaboration will be conducted according to the specific characteristics and requirements of different situations, which results into sets of contextual norms. This elaboration process facilitates detailed explanation of abstract norms in a concrete implementing environment.

Figure 2 depicts the process of modeling norms from abstract statements to concrete operation. It starts from an abstract norm net which describes the expectations and boundaries for agent behavior in general. At this level, specification of the norms of the system is abstract and assumed to be stable throughout the life cycle of systems. Meanwhile the actual implementation of the MAS should be flexible and adapt to changing environments and contexts. Therefore, according to different contexts, the abstract norm net is transformed into sets of contextual norm nets which give more specific information on the roles, actions, conditions and the relations between the elaborated norms.

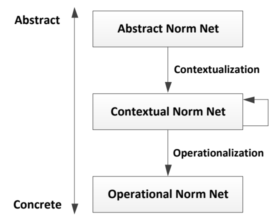


Figure 2: Contextualization and operationalization

Moreover, a contextual norm net can again be further contextualized in a recursive manner, which enables a flexible normative structure and makes it possible for designers at different levels to decide their norm elaborations. Finally, based on the contextual norm nets which contain enough information for the actors to reason about their dos and don'ts in a specific situation, the norms will be extended with operational aspects to capture the operational meaning of the norms.

4. CONCLUSIONS

In this paper, we proposed a normative structure that not only captures the characteristics of a single norm but also the relationships between norms. Given that agents in MAS interact with each other to achieve certain goals, the interrelated effects of norms on their behavior are very important for both individuals and the system. Therefore, the connections between norms should be explicitly indicated in a structural way. Moreover, contexts play an important role in the construction of norms, in the sense that the application of a norm heavily depends on its institutional context and a norm may have different interpretations in different situations. To this end, the concept of norm net in this paper expresses how a set of recursive norm sets organize in a hierarchy of contexts.

Most importantly, this paper presents a norm net contextualization process that describes norms from general to specific. This enables a modular approach for building normative structure and also distributes its complexity. Furthermore, following this contextualization process, actors can have a better understand of their dos and don'ts with the evolution of contextual norm nets. To verify the proposal, we map norm nets to Colored Petri Nets (CPNs) and incorporate agents/actors as colored tokens in the analysis, which presents the state transition process of norm nets and provides a potential approach for compliance checking on norms.

In future work, the normative structure will be extended to the operational level and a complete mapping for contextual norm nets will be built using advanced CPNs.

5. REFERENCES

[1] E. Ostrom. *Understanding institutional diversity.* Princeton University Press, 2005.

A Programming Approach to Monitoring Communication in an Organisational Environment

(Extended Abstract)

Mehdi Dastani
Utrecht University
The Netherlands
M.M.Dastani@uu.nl

Leendert van der Torre
University of Luxembourg
Luxembourg
leon.vandertorre@uni.lu

Neil Yorke-Smith
American University of Beirut
and SRI International, USA
nysmith@aub.edu.lb

ABSTRACT

Agreement technologies [1] achieve coordination among autonomous computational entities, by combining technologies for norms, semantics, organisations, argumentation, negotiation, and trust. We consider how an organisational programming language, such as 2OPL [2], can be extended to monitor communication. Such an extended programming language can be used to facilitate the development of electronic institutions, organisations, or marketplaces that aim at monitoring agent interaction (including both communication and non-communication actions), checking compliance with norms, and enforcing norms by means of sanctions. This abstract reports on specifying an operational semantics for agent interactions within such a setting, distinguishing constitutive norms for monitoring and sanction rules for enforcement of norms.

Categories and Subject Descriptors

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

General Terms

Theory

Keywords

Organisation, Environment, Communication, Commitments, Norms

1. INTRODUCTION

It has been established how to define an organisation where agent communication actions create and operate on social commitments (e.g., [5, 3]). Fornara *et al.* [3, 4] propose an Agent Communication Language (ACL) based on communication actions and define norms as "rules that manipulate commitments of the agents engaged in an interaction". Dastani *et al.* [2] specify norms to govern agent interaction; they provide rules to specify norms and sanctions, but do not focus on communication actions.

This abstract reports on extending the approach of [2] with communication actions by following the successful line of work using commitments. The extension is based on the assumption that communication should respect a set of generic norms which are inherent in communication actions.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1.1 Research questions

The following questions guide the design of a programming language (and its operational semantics) that facilitates the implementation of norm-based organisations:

1. How to define an operational basis for both communication and non-communication actions in norm-based organisations?
2. How to uniformly handle monitoring and enforcing of norms for both types of actions?
3. How to develop a full operational semantics with desirable properties—without focusing on the protocol or semantics concerns of a full ACL—by adopting a simple (but extendible) set of communication actions?

Compared to the foundational work of Singh and Colombetti and colleagues, our approach differs in that: (1) We aim to use *counts-as rules* explicitly as technical constructs while Fornara *et al.* treat counts-as relation primarily as linguistic conventions. (2) We want to provide an operational semantics for interactions (among which communication actions) within an organisational setting and analyze the properties of interaction. Fornara *et al.* provide semi-formal specification of organisations and consider only communication actions. (3) We aim to consider the effect of non-communicative actions as further the elapse of deadlines. (4) We want to allow for sanction rules whereas earlier works leave open the question of what should happens when norms or commitments are violated. We adopt a standard, contemporary lifecycle of commitments, following [6].

2. OVERVIEW OF THE APPROACH

An organisation monitors the agents' interactions (both communication and non-communication actions), determines the (state of) commitments and the violated norms, and ensures that the agents fulfil their commitments and norms—or otherwise imposes sanctions such as putting the violating agent on a blacklist. In our approach, the organisation, as an exogenous process, cannot *intervene* in the decision making of individual agents; in this setting agents are assumed to be autonomous in the sense that they decide their own actions. In the following subsections, we indicate how such organisations can be programmed.

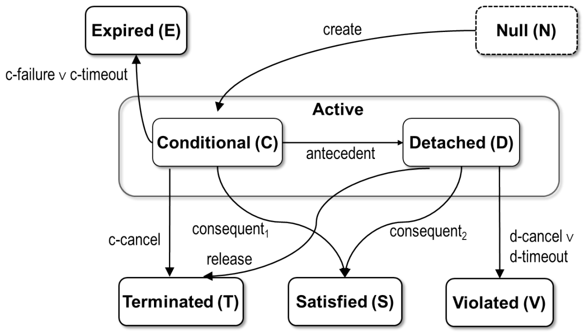


Figure 1: State transitions of commitment lifecycle.

2.1 Commitments

Social commitments represent a popular means of capturing relations between two agents with deontic force; they provide the basis for an ACL within an organisational setting [5, 3]. Formally, a *commitment* [6] is defined as a tuple $\mathsf{C}(x, y, p, q, d_1, d_2)$, which can be read as "Agent x (as debtor) commits to agent y (as creditor) that if proposition p (the antecedent) is brought about by deadline d_1 then x will bring about q (the consequent) by deadline d_2".

Fig. 1 shows the states of a simplified lifecycle of a commitment, adapted from [6] (we omit suspension and delegation). Boxes indicate states and arrows transitions. We write commitment state with superscript, i.e., $\mathsf{C}^{\text{state}}$.

2.2 Agent interactions

Possible actions that agents perform to interact with each other or with their shared environment include pure communication actions (e.g., promise to pay), and non-communicative actions that change the actual state of their environment (e.g., make a payment). Our purpose is not to define an ACL or a communication protocol; instead, we select a representative set of actions influencing the generation and state of commitments. We take the following set to demonstrate our methodology for programming an organisational model and the management and enforcement of commitment-based norms. We use variables $x,\ y, \ldots$ to range over the agent names $i,\ j, \ldots$; propositional variables $p,\ q, \ldots$ to range over propositions $a,\ b, \ldots$; and finally $d,\ d', \ldots$ to range over deadlines $t_m,\ t_n, \ldots$, where $m, n \in \mathbb{N}$.

- $\mathit{offer}(x, y, p, q, d_1, d_2)$ — x tells y that x will make q true in the environment by deadline d_2 if p becomes true in the environment by deadline d_1
- $\mathit{tell}(x, y, p)$ — x tells y that p is true in the environment
- $\mathit{cancel}(x, y, q)$ — x tells y that x will not make q true
- $\mathit{release}(y, x, q)$ — y tells x that x needs not make q true
- $\mathit{failure}(x, y, p)$ — x tells y that p cannot be made true in the environment
- $\mathit{do}(x, p)$ — x performs an action to make proposition p true in the environment

We assume here that agents are trusted, i.e., their utterances are according to their beliefs. Note that an organisation may develop a list of trusted agents.

2.3 Organisation

An *organisation* is specified by facts, norms, and sanctions. Norms are states that an organisation aims at enforcing and can therefore be seen as the goals of the organisation. We distinguish *brute* and *institutional* facts. Brute facts denote the state of the shared environment (e.g., b_j denoting the fact that agent j has book b or $p_{(b,20)}$ denoting the fact that 20 euro is paid for book b), while institutional facts denote the normative state of an organisation (e.g., $\mathsf{C}^D(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5)$ denoting the fact that agent i is committed to pay 20 euro before t_5 if agent j sends book b before t_2, or $viol_{reg-b}$ denoting the fact that agent b has violated the registration norm).

We follow [2] and represent norms by means of the *counts-as* construct. The original version of the counts-as construct is of the form "ϕ counts as ψ in the context c". We program the monitoring component of an organisation by constructs of the form $\phi \wedge c \Longrightarrow_{cr} \psi$, where $\phi \wedge c$ can be either brute or institutional facts and ψ is an institutional fact. For example, the counts-as rule $\mathit{offer}(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5) \Longrightarrow_{cr} \mathsf{C}^C(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5)$ implements a norm that an offer by agent i to agent j to do a payment if j sends i a book *counts-as* a conditional commitment. Finally, we program sanctions by rules of the form $\phi \Longrightarrow_{sr} \psi$, where ϕ can comprise brute and institutional facts and ψ is a brute fact. In our running example, $\mathsf{C}^C(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5) \Longrightarrow_{sr} \mathit{blacklist}(i)$ implements the sanctions to add agent i to a blacklist if agent i payment is not fulfilled after day 5. Following the above, an *organisation* is programmed by the tuple (F, cr, sr), where F is a set of initial brute facts, cr is a set of counts-as rules, and sr is a set of sanction rules. The institutional facts are generated during run-time.

2.4 Discussion

The details of the operational semantics consist of transition rules specifying how agent actions (communication and non-communication) create and modify institutional facts, including commitments, how the organisation determines norm violations by applying counts-as rules, and how the organisation respond to norm violations by applying sanction rules. Our ongoing work is to explore the properties of normative multi-agent system executions and to apply the programming approach to realistic scenarios.

3. REFERENCES

[1] Billhardt, H. et al. Organisational structures in next-generation distributed systems. *Multiagent and Grid Systems*, 7(2–3):109–125, 2011.

[2] M. Dastani, D. Grossi, J.-J. C. Meyer, and N. Tinnemeier. Normative multi-agent programs and their logics. In *Proc. Workshop on Knowledge Representation for Agents and Multi-Agent Systems*, LNCS 5605, pages 16–31, 2009.

[3] N. Fornara and M. Colombetti. A commitment-based approach to agent communication. *Applied Artificial Intelligence*, 18(9–10):853–866, 2004.

[4] N. Fornara, F. Viganò, and M. Colombetti. Agent communication and artificial institutions. *Autonomous Agents and Multi-agent Systems*, 14:121–142, 2007.

[5] A. U. Mallya, P. Yolum, and M. P. Singh. Resolving commitments among autonomous agents. In *Proc. Workshop on Agent Comm.*, pages 166–182, 2003.

[6] P. R. Telang and M. P. Singh. Specifying and verifying cross-organizational business models. *IEEE Trans. Services Computing*, PrePrint, 2011.

On modeling punishment in multi-agent systems

(Extended Abstract)

Subhasis Thakur
Griffith University and NICTA
Brisbane, Australia
subhasis.thakur@nicta.com.au

Guido Governatori
NICTA
Brisbane, Australia
Australia
guido.governatori@nicta.com.au

Abdul Sattar
Griffith University and NICTA
Brisbane, Australia
abdul.sattar@nicta.com.au

ABSTRACT

In this paper we study isolation as a form of punishment. Although an isolated violator is punished as it can not benefit from the interactions with other agents, compliant agents may also suffer from not engaging with the violators. In this paper we analyze such problems. Certain modifications of multi agent systems are needed to solve this problem. These modifications are aimed to make the violator redundant so that it can be ignored and hence isolated. Deciding on these modifications is NP-complete and approximation algorithms exist.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Algorithms

General Terms

theory

Keywords

Punishment, Multi-Agent Systems, Norms

1. INTRODUCTION

Compliance with norms makes an agent's behavior more predictable, which enhances coordination among agents [9, 10, 8]. Besides improving coordination, compliance with norms also satisfies some system level conditions set by the designer of the norms. Although norms are beneficial for the agents, violations of norms occur because certain self interested agents may prefer to achieve some individual goals over compliance with the norms. To encourage compliant behaviors various norm enforcement techniques have been developed. Norm enforcement can be classified [11] into two categories: (a) self enforcement and (b) third party enforcement. In self enforcement the agents themselves execute the punishment and in case of third party enforcement, an external agent decides the amount of punishment and executes the necessary procedures for it. Downgrading the violators reputation is a common practice in various enforcement models [2][5]. As the reputation is downgraded the compliant agents do not interact with the violator as it do not treat the violator as trustworthy. Although there is vast literature dedicated towards norm enforcement models, there is a lack of formal models of punishment. Most of the existing models of punishment suggests that a violator must not be engaged in further joint actions or other multi agent activities. But these models do not consider the effects of this isolation on compliant agents. As compliant agents are not interacting with the violators, they may lose some utility. This can happen because a violator owns certain unique abilities, and it can no longer act as a mediator. In this paper we assume that abilities of a violator are not replaceable but its role as mediator can be replaced. Let us consider the following example that illustrates the adverse effect of isolation: consider the normative multi-agent system *nmas* depicted in Figure 1 as a graph where the nodes $(a_1, \dots, a_7)$ represent compliant agents and v represents the violator. The edges represent communication links between agents. Agents can perform joint actions if they are connected. Due to isolation, the edges connecting the violator with other agents are removed as a form of punishment. One of the adverse effects of this isolation is that $a5$ can not perform any joint actions with other agents. Also, suppose that, agents interact with other agents if they are trustworthy. An agent computes the trust of other agents by recommendations from their neighbours. Trust propagation models typically assume that the trust between two agents is proportional to the number of mediators between them. So, it may happen that, as v is not mediating between $a7$ and $a4$, the trust on $a4$ by $a7$ results lower after the isolation of v, and as consequence $a4$ and $a7$ will not interact. Research

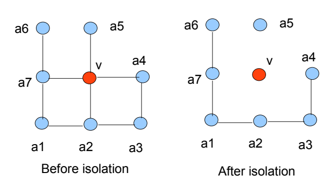


Figure 1: Effect of Isolation

in multi agents has shown that *mediators* play an important role in a variety of multi agent problems, for example trust and joint actions [12], recommendation systems [6], and multi agent negotiation [3, 7]. Thus punishment mechanisms based on isolation have to take into account how to minimise the adverse effects of isolation. In this paper

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

we propose a model of punishment which neutralizes these adverse effects of *isolation*. Our model of *isolation* is as follows: a *nmas* is represented as a graph. Due to isolation of the violators, all edges with the violators are deleted. The effect of isolation on the compliant agents is modeled as follows: **Problem 1** Assume that there is a set of nodes called *sources*. Agents preferences over the sources are described as an order over the sources such that an agent is willing to pay more to connect with a more preferred source than a less preferred one. Agents coordinate their actions to form a minimum cost spanning tree over the *nmas* such that the costs to connect with the sources along the spanning tree is according to their preferences. Assume that, before isolating a violator, there was such a spanning tree. So the isolation problem becomes the creation of new edges at a minimum cost such that it is possible to construct such a spanning tree. Thus addition of the new edges makes the violators redundant as the compliant agents can form an optimal tree without cooperation from the violators and hence they can afford to ignore the violators. Notice that this problem can be used to simulate general coordination problems in multi agent systems. **Problem 2** Assume that, before the isolation for every agent there is a subset of agents who can be reached within a predefined length (say K). Due to the isolation of the violators, some edges are no longer in use. So an agent losses certain agents within its reachable range. So in this case the isolation problem becomes the creation of certain new edges at a minimum cost such that each affected agent can reach the previously lost agents. Thus after the addition of the new edges the role of violators as mediators become redundant. Hence the compliant agents can isolate the violators as they do not need cooperation from them. This problem is motivated by trust propagation models and recommendation system models. In both cases, an agent's reputation or trust ranking usually can be changed if the topology of the system is changed. Isolating the violators triggers such changes and hence it must be neutralized by creating alternate paths between the compliant agents.

2. RESULTS

Theorem 1. *Isolation with preference is NP-complete for 2 sources.*

Theorem 2. *Isolation with preference is NP-complete for more than 2 sources.*

Theorem 3. *A polynomial time* $(\frac{1}{1-m})$*-approximation solution for the problem of isolation with preference exists.*

Theorem 4. *K-dedicated source-minimum cost spanning tree is NP-complete for two sources.*

Theorem 5. *K-dedicated source-minimum cost spanning tree with variable peer limit is NP-complete for 2-sources.*

Theorem 6. *K-dedicated source-minimum cost spanning tree with variable peer limit is NP-complete for 2-sources.*

Theorem 7. *A polynomial time 2-approximation solution to 2- dedicated source minimum cost spanning tree exists.*

As a comparison with relevant works in *nmas*, [1] gives a notion of enforceable social laws in terms of the efficiency of the system to remain compliant. In [4] a similar concept of punishment is introduced. By the adverse effect of punishment on the compliant agents is never analyzed in *mas* literature. In that regard this paper is the first towards formalizing the cost of punishment. A theory of cost of punishment helps to build enforcement laws in *mas* in such a way that the cost to punish the agents becomes affordable.

3. ACKNOWLEDGMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program

4. REFERENCES

[1] G. Boella and L. van der Torre. Enforceable social laws. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 682–689, New York, NY, USA, 2005. ACM.

[2] C. Castelfranchi, R. Conte, and M. Paolucci. Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation*, 1, 1998.

[3] M. Chalamish and S. Kraus. Automed: an automated mediator for bilateral negotiations under time constraints. In *AAMAS*, page 248, 2007.

[4] A. P. de Pinninck, C. Sierra, and W. M. Schorlemmer. Friends no more: norm enforcement in multiagent systems. In *AAMAS*, page 92, 2007.

[5] D. Hales. Group reputation supports beneficent norms. *Journal of Artificial Societies and Social Simulation*, 5:4, 2002.

[6] A. Perreau De Pinninck, C. Sierra, and M. Schorlemmer. A multiagent network for peer norm enforcement. *Autonomous Agents and Multi-Agent Systems*, 21:397–424, November 2010.

[7] T. Rauenbusch. The mediation algorithm for real time negotiation. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*, AAMAS '02, pages 1139–1140, New York, NY, USA, 2002. ACM.

[8] Y. Shoham and M. Tennenholtz. Emergent Conventions in Multi-Agent Systems: Initial Experimental Results and Observations (Preliminary Report). In *Proceedings of the 3rd International Conference on KR&R*, pages 225–232, 1992.

[9] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *AAAI*, pages 276–281, 1992.

[10] A. Walker and M. Wooldridge. Understanding the emergence of conventions in multi-agent systems. In *ICMAS*, pages 384–389, 1995.

[11] B. V. Yarbrough and R. M. Yarbrough. Governance structures, insider status, and boundary maintenance. *Journal of Bioeconomics*, 1:289–310, 1999. 10.1023/A:1010058113016.

[12] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7:337–358, December 2005.

Strategic Pseudonym Change in Agent-Based E-Commerce

(Extended Abstract)

J. M. Such
Dep. de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, València, Spain
jsuch@dsic.upv.es

E. Serrano
Departamento de Ingeniería de la Información y las Comunicaciones
University of Murcia, Spain
emilioserra@um.es

V. Botti, A. García-Fornes
Dep. de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, València, Spain
{vbotti,agarcia}@dsic.upv.es

ABSTRACT

In agent-based e-commerce applications, vendors can construct detailed profiles about customers' preferences. These profiles can then be used to perform practices such as price discrimination, poor judgment, etc. The use of pseudonyms and, specially, changing pseudonyms from time to time are known to minimize profiling. Although there are some agent frameworks and platforms that support pseudonym change, there are few proposals that suggest or directly change the pseudonym in an automatic fashion. Instead, users are usually provided with the mechanisms to change pseudonyms but without any mechanism that aids them to decide when to change their pseudonyms. We present in this paper an approach to pseudonym change based on human privacy attitudes.

Categories and Subject Descriptors

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

General Terms

Theory, Experimentation

Keywords

Privacy

1. INTRODUCTION

The explosive growth of the Internet in the last decades has caused that as of 2011 more than 2 billion users are connected to it[1]. In this environment, privacy is of great concern. Users are constantly exposed to personal information collection and processing without even being aware of it [2].

[1]`http://www.internetworldstats.com/stats.htm`

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

In this paper, we focus on a type of information processing called buyer profiling [5], in which vendors obtain detailed profiles of their customers and tailor their offers regarding customer's tastes. These profiles can represent a serious threat to privacy. For instance, these profiles can be used to perform *price discrimination* [6]. This is when vendors charge customers different prices for the same good according to the customers' profiles, i.e., if a vendor knows that some good is of great interest to one customer, the vendor could charge this customer more money for this good than other customers for the same good.

Hansen et al. [4] encourage the use of pseudonyms to prevent buyer profiling. Specifically, they claim that pseudonyms should be changed from time to time to avoid profiling. Indeed, the most privacy-preserving option is to use transaction pseudonyms, i.e., to use a different pseudonym for each different transaction.

Pseudonym-based techniques have been integrated in agent technologies. Such et al. [8] present a pseudonym management model that has been implemented into a an agent framework [7]. Warnier and Brazier [10] also present a proposal for supporting pseudonym management in an agent framework. Both proposals include the necessary mechanisms for agents to be able to hold and change their pseudonyms but nothing is said about when a pseudonym should be changed or not. Moreover, the proposal of Warnier and Brazier [10] allows the automatic change of pseudonyms for each message sent. However, they do not consider the fact that there are many cases in which the user can be interested in reusing the same pseudonym even though this could cause a potential privacy loss, e.g., when some benefit is expected if they reuse the same pseudonym, such as price discounts, the building of a reputation, etc.

We present in this paper an approach to pseudonym change based on these general human attitudes towards privacy. In this way, agents obtain an estimation of the privacy loss and the utility of reusing a pseudonym. Thus, agents can automatically decide whether or not to change a pseudonym without the need of human intervention, but complying with its user's attitude towards privacy.

2. STRATEGIC PSEUDONYM CHANGE

Some studies have concluded that Humans have different general attitudes towards privacy [9, 11]. Privacy fundamentalists are extremely concerned about privacy and reluctant to lose privacy, they feel that they have already lost too much privacy and are reluctant to lose privacy any more. Privacy pragmatists are concerned about privacy (i.e. they are not willing to lose privacy a priory), but if they expect some utility (e.g. a monetary benefit) they may accept a privacy loss in exchange of this utility. Finally, privacy unconcerned do not consider privacy loss at all. A survey made in 2003 among 1.010 US adult citizens [9] shows that 26% of that citizens are considered privacy fundamentalists, 64% privacy pragmatists, and 10% privacy unconcerned.

To model these attitudes when it comes to pseudonym change, we consider that the decision of whether or not to change a pseudonym is based on a tradeoff between the privacy that will be lost if the pseudonym is not changed and the utility that will be earned if the pseudonym is not changed. For instance, in the case of privacy pragmatists, the agent can decide to not change its pseudonym in the next transaction if the privacy that will be lost is worth the utility that will be gained. We model this problem as a multi-objective optimization problem [1], in which an agent tries to minimize privacy loss while maximizing its utilitarian benefit.

One of the most used approaches to solve multi-objective optimization problems consists of transforming it into a single-objective problem[2] [3]. This is typically done by assigning a numerical weight to each objective (evaluation criterion) and then combining the values of the weighted criteria into a single value by adding all the weighted criteria.

In our case, agents consider two criteria: privacy loss and utility. Considering these two criteria, agents have two options: either to change or not to change its pseudonym in their next transaction. Thus, we are interested in measuring the quality in terms of the privacy loss and the utility of each of these options. An agent will choose the option with the highest quality. We formally define the option set as $\Theta = \{change, nochange\}$. Moreover, we define the quality of an option as:

Definition 1 (Option Quality). *Given a criterion function $c_p(\cdot)$ that evaluates privacy loss, a criterion function $c_u(\cdot)$ that evaluates utility, and weights $w_p, w_u \in [0,1]$ so that $w_p + w_u = 1$, the quality Q_δ of an option $\delta \in \Theta$ is:*

$$Q_\delta = w_p \cdot c_p(\delta) + w_u \cdot c_u(\delta) \tag{1}$$

The specific criterion functions $c_p(\cdot)$ and $c_u(\cdot)$ are domain-dependent. Moreover, as privacy loss units may be different from utility units, both criterion functions are expected to return a value in the interval $[0,1]$ so that they can be comparable. Depending on the final domain, this could require a normalization process. This also implies that the quality of an option $\delta \in \Theta$ will be in that same interval, i.e., $Q_\delta \in [0,1]$.

With the option quality formula, agents are able to obtain the quality of each of the options. Thus, they are able to choose whether or not to change their pseudonym in the next transaction. Agents will choose the option with the maximum quality. Formally, an agent will choose an option $\delta^* \in \Theta$ so that:

$$\delta^* = \arg\max_{\delta \in \Theta} Q_\delta \tag{2}$$

We model privacy attitudes by appropriately setting the values for the weights in the option quality formula (Equation 1), i.e., by setting w_p and w_u. If $w_p = 1$ (so $w_u = 0$) we are modeling privacy fundamentalists because they will only try to minimize privacy loss. Thus, they will not consider utility at all. If $w_p = 0$ (so $w_u = 1$) we are modeling privacy unconcerned because they will not consider privacy loss but the maximization of their utility. Finally, if $w_p \neq 1 \wedge w_p \neq 0$ we are modeling privacy pragmatists. Moreover, the specific value for w_p and w_u will vary according to how much a user valuates privacy in front of utility.

3. CONCLUSIONS

In this paper, agents decide whether to change a pseudonym or not based on the specific attitude towards privacy of their users. This specific attitude is what determines to what extent an agent valuates the privacy loss and the utility of changing/not changing a pseudonym.

4. ACKNOWLEDGMENTS

This work has been partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, and project TIN2008-04446.

5. REFERENCES

[1] K. Deb. Multi-objective optimization. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 273–316. Springer US, 2005.

[2] S. Fischer-Hübner and H. Hedbom. Benefits of privacy-enhancing identity management. *Asia-Pacific Business Review*, 10(4):36–52, 2008.

[3] A. Freitas. A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations Newsletter*, 6(2):77–86, 2004.

[4] M. Hansen, A. Schwartz, and A. Cooper. Privacy and identity management. *IEEE Security & Privacy*, 6(2):38–45, 2008.

[5] M. Hildebrandt and S. Gutwirth. *Profiling the European Citizen: Cross-Disciplinary Perspectives.* Springer Publishing Company, Inc., 2008.

[6] A. Odlyzko. Privacy, economics, and price discrimination on the internet. In *Proc. of ICEC*, pages 355–366, 2003.

[7] J. M. Such, A. Espinosa, and A. Garcia-Fornes. An agent infrastructure for privacy-enhancing agent-based e-commerce applications. In *Proc. of ITMAS*, pages 16–30, 2011.

[8] J. M. Such, A. Espinosa, A. Garcia-Fornes, and V. Botti. Partial identities as a foundation for trust and reputation. *Engineering Applications of Artificial Intelligence*, 24(7):1128–1136, 2011.

[9] H. Taylor. *Most People Are "Privacy Pragmatists" Who, While Concerned about Privacy, Will Sometimes Trade It Off for Other Benefits.* Harris Interactive, 2003.

[10] M. Warnier and F. Brazier. Anonymity services for multi-agent systems. *Web Intelligence and Agent Systems*, 8(2):219–232, 2010.

[11] A. Westin. *Privacy and Freedom.* New York Atheneum, 1967.

[2] For the sake of clarity and simplicity we only consider the transformation of multi-objective optimization problems into single-objective problems. However, there are other approaches to solve these kind of problems in the existing literature on multi-objective optimization (refer to [1] and [3]).

Multi-dimensional Transition Deliberation for Organization Adaptation in Multiagent Systems

(Extended Abstract)

Juan M. Alberola
Univ. Politècnica de València
Camí de Vera s/n. 46022
València. Spain
jalberola@dsic.upv.es

Vicente Julian
Univ. Politècnica de València
Camí de Vera s/n. 46022
València. Spain
vinglada@dsic.upv.es

Ana Garcia-Fornes
Univ. Politècnica de València
Camí de Vera s/n. 46022
València. Spain
agarcia@dsic.upv.es

ABSTRACT

In this paper, we provide an approach for organization adaptation in Multiagent Systems that considers transitions in multiple dimensions and it is aimed at obtaining the adaptation with the highest potential for improvement in utility based on the costs of adaptation.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Design

Keywords

Organizations,Adaptation,Transitions

1. INTRODUCTION

Organization adaptation eliminates the need to determine all possible runtime conditions a priori, which is unknown in many systems. Before this can occur, the space of organizational options must be mapped and their relative benefits and costs understood [3]. To date, however, few models have emerged that incorporate mechanisms for adaptation that focus on changes in different dimensions of the organization according to the heterogeneous impact that these changes causes in the components of the organization [2]. One main reason is that current approaches do not provide support for specifying the requirements of organizations that are to be achieved. The other reason is that without this support, it is difficult to measure without carrying out the adaptation, the impact on the costs of applying the adaptation and on the performance of the whole organization.

In this paper, we propose a novel approach for organization adaptation called Multi-dimensional Transition Deliberation Mechanism (MTDM). This mechanism provides a decision-making support that considers transitions in different dimensions such as role reallocation, agent population and the structural topology, which increases the range of adaptation solutions. By specifying the requirements of the final organization that is to be achieved, the MTDM accurately predicts the impact of the transition in terms of two aspects: the costs associated to the organization transition, and the benefits or costs that this transition causes not only to the agents involved in the change but also to the whole organization.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ORGANIZATION TRANSITION MODEL

The Organization Transition Model presented in [1] defines the state of organization at two different moments and determines how to carry out a transition from organization to another. An *organization* at a specific moment t is composed by a set of roles R^t, services S^t, and agents A^t. Furthermore, organizational relationships represent links between these elements, where $offers^t$ represents the relationships between roles and services; $provides^t$ represents the relationships between agents and services; $plays^t$ represents the relationships between agents and roles; and $acquaintance^t$ represents the relationships between a pair of agents.

An *event* (ε) defines each individual change that can be applied during the organization transition in terms of addition or deletion. Given two organizations, O^c and O^f, we define $\tau = \{\varepsilon_1 \dots \varepsilon_n\}$ as the *set of events* that cause a transition to O^f when all of them are applied to O^c.

3. MULTI-TRANSITION DELIBERATION MECHANISM

The MTDM is a multi-stage mechanism that is based on a model proposed by Zott [5] in the strategic management research area for analyzing the performance of business firms. This mechanism calculates transitions in different dimensions to other organizations with high expected utility based on the cost for transition to these organizations. The benefits and costs of transition are measured in terms of Organization Transition Impacts (OTIs). Then, the MTDM decides which transition is finally implemented and provides the sequence of changes required to carry out the transition.

3.1 Calculating the Organization Transitions

The first stage calculates the organization with the highest potential for improvement in utility based on the transition cost for several transitions in different dimensions: changing the roles played by agents, the structural topology, and the agent population.

Each event ε has an associated impact $i(\varepsilon)$ that represents the costs/benefits that the application of this event causes in the organization. This impact shows the effect of this event in the components involved in the change and also how other components are affected by this event. Moreover, the impact shows the cost for carrying out the application of the event. Therefore, for any set of events τ that allow a transition from a current organization O^c to a future organization O^f, we define the OTI that is associated to this transition as the impact of applying all the events of τ: $I(\tau) = \sum_{\varepsilon \in \tau} i(\varepsilon)$.

3.1.1 Role Reallocation Transition

A role reallocation transition entails the application of a specific set of events τ_R , which transforms the $provider^c$ and $plays^c$ relationships into $provider^f$ and $plays^f$, respectively.

Let Θ_R denote the set of all the possible sets of events τ_R that define a different role reallocation transition from O^c to O^f. The challenge of the role reallocation transition is to find the specific set of events $\widehat{\tau}_R$ that minimizes the role reallocation transition impact:

$$OTI(\widehat{\tau}_R) = \underset{\tau_R \in \Theta_R}{\operatorname{argmin}}\, OTI(\tau_R)$$

The application of the set of events of the minimal impact $\widehat{\tau}_R$ to O^c would cause a transition to a future organization O_R, which can be transitioned to at the minimal OTI.

3.1.2 Acquaintance Transition

An acquaintance transition entails the application of a specific set of events τ_A, which transforms $acquaintance^c$ into $acquaintance^f$.

Let Θ_A denote the set of all the possible sets of events τ_A that define a different acquaintance transition from O^c to O^f. The challenge of the acquaintance transition is to find the specific set of events $\widehat{\tau}_A$ that minimizes the acquaintance transition impact:

$$OTI(\widehat{\tau}_A) = \underset{\tau_A \in \Theta_A}{\operatorname{argmin}}\, OTI(\tau_A)$$

The application of the set of events of the minimal impact $\widehat{\tau}_A$ to O^c would cause a transition to a future organization O_A, which can be transitioned to at the minimal OTI.

3.1.3 Agent Population Transition

An agent population transition entails the application of a set of events τ_P, which causes the modification of $agents^c$, $provides^c$, $plays^c$, and $acquaintances^c$ into $agents^f$, $provides^f$, $plays^f$, and $acquaintances^f$, respectively.

Let Θ_P denote the set of all the possible sets of events τ_P that define a different agent population transition from O^c to O^f. The challenge of the agent population transition is to find the specific set of events $\widehat{\tau}_P$ that minimizes the agent population transition impact:

$$OTI(\widehat{\tau}_P) = \underset{\tau_P \in \Theta_P}{\operatorname{argmin}}\, OTI(\tau_P)$$

The application of the set of events of the minimal impact $\widehat{\tau}_P$ to O^c would cause a transition to a future organization O_P, which can be transitioned to at the minimal OTI.

3.2 Deliberation

Once the organizations that minimizes the OTI for each dimension are calculated, the second stage of the MTDM decides which transition is finally implemented depending on the deliberation strategy. The deliberation strategy used in this implementation is focused on selecting the transition or the combination of transitions that minimizes the OTI.

3.3 Calculating the sequence of events

Finally, once the final organization O^f that is transitioned to is selected, this stage obtains the specific sequence of events τ that allow this transition from O^c to O^f and the impact associated to applying these events $OTI(\tau)$.

4. CONCLUSIONS

The contributions of this work can be viewed from different perspectives. The MTDM provides an accurate estimation of the transition impact since the organization that is to be achieved is calculated by each transition. Thus, the impact associated to each change that is required to carry out the transition, can be measured individually and more accurately than other approaches. The suitability of the adaptation must be considered taking into account not only the benefits obtained by adaptation but also the costs associated to this process. Approaches that only focus on criteria to improve the utility, the costs for achieving these transitions may be so high that the mean utility gets worse. This issue is also important in human organizations since most organizational changes may encounter problems when they are applied [4].

Another contribution of the MTDM is the possibility of including several transitions into the deliberation decision mechanism. Approaches that consider one-dimensional transitions (roles, structural topology, population, etc.) offer a more limited range of solutions than the MTDM. Thus, in heterogeneous scenarios in which several changes can affect the performance of the organization, a multi-transition criteria for deliberation would provide better decisions for adaptation.

Acknowledgments

This work has been partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, TIN2009-13839-C03-01, TIN2011-27652-C03-01, and PROMETEO/2008/051. Juan M. Alberola has received a grant from Ministerio de Ciencia e Innovación de España (AP2007-00289).

5. REFERENCES

[1] J. M. Alberola, V. Julian, and A. Garcia-Fornes. A cost-based transition approach for multiagent systems reorganization. In *Proc. 10th Int. Conf. on Aut. Agents and MAS (AAMAS11)*, pages 1221–1222, 2011.

[2] J. M. Alberola, V. Julian, and A. Garcia-Fornes. Open issues in multiagent system reorganization. In *Proc. 9th Int. Conf. on Practical Applications of Agents and MAS*, pages 151–158, 2011.

[3] B. Horling and V. Lesser. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2005.

[4] J. Kotter and L. Schlesinger. Choosing strategies for change. In *Harvard Business Review*, pages 106–114, 1979.

[5] C. Zott. Dynamic capabilities and the emergence of intraindustry differential firm performance: Insights from a simulation study. *Strategic Management Journal*, 24(2):97–125, 2003.

Using a hierarchy of coordinators to overcome the frontier effect in social learning*

(Extended Abstract)

Sherief Abdallah
British University in Dubai
University of Edinburgh
shario@ieee.org

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Performance

Keywords

Supervision, Hierarchy, Reinforcement Learning

1. INTRODUCTION

We propose in this paper the use of a hierarchy of coordinators to improve the convergence of a network of agents to a global norm. A norm or a convention is an unwritten law that a society of agents agree on. Social norms are used by humans all the time. Choosing on which side of the road to drive a car and the right-of-way at an intersection are well-known examples. In a multi-agent setting, a convention may refer to a dominant coordination strategy, a common communication language, or the right of way among a group of robots. Upon establishing a norm, the overhead of coordination drops and the reliability of the multi-agent system increases [2]. When studying the emergence of norms and conventions, we typically assume the interaction between agents is random: a pair of agents are selected randomly to interact with one another. The process repeats both concurrently (several pairs interact at the same time) and consecutively (each agent collects history of interactions). When agents are adaptive, the process is then referred to as social learning. The coordination game is perhaps the most widely used game for studying social learning as it presents an agent community with two equally plausible norms to choose from (i.e. two Nash equilibriums). It was shown that in the absence of any restriction on agent interactions, a norm is guaranteed to emerge in the simple social learning setting where agents play the coordination game [1].

*The work reported in this paper was supported by Emirates Foundation grant #2010/107 , Program: Science and Engineering Research Grants

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Recently social leaning was studied in networks [3]. The main difference here is that an underlying network restricts the interactions between the agents. In such a setting, convergence to a global norm is no longer guaranteed as more than one (sub)convention might emerge concurrently and remain stable. A sub-convention is a convention that is not adopted by the vast majority of the agents. The reason for the emergence of multiple stable sub-conventions is the existence of a stable barrier that separates the sub-conventions from one another (or equivalently, prevents each convention from invading the other). Such a barrier creates a suboptimal equilibrium. The frontier effect was reported to either prevent or significantly slowdown the convergence to a global norm across variety of network types. One of the recent proposed solutions to overcome the frontier effect was the use of social instruments [3]. Although the social instruments were successful in overcoming the barrier in regular and random networks, the social instruments failed in the case of scale-free networks. Furthermore, these social instruments had several limitations.

2. PROPOSED SOLUTION: HIERARCHY OF COORDINATORS

When we looked at examples of the frontier effect in scale-free networks, it became apparent that the problem was in the strictly local view of agents on the frontier. If only agents had a more global view, they would have reached a global convention. The social instruments that were proposed before [3] effectively provided (implicitly) individual agents with slightly more global view. For example, the observation social instrument allowed an agent to observe another agent in the network without being restricted to the underlying network (i.e. an agent could observe what convention was adopted by a randomly chosen agent anywhere in the network). The re-wiring social instrument also allowed agents to extend their view beyond their immediate neighbors. Here we propose a more structured mechanism for agents to exchange information (with varying detail and range) about their current state and exchange advices about the best course of action to reach a convention. We propose the use of an organization (hierarchy) of coordinators, as follows. Agents are separated in-to clusters, where each cluster is assigned a coordinator from the agents in the cluster. The clusters are then grouped in to meta-clusters, again selecting one of the cluster members to be the coordinator. The process is repeated recursively until we end up with root coordinator (the hierarchy can stop at a lower level with a

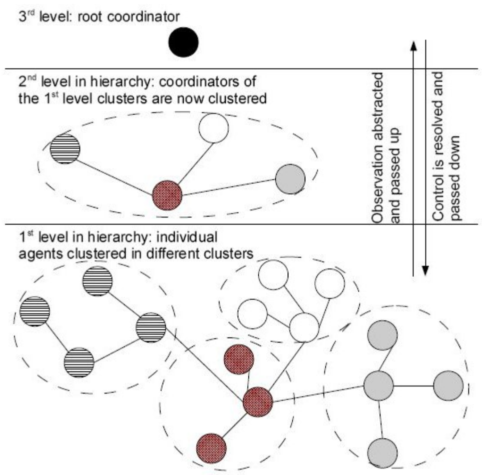


Figure 1: Illustration of the coordinators hierarchy. In the first level we have individual agents who are clustered into four clusters. In the second level we have the coordinators of the clusters in the first level (each coordinator has the same color as its cluster). In the final level we have the root coordinator (note that the hierarchy does not have to end with an individual root).

set of clusters at the top). Figure 1 illustrates the hierarchy of coordinators operating over a scale-free network. In our experiments we use a simple bottom-up hierarchical clustering to automatically build the coordinators' hierarchy, but more sophisticated clustering techniques can be used.

An important property of our solution is that it does not change the underlying network structure (no re-wiring) nor does it force individual agents to permanently adopt any convention. The coordinators only coordinate agent exploration of the state-action space so the agents can reach a global convention. Intuitively, our approach works as follows. Individual agents interact normally through the network that govern their interactions. Each group of agents is assigned a coordinator that observes their convergence. If the agents in a group does not converge after some period of time, the coordinator then asks its group to try a recommended convention for a short period of time. However, since each coordinator only observes its own group, it is still possible that different coordinators recommend different conventions. However, because there is a hierarchy of coordinators, inconsistencies are guaranteed to be discovered higher in the hierarchy. Such arrangement does appear in real-life. For example, when a new technology is discovered, private companies that produce the new technology are left unregulated. If after a while no industry-wide standard (convention) is adopted, individual states may start enforcing some standards, and if needed a federal law may be put in place to ensure the quick reach to a convention. Once a convention is established it becomes self-enforcing without an external enforcement.

3. EXPERIMENTS AND DISCUSSION

Our experiments focus on applying social learning in scale-free networks to reach one of the two possible norms of the coordination game. The state observed by coordinators in this case is the ratio of its subordinates that adopted the first norm. So for example if coordinator X controls a cluster of 5 individuals, and 4 which adopted the first norm, then X's state is 0.8. Each coordinator has two control actions: to ask its subordinates to try either norm 1 or norm 2 for short period of time T_{try} . To avoid conflicting control actions, a coordinator does not issue a control action if a control action from its superior is currently being executed. The strategy of the coordinator for choosing a control action is simple: if the ratio of subordinates that adopted the first norm is between δ and $1-\delta$ (we used $\delta = 0.25$ in our experiments) this means no norm is adopted yet within the cluster controlled by the coordinator. The coordinator then chooses a consistent control action (e.g. if the ratio is 0.3, the coordinator asks subordinates to try the second norm). The results we report here have been obtained by randomly generating 10 different scale-free networks. Every network consists of 225 agents and individual agents execute Q-learning with learning rate = 0.1 and exploration rate = 0.1. To avoid initial bias, the Q-learning action values are initialized to uniformly random values between 0 and 1. The coordinators hierarchy is generated randomly such that the hierarchy has 5 levels and each coordinator controls a cluster of at least 5 individuals. Figure 2 illustrates how using a hierarchy of coordinators ensured the convergence to a global norm.

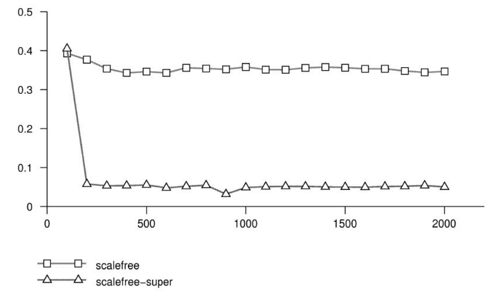


Figure 2: Comparing the percentage of the first norm adopted by agents for both social learning with and without the use of hierarchy coordinators in a sample run. Without coordinators no clear norm dominates the population, while with the hierarchy of coordinators a quick adoption of the second norm takes place.

4. REFERENCES

[1] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the international joint conference on Artifical intelligence*, IJCAI'07, pages 1507–1512, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[2] T. Sugawara. Emergence and stability of social conventions in conflict situations. In *Proceedings of the international joint conference on Artifical intelligence*, pages 371–378, 2011.

[3] D. Villatoro, J. Sabater-Mir, and S. Sen. Social instruments for robust convention emergence. In *IJCAI*, pages 420–425, 2011.

Towards Student/Teacher Learning in Sequential Decision Tasks

(Extended Abstract)

Lisa Torrey
St. Lawrence University
ltorrey@stlawu.edu

Matthew E. Taylor
Lafayette College
taylorm@lafayette.edu

Categories and Subject Descriptors

I.2.6 [**Learning**]: Miscellaneous

General Terms

Algorithms, Performance

Keywords

Reinforcement Learning, Inter-agent teaching, Transfer Learning

1. INTRODUCTION

Significant advances have been made in allowing agents to learn, both autonomously and with human guidance. However, less attention has been paid to the question of how agents could best teach each other. For instance, an existing robot in a factory should be able to instruct a newly arriving robot, even if it is from a different manufacturer, has a different knowledge representation, or is not optimal itself.

This work investigates teaching methods in sequential decision tasks. In particular, we consider a reinforcement learning student-agent that must learn from 1) autonomous exploration of the environment and 2) the guidance of another teacher-agent. In order to minimize inter-operability requirements, the teacher and student are presumed not to know each others' internal workings; teachers can only help students by suggesting actions. Furthermore, the teacher may have limited expertise in the student's task and should be careful not to over-advise the student. Our primary question: how should the teacher decide when to give advice?

This teaching context is related to the more well-studied problem of *transfer learning* [5], in which an agent uses knowledge from a source task to aid its learning in a target task, but differs in that we do not assume agents can directly access each others' internal knowledge. Another related area is *learning from experts* [1, 3], where agents may imitate experts or ask for their advice. Our approach differs because control is given to the teacher, rather than the student, and we focus on non-expert teachers. Our hope is that this paper enables and inspires the agents community to develop further methods by which agents can teach other agents.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. METHODS

Our methods build upon Probabilistic Policy Reuse [2] (PPR), which allows an agent to learn a task faster by taking advantage of an existing policy. The PPR method changes the action selection step of model-free RL methods. With probability ψ, the agent exploits an old policy; the rest of the time, it uses normal ϵ-greedy action selection. The value of ψ decays over time according to a decay rate v so that the agent makes less use of old policies as it improves its own. We use this method for teaching by letting ψ be the teacher's probability of giving advice.

However, for a teacher with limited expertise, PPR may not be the optimal teaching algorithm. A PPR teacher provides action advice with a global probability ψ that is uniform across all states. If the teacher is more confident in some states than others, it makes more sense for advice probabilities to be higher in some states than others. We therefore propose a new approach that uses *confidence measures* to make advice probabilities state-specific.

2.1 Confidence Measure

In our proposed approach, agents need to be able to estimate their confidence in a state. Because we assume limited expertise, agents may not have much data to work with. To allow meaningful estimates with limited data, we introduce a confidence measure called *update counting*. The update-count of a state indicates how many times a non-zero Q-value update has been made there. This measure has a straightforward tabular implementation in discrete settings, but it is also adaptable to continuous settings through tile coding. Update-counts can be associated with tiles; the update-count of a state is then the sum of the update-counts of its component tiles.

2.2 Advice Probabilities

Because our proposed approach builds upon PPR, our teachers compute a probability of giving advice in a state. We believe there are several desirable properties for advice probabilities. First, they should be higher in states where the teacher is more confident, so that teachers give advice in proportion to their expertise. Second, they should be capped at the ψ of the PPR algorithm, so that the confidence-based teaching method integrates cleanly with the PPR framework. Third, they should decay over time, so that teachers gradually decrease their guidance as the student learns.

We now propose an algorithm that computes a probability of giving advice $p(s)$ with the above properties. Let $c_t(s)$ and $c_s(s)$ represent the teacher and student confidences (i.e., update-counts) in that state, respectively.

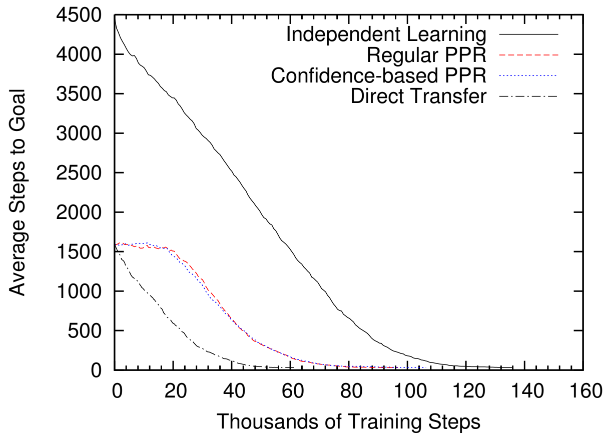


Figure 1: Q-learning agents in a maze

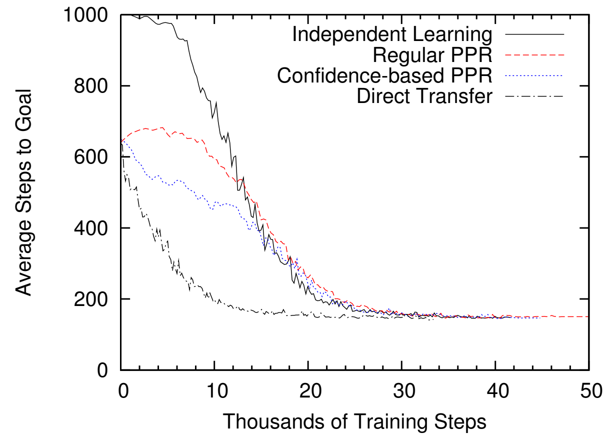


Figure 2: Sarsa learning students in mountain car

The *confidence-based PPR* algorithm computes:

$$p(s) = \begin{cases} 0 & \text{if } c_t(s) < 1 \\ \min\left(1 - \frac{c_s(s)}{c_t(s)+d}, \psi\right) & \text{if } c_t(s) \geq 1 \end{cases}$$

The first condition states that a teacher should never give advice if it has no knowledge about a state. The second dictates that the advice probability $p(s)$ depends on the relationship between the student's confidence and the teacher's confidence. In states where the teacher has higher confidence than the student, it gives advice with a higher probability, up to a maximum of ψ. As the student's confidence in a state grows, the teacher gives advice with lower probability. As the delay parameter d increases, teachers require students to reach higher levels of confidence before they stop giving advice.

3. EVALUATION

To evaluate this novel teaching algorithm, we perform teaching experiments in two domains. One is a discrete 20×20 maze, fully described in the earlier Ask-For-Help work [1], in which our teachers learn via standard tabular Q-learning. The other is mountain car, a benchmark continuous domain [4], in which our teachers learn via Sarsa with tile coding.

To produce teachers with limited expertise, we do not allow them to train until their policies converge. Instead, we train teachers for only 20 episodes. Each teacher then gives advice to students using regular PPR or confidence-based PPR. We average 100 teacher-student pairs for each experiment. Lower bounds for students are represented by *independent* agents, who learn without teachers. Upper bounds are represented by *direct-transfer* agents, who copy the teacher's entire Q-function, which our students do not have access to.

Figure 1 shows some results from the maze. The most effective parameter settings here are $v = 0.99$ and $d = 0$. As expected, students with teachers outperform students without teachers. The two teaching algorithms perform comparably in this domain. Confidence-based PPR does not outperform regular PPR because this domain lacks critical decision points, which means teachers can be less discerning about giving advice. However, we did find that it achieves comparable performance using two orders of magnitude less advice.

Figure 2 shows some results from mountain car. The most effective parameter settings here are $v = 0.99$ and $d = 100$. Both teaching methods again speed up learning, but in this domain confidence-based PPR outperforms regular PPR. Differences in the areas under these learning curves are statistically significant at the 95% confidence level.

There are several potential reasons that confidence-based PPR outperforms regular PPR in this domain. Mountain car may have some critical decision points, where the relative values of student and teacher confidences can play important roles in advice decisions. The tile coding in mountain car provides state-space generalization, which can cause student confidence to grow quickly in some sets of states. Using confidence-based PPR, teachers are able to back off quickly in these states, while still giving advice in less common states.

4. FUTURE WORK AND CONCLUSIONS

This paper contributes an initial study of algorithms that agents can use to teach each other in sequential decision tasks. We assume a broadly applicable setting, in which teachers and students interact only through action advice and in which teachers can have limited expertise.

There are many potential directions for future work in this area. For instance, teachers could explicitly reason about the expense of communication versus the expected gain, which would be appropriate in domains where communication has a non-zero cost. There could also be multiple teachers, with different areas of expertise, who must coordinate with each other.

5. REFERENCES

[1] J. A. Clouse. *On integrating apprentice learning and reinforcement learning*. PhD thesis, University of Massachusetts, 1996.

[2] F. Fernandez and M. Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the 5th International Conference on Autonomous Agents and Multiagent Systems*, 2006.

[3] B. Price and C. Boutilier. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:569–629, 2003.

[4] S. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158, 1996.

[5] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.

Bayes-Optimal Reinforcement Learning for Discrete Uncertainty Domains

(Extended Abstract)

Emma Brunskill
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA
ebrun@cs.cmu.edu

ABSTRACT

An important subclass of reinforcement learning problems are those that exhibit only discrete uncertainty: the agent's environment is known to be sampled from a finite set of possible worlds. In contrast to generic reinforcement learning problems, it is possible to efficiently compute the Bayes-optimal policy for many discrete uncertainty RL domains. We demonstrate empirically that the Bayes-optimal policy can result in substantially and significantly improved performance relative to a state of the art probably approximately correct RL algorithm. Our second contribution is to bound the error of using slightly noisy estimates of the discrete set of possible Markov decision process parameters during learning. We suggest that this is an important and probable situation, given such models will often be constructed from finite sets of noisy, real-world data. We demonstrate good empirical performance on a simulated machine repair problem when using noisy parameter estimates.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

reinforcement learning, MDPs, POMDPs

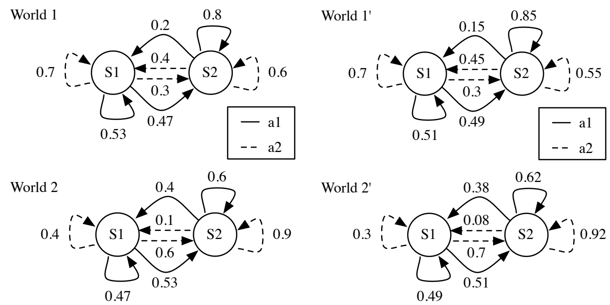
1. INTRODUCTION

Reinforcement learning (RL) is a critical challenge in artificial intelligence, because it seeks to address how an agent can autonomously learn to act well given uncertainty over how the world works. Model-based RL explicitly estimates parameters about the world dynamics and reward. Uncertainty over these parameters is typically allowed to be a continuous distribution. In contrast, there are many scenarios which are commonly represented by discrete uncertainty: the specific world is initially unknown, but there are only a finite set of possible worlds (which we represent by Markov decision processes). Such problems can be represented exactly as a finite-state partially observable MDP, where the discrete hidden state represents the true world (and associated parameters). While a related observation was made in passing by Poupart et al. [7] who noted that a discrete representation could be used to approximate continuous uncertainty, here we argue that many problems naturally exhibit finite uncertainty. For example, in customer relationship management, there may be several different types of customers, and the parameters of such customers can be estimated, but the type of a new customer is unknown.

Due to the finite nature of the uncertainty of these RL problems, we can use existing POMDP solvers to exactly compute a Bayes-optimal (or ϵ-optimal) policy. A Bayes-optimal RL policy is one that maximizes the expected discounted sum of future rewards over the specified time horizon, given an initial distribution of possible MDP model parameters. This is a different objective than Probably Approximately Correct (PAC) RL algorithms (e.g. [5, 2, 9]) which guarantee, with high probability, to select actions whose value is close to the value of the action that would be taken in the optimal policy if the MDP parameters were known, on all but a finite set of time steps. Though elegant, the number of time steps on which the algorithm may be far from optimal is often prohibitively large. To address this, practical instantiations of PAC RL algorithms typically involve a tuning parameter, resulting in good empirical performance, but eliminating theoretical guarantees. If we could solve for the Bayes-optimal policy by treating the problem as a POMDP [4], that would be appealing. However, in generic RL the model parameter values can be drawn from a real-valued set, this results in a continuous-state POMDP which are very challenging to solve, and prior Bayesian RL algorithms typically struggle to scale to large problems, and/or do not provide bounds on the computed policy's performance(e.g. [7, 8]).

However, it may often be possible to efficiently solve for an ϵ-Bayes-optimal policy in finite uncertainty domains. We first demonstrate the benefit of Bayes-optimal RL on an existing domain that naturally exhibits finite uncertainty. In the Wumpus grid world, an agent seeks to kill a wumpus without being first killed by the wumpus or falling into a pit. Our domain is almost identical to that described in [9], except that there are only 8 possible pit locations instead of

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.



(a) Discrete uncertainty RL (b) Erroneous parameters

Figure 1: The agent is placed in one of the two worlds, but it does not know which.

15. There are 3840 possible worlds, each with an associated wumpus location and set of pits; however the agent originally does not know which world it is in. We used the freely available APPL POMDP toolkit[1] to compute an e-Bayes-optimal policy (we set $e = 0.001$).

We compared to our approach to a PAC RL algorithm that computes a policy by adding a reward bonus to state-action pairs. This bonus is based on the variance of the possible hidden model parameters [9]. We focus our comparison to this variance-based bonus approach as the authors' approach outperformed a number of other approaches, including [1, 6].

In the Wumpus problem our Bayes-optimal policy has formal bounds on the performance, and empirically outperformed (mean=0.656, t-test $p < 0.001$) the variance bonus PAC RL approach without formal bounds (mean=0.478, tuning parameter=0.25), highlighting the benefit of Bayes-optimal RL. This, and many other, PAC RL approaches provide a fixed bonus for exploration, independent of the resulting possible benefit of such exploration, or the cost that may be incurred to perform this exploration, in contrast to Bayesian RL approaches.

2. IMPERFECT MODELS

We are interested in discrete uncertainty RL problems that capture real-world domains. In such environments, the possible models will generally be constructed from data. The model parameters estimated from the data will likely have a some error compared to the true generating parameters, due to limited data or local-optima finding fitting methods such as EM. For example, the true state of the world may be that the agent is acting in one of the two MDPs shown in Figure 1(a). However, the parameters of these two MDPs may have been estimated with some error, and the agent may think it is acting in one of the two MDPs shown in Figure 1(b). We can bound the error in the value function resulting from computing the value in a discrete uncertainty RL problem which has parameters that have some error relative to the true parameters:

THEOREM 1. *Let P denote a discrete uncertainty reinforcement learning problem $\langle S, A, R, \gamma, T_1, \ldots, T_M, b_0 \rangle$. In each transition model define $p(s'|s,a,m) = \theta_{sas'm}$. Let $\hat{P}$ be a second discrete uncertainty reinforcement learning problem $\langle S, A, R, \gamma, \hat{T}_1, \ldots, \hat{T}_M, \hat{b}_0 \rangle$. $\hat{\theta}_{sas'm'} = p(s'|s,a,m') = p(s'|s,a,m) + \epsilon_{sas'm}$, where $\sum_{s'} \epsilon_{sas'm} = 1$. $Q(b,s,a)$ is the optimal expected discounted sum of future reward from starting in belief state b and state s, and taking action a in RL problem P. $\hat{Q}(\hat{b},s,a)$ is the same quantity for the RL problem $\hat{P}$ for in belief state $\hat{b}$. Let*

$$\Delta_Q \equiv \max_{b,\hat{b},s,a} |Q(b,s,a) - \hat{Q}(\hat{b},s,a)|.$$

Then

$$\Delta_Q \leq \frac{\gamma V_{max} \max_{b,\hat{b},s,a} \sum_{s'} \left| \sum_i -\epsilon_{sas'i} b(i) + (\theta_{sas'i} + \epsilon_{sas'i})(b(i) - \hat{b}(i)) \right|}{1-\gamma}.$$

Given space limitations we omit the proof. In the worst case the bound provides little limitations. However, the bound is tight when no error is present: if $\epsilon_{sas'i} = 0 \; \forall i$, then $b(i) = \hat{b}(i)$ at all time steps, and $\Delta_Q = 0$, as expected.

We are interested in the empirical performance of a Bayes-optimal algorithm computed for a discrete uncertainty RL problem when the parameters provided are slightly different than the true domain MDPs' parameters. We have conducted preliminary experiments on a machine maintenance problem similar to that in [3]. These initial results suggest that a Bayes-optimal RL approach performs as well or better than the PAC RL variance bonus approach [9].

Our work is the first to examine Bayes-optimal RL in domains which inherently exhibit finite uncertainty. We have demonstrated that it is computationally tractable to compute Bayes-optimal policies in some such domains, and that such policies can perform significantly better than PAC RL approaches. We also provided a bound on the error of using slightly erroneous model parameters, which may be an important and common scenario in real-world situations.

3. ACKNOWLEDGMENTS

We thank Google for support.

4. REFERENCES

[1] J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *UAI*, 2009.

[2] R. I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 3:213–231, 2002.

[3] E. Delage and S. Mannor. *Operations Research.*

[4] M. Duff. *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes.* PhD thesis, University of Massassachusetts Amherst, 2002.

[5] M. J. Kearns and S. P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2–3):209–232, 2002.

[6] Z. Kolter and A. Y. Ng. Near-Bayesian exploration in polynomial time. In *ICML*, 2009.

[7] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *ICML*, 2006.

[8] S. Ross, B. Chaib-draa, and J. Pineau. Bayesian reinforcement learning in continuous POMDPs with application to robot navigation. In *ICRA*, 2008.

[9] J. Sorg, S. Singh, and R. L. Lewis. Variance-Based Rewards for Approximate Bayesian Reinforcement Learning. In *UAI*, 2010.

[1]http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/

Algorithms for Scaling in a General Episodic Memory (Extended Abstract)

Nate Derbinsky
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109-2121 USA
nlderbin@umich.edu

Justin Li
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109-2121 USA
justinnh@umich.edu

John E. Laird
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109-2121 USA
laird@umich.edu

ABSTRACT
Episodic memory endows autonomous agents with useful cognitive capabilities. However, for long-lived agents, there are numerous unexplored computational challenges in supporting useful episodic-memory functions while maintaining real-time reactivity. This paper presents and summarizes the evaluation of an algorithmic variant to the task-independent episodic memory of Soar that expands the class of tasks and cues the mechanism can support while remaining reactive over long agent lifetimes.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning

General Terms
Algorithms, Measurement, Performance, Experimentation

Keywords
Computational architectures for learning, Single agent learning

1. INTRODUCTION
Prior research has shown that autonomous agents with episodic memory, a task-independent, autobiographical store of prior experience [11], are more capable in problem solving, both individually [5][10] and with other agents [3][9]; better account for human psychological phenomena, such as those relating to memory blending [1] and emotional appraisal [4]; and are more believable as virtual characters [4] and long-term companions [8].

However, little work examines the computational challenges associated with maintaining effective and efficient access to experience over long periods of time. Most approaches to storing and retrieving episodic knowledge are task-specific (e.g. [9]) or apply to temporally limited problems (e.g. [5]).

By contrast, the episodic memory that is part of Soar [6] is task-independent and has been applied to complex, temporally extended tasks, such as action games [2] and mobile robotics [7]. To support effective and efficient episodic operation, the current mechanism makes specific design decisions within a space of algorithmic options. This paper presents and summarizes the evaluation of an algorithmic variant that expands the tasks and cues the mechanism can support while remaining reactive over long time periods, without adversely affecting performance.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4–8 June 2012, Valencia, Spain.

2. EPISODIC MEMORY IN SOAR
Soar's episodic memory [10] comprises three phases: (1) automatically *encoding* agent state; (2) *storing* this information as episodic knowledge; and (3) supporting *retrieval* at a later time.

The state of a Soar agent is represented as a connected di-graph. Episodic memory automatically encodes and permanently stores changes to this graph. Agents can later retrieve an episode by constructing a cue: a directed, connected, acyclic graph that specifies task-relevant relations and features. The *cue-matching* process identifies the "best" matching episode: the most recent episode that has the greatest number of structures in common with cue leaf nodes. Episodic memory then reconstructs this episode within a pre-specified region of the agent-state graph.

The cue-matching process (a) returns an episode if one exists that contains at least one feature in common with a cue leaf and (b) returns the "best" episode with respect to cue structure, cue leaves, and temporal recency. In the worst case, the encoding, storage, and retrieval operations scale at least linearly with state changes. However, exploiting regularities in state representation and dynamics may improve expected performance.

The current episodic-memory mechanism [2] exploits two regularities of agent state, both of which have been applied in the rule-matching literature. The first is *temporal contiguity*: agent-state changes between episodes will be few relative to agent-state size. The second is *structural regularity*: agent knowledge will reuse representational structure, and so over time, the number of distinct structures will be much smaller than the total experienced. Soar's episodic memory exploits these assumptions. Episodic knowledge is captured in a dynamic-graph index, composed of (1) a global structure, termed the Working-Memory Graph (WMG), which captures all *distinct* graph edges that have been encoded, and (2) a set of temporal intervals that capture when each edge of the WMG was added to/removed from agent state. The cue-matching algorithm uses a subset of the WMG as a discrimination network (termed the DNF Graph), through which it streams relevant *changes*, such as to evaluate episodes relative to the cue.

3. NOVEL ALGORITHMIC VARIANT
Our algorithmic variant exploits a stronger form of the structural-regularity assumption: over long agent lifetimes, the number of distinct structures represented within a *single* episode is likely to be much smaller than the *total* number of distinct structures. The algorithm exploits this assumption by building the DNF Graph incrementally, adding edges when they are relevant and removing them as they become obsolete. We hypothesized that over long agent lifetimes, this algorithm would improve retrieval time, especially for cues that match relatively recent episodes; however, tradeoffs exist. First, extra computation is required to dynamically

maintain the DNF Graph, which may exceed any performance gains, especially for simple cues. Second, the storage process must encode additional information: the most recent episode during which each WMG edge was represented in agent state.

4. EVALUATION

We implemented our algorithmic variant in Soar v9.3.1 and evaluated agents that use episodic memory for hours to days of real time. We applied over 100 cues in numerous tasks, spanning word-sense disambiguation, 44 instances of 12 planning domains (e.g. *Grid* and *Logistics*), 3 video games, and mobile robotics.

To evaluate scaling, we measured the time to perform episodic operations. For cue matching, we instrumented Soar to perform this operation 100 times for each cue at regular intervals. All experiments were performed on a Xeon L5520 2.26GHz CPU with 48GB RAM running 64-bit Ubuntu v10.10.

Our algorithm did not greatly impact performance in most tasks; however, it did enable a new, general capability for long-running agents using episodic memory: the management of long-term goals. This capability is best illustrated in the mobile-robotics domain, which has been used in prior work both in simulation and on physical hardware [7]. The agent perceives both physical perception data and symbolic representations of objects, rooms, and doorways. The agent's task is to explore a building, consisting of 100 offices, and then execute a fixed patrol pattern. While performing these tasks, the agent builds an internal map, which it uses for path planning and navigation. One cue in this domain asks, "*When was my desired destination doorway #5*?" The agent could examine episodes that followed this retrieval to recall progress made towards that goal. We ran the agent for 12 hours of real-time operation and measured performance every 300K episodes (~10 min.).

Figure 1 shows timing data to evaluate this cue, comparing maximum cue-matching time (in msec.) between Soar's current algorithm ("baseline") and our algorithmic "variant." Both algorithms exhibit growth in cue-matching time because some features in the cue are relevant with each new goal the agent encodes, but the goal of interest is increasingly distant in time. The first difference between the data sets is the number of episodes encoded over the 12-hour period: whereas the baseline algorithm encoded over 58 million, our variant encoded nearly 109 million. This difference has to do with optimizations we implemented in the incremental episodic-encoding algorithm, which resulted in an average of more than 50% improvement in encoding/storage speed in this task. However, both algorithms exhibit a common shift in behavior when the agent has finished exploring the building and proceeds to execute a patrol (~8M episodes for baseline, ~12M for variant). Before this point, the agent encodes new navigation goals much more frequently than after, and so the maximum search time grows more slowly after this point. Before the shift, our variant grows 3.6x slower than the baseline, and after it grows 4.9x slower. Furthermore, in fewer than 12 hours, the maximum computation time for the baseline algorithm grows above 50 msec., a level of reactivity that has been established in numerous domains, including video games, robotics, and HCI. By contrast, given the rate of growth in this task, our variant algorithm can continue to provide reactive real-time cue-matching performance for nearly 694M episodes (> 3 days of real time). The goal-management cue in the mobile-robotics domain is just one instance in a class of episodic cues and tasks in which there is a growth of distinct structures over the agent's lifetime. This data provides evidence that our algorithmic variant expands the problems in which Soar's episodic memory can support useful operation while agents remain reactive in dynamic environments over long time periods.

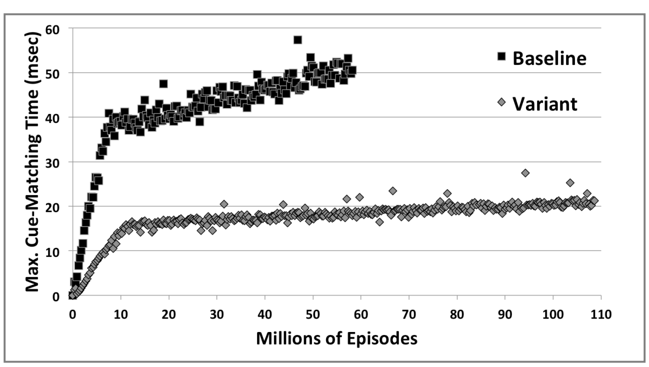


Figure 1. Timing comparison for goal-management cue in the mobile-robotics evaluation task.

5. ACKNOWLEDGMENTS

The authors acknowledge the funding support of the Air Force Office of Scientific Research under contract FA2386-10-1-4127.

6. REFERENCES

[1] Brom, C., Burkert, O., Kadlec, R. 2010. Timing in Episodic Memory for Virtual Characters. In Proc. of the IEEE 2010 Conf. on Computational Intelligence and Games, Copenhagen, Denmark, 305-312.

[2] Derbinsky, N., Laird, J. E. 2009. Efficiently Implementing Episodic Memory. In Proc. of the 8th Int. Conf. on Case-Based Reasoning, Seattle, WA, USA, 403-417.

[3] Deutsch, T. Gruber, A., Lang, R., Velik, R. Episodic Memory for Autonomous Agents. In Proc. of the 5th Int. Conf. on Human Sys. Interaction, Krakow, Poland, 621-626.

[4] Gomes, P. F., Martinho, C., Paiva, A. 2011. I've Been Here Before! Location and Appraisal in Memory Retrieval. In Proc. of the 10th AAMAS, Taipei, Taiwan, 1039-1046.

[5] Kuppuswamy, N. S., Cho, S., Kim, J. 2006. A Cognitive Control Architecture for an Artificial Creature using Episodic Memory. In Proc. of the 3rd SICE-ICASE Int. Joint Conf., Busan, Korea, 3104-3110.

[6] Laird, J. E. 2012. *The Soar Cognitive Architecture*. MIT Press, Cambridge.

[7] Laird, J. E., Derbinsky, N., Voigt, J. R. 2011. Performance Evaluation of Declarative Memory Systems in Soar. In Proc. of the 20th Behavior Representation in Modeling & Simulation Conf., Sundance, UT, USA, 33-40.

[8] Lim, M. Y., Aylett, R., Vargas, P. A., Ho, W. C., Dias, J. 2011. Human-like Memory Retrieval Mechanisms for Social Companions. In Proc. of the 10th AAMAS, Taipei, Taiwan, 1117-1118.

[9] Macedo, L., Cardoso, A. 2004. Exploration of Unknown Environments with Motivational Agents. In Proc. of the 3rd AAMAS, New York, NY, USA, 328-335.

[10] Nuxoll, A. M., Laird, J. E. 2011. Enhancing Intelligent Agents with Episodic Memory. *Cognitive Systems Research* (In Press).

[11] Tulving, E. 1983. Elements of Episodic Memory. Clarendon Press, Oxford.

Break with agents who listen to too many others (at least when making Boolean decisions!)

(Extended Abstract)

Daniel Epstein, Ana L. C. Bazzan, and André M. Machado
Instituto de Informática/UFRGS, 91501-970, P.Alegre, RS, Brazil
{depstein,bazzan,ammachado}@inf.ufrgs.br

ABSTRACT

In multiagent scenarios where decision-makers have to coordinate actions (e.g., minority and congestion games), previous works have shown that agents may reach coordination mostly by looking at past decisions. Not many works consider the structure behind agents' connections. When structure is indeed considered, it assumes some kind of random network with a given, fixed connectivity degree. The present paper departs from this approach mainly as follows. First, it considers network topologies based on preferential attachments (especially useful in social networks). Second, the formalism of random Boolean networks is used to allow agents to consider their acquaintances. Our results using preferential attachments and random Boolean networks show that an efficient equilibrium can be achieved, provided agents do experimentation. Also, we show that influential agents tend to consider few inputs in their Boolean functions.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Multiagent Systems, Self-organizing System, Minority Game, Traffic Simulation

1. INTRODUCTION

In multiagent systems, agents often face binary situations that require *coordination among the agents*. In minority games, previous works have shown that agents may reach appropriate levels of coordination, mostly by looking at the history of past decisions. Not many approaches consider any kind of structure of the network, i.e. how agents are connected. When structure is indeed considered, it assumes some kind of random network with a given, fixed connectivity degree. The present paper departs from the conventional approach in three main ways. First, it considers more realistic network topologies, based on preferential attachments [2]. This is especially useful in social networks. Second, the formalism of random Boolean networks is used to help agents to make decisions given their attachments (for example acquaintances). This is coupled with a reinforcement learning mechanism that allows agents to select strategies that are locally and globally efficient. Third, for the sake of illustration we use two different scenarios that differ greatly in the way the reward function is structured, namely the El Farol Bar Problem (EFBP, a kind of minority game) [1], and an iterated binary route choice scenario (adapted from [5]), henceforth referred as IRC. With this approach we target systems that adapt dynamically to changes in the environment, including other adaptive decision-makers.

Minority games have been the focus of many works. Regarding the general idea, the most similar works to the present paper have appeared in [4] and in [3]. In these cases, a kind of social network was considered. However, the connectivity was such that the average number of neighbors with whom each agent interacts was fixed. In the present paper we use a topology with preferential attachment, which basically means that a few nodes have big connectivity while the majority of the nodes are connected to just another node.

2. METHODS

Here we use RBN's to equip the agents with a decision-making framework. RBN's are made up of binary variables. N agents form a network and each must decide which binary action to perform. Each agent is represented by one of these binary variables. These in turn are, each, regulated by some other variables, which serve as inputs. The dynamical behavior of each agent, namely which action it will execute at the next time step, is governed by a logical rule based on a Boolean function. These functions specify, for each possible combination of K input values, the status of the regulated variable. Thus, being K the number of input variables regulating a given agent, since each of these inputs can be either on or off (1 or 0), the number of combinations of states of the K inputs is 2^K. For each of these combinations, a specific Boolean function must output either 1 or 0, thus the total number of Boolean functions over K inputs is 2^{2^K}. When $K = 2$, some of these functions are well-known (AND, OR, XOR, NAND, etc.) but in general there is no obvious semantics. More details and a simple example of this regulation process can be found in [3].

By using RBN's, in the EFBP for instance, we replace the space of possible strategies described in [1] by a set of

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Boolean functions. This also means that each node is connected to a given number of others. Hence, instead of having a random strategy, each node has random Boolean functions and uses them to determinate whether or not to go to the bar. Similarly, in the IRC scenario, instead of using a probabilistic approach to select a route (as in [5]), each driver agent explores a set of functions and a set of connections to other agents in order to make the route decision.

Each agent $i \in \mathcal{N}$ is a node in a random Boolean network and is connected to K_i others (notice that K hence may vary from agent to agent). Another parameter of the model is the number of functions each node possesses, $|\mathcal{F}|$.

Given the nature of minority games, the utility is highly coupled with the efficiency at system level. Efficiency is a domain-dependent concept related to the equilibrium of the particular system. In the EFBP the equilibrium calls for the bar accommodating ρ agents (in the original work $\rho = 60\%$). In the IRC, the equilibrium is such that route M carries $\rho = \frac{2}{3}$ of the traffic. Therefore agents must adapt and find functions that are efficient, i.e., provide high utility. Our approach for adaptation of the functions that are used at local level is based on an ε-greedy exploration process. At time step $t = 0$ one function from the set of $|\mathcal{F}_i|$ is assigned to each i. Then, at each further time step, the node decides to change the current function with probability φ. In case of a change, a new one is ε-greedy selected based on the utility it has provided so far. In the beginning of the simulation $\varepsilon = 1$ to allow exploration, but every time a function is changed, the value of ε is multiplied by $\delta < 1$.

According to f_i and to the value of the K_i entries, either 0 or 1 is output. In the EFBP scenario 0 means the agent stays at home; 1 means go to the bar. In the route choice scenario 0 and 1 mean the agent selects the main route (M) or its alternative respectively.

So far we have introduced the basic procedures, where each node has fixed connections, i.e., the set of K_i acquaintances does not change with time. Next, a variant called *change worst* (CW) is described. The CW variant is more utilitarian but also more realistic, in the sense that now agents evaluate the quality of their acquaintances. In the real-world, if someone is not performing well in the game, it will be likely to be labeled a black sheep and will be less and less considered as a part of a group (even if its bad performance may not be directly related to others). Thus, in this variant, each i looks at the reward r_j of each j it is connected and finds the agent with the worst reward. Let j^- be this agent. Agent i then marks j^- as a candidate for replacement, meaning that if i finds a better friend, it will no longer consider the action of j^- when deciding its own action. To replace j^-, i will look for a better connection among the best friends of its friends. Let j^+ be this agent. This does not affect j^+ since the relationship is not bidirectional. j^- however becomes less popular while j^+ increases its popularity and influence. In this process some nodes turn highly influential.

3. EXPERIMENTS AND RESULTS

The experiments performed (each repeated 30 times) consider the following values for the mentioned parameters: the horizon of simulation is $t_{max} = 1000$ time steps; $N = 900$ in both scenarios (this was the experimental setting in [5]); $|\mathcal{F}| = 10$; $\varphi \in \{0.1, 0.2, 0.9\}$ and $\delta \in \{0.999, 0.99, 0.9\}$.

In the case of the EFBP the main metric to be analyzed is the amount of agents ρ that go to the bar (as in [1] in which $\rho = 60\%$). In the IRC, for $N = 900$ agents, the reward function is balanced in a way that an equilibrium for the distribution of reward occurs when 600 agents select M. In the CW variant, besides these two metrics for each domain, we have also performed microscopic analysis about how the topology of the network changes (details of this dynamics available on demand). We do this aiming at understanding the role of degree in the reward of the agents, as well as the degree distribution in the efficiency of the whole system.

Due to lack of space we do not show the plots but notice that in the case in which K_i does not change, in both the EFBP and the IRC the system efficiency is reached (the time taken depends on values of the parameters, especially φ). In the CW variant, we notice that the convergence to the efficient action selection now takes more time, due to the fact that not only functions change in the CW case but also the connections. We have also conducted microscopic analysis, whose conclusion is that more influential nodes tend to be simple (low K), probably because, having less functions to try, they are more foreseeable, thus making the adaptation to them (by the neighbors) easier.

4. CONCLUSION

As it is common in minority and congestion games, instead of assuming knowledge about the history of the interactions, in the present paper we assume that agents interact in a social network. In particular, differently from previous works, the connectivity degree K is not homogeneous. Rather, agents are connected based on preferential attachment. We then let each agent i decide which action to do based on a Boolean function that maps the inputs of K_i acquaintances to i's output. Our approach admits some variants that were tested, as for instance whether or not to exchange acquaintances. We have found that using our approach, each agent is able to select an action that brings the system to the equilibrium, thus achieving the implicit coordination already observed by other authors. Moreover, using agent-based simulation, we are able to study microscopic properties such as how the influences change within time. The main finding is that more influential nodes tend to be simpler, i.e., have few inputs only.

5. REFERENCES

[1] B. Arthur. Inductive reasoning, bounded rationality and the bar problem. Technical Report 94–03–014, Santa Fe Institute, 1994.

[2] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] Daniel Epstein and Ana L. C. Bazzan. Decision making using random Boolean networks in the El Farol Bar Problem. In *Proc. of the Agent-Based Simulation (ABS / ECMS 2011)*.

[4] Aram Galstyan, Shashikiran Kolar, and Kristina Lerman. Resource allocation games with changing resource capacities. In *Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 145–152. ACM Press, 2003.

[5] F. Klügl and Ana L. C. Bazzan. Route decision behaviour in a commuting scenario. *Journal of Artificial Societies and Social Simulation*, 7(1), 2004.

Adaptive Agents on Evolving Networks
(Extended Abstract)

Ardeshir Kianercy
USC Information Sciences Institute
Marina del Rey, CA, USA
kianercy@usc.edu

Aram Galstyan
USC Information Sciences Institute
Marina del Rey, CA, USA
galstyan@usc.edu

Armen Allahverdyan
Yerevan Physics Institute
Yerevan, Armenia
aarmen@usc.edu

ABSTRACT

We propose a model of strategic network formation in repeated games where players adopt actions and connections simultaneously using a simple reinforcement learning scheme. We demonstrate that under certain plausible assumptions the dynamics of such systems can be described by so called replicator equations that characterize the co-evolution of agent strategies and network topology. Within this framework, the network structures emerging as a result of the game-dynamical interactions are described by the stable rest points of the replicator dynamics. In particular, we show using both simulations and analytical methods that for certain N-agent games the stable equilibria consist of star motifs as the main building blocks of the network.

Categories and Subject Descriptors

I.2 [**Artificial intelligence**]: Distributed Artificial Intelligence

General Terms

Theory, Algorithms

Keywords

Strategic network formation, Q-learning, Evolutionary game theory

1. INTRODUCTION

Many complex systems can be represented as networks where nodes correspond to entities and links encode interdependencies between them. Generally, statistical models of networks can be classified into two different approaches. In the first approach, networks are modeled via active nodes with a given distribution of links, where each node of the network represents a dynamical system. In this setting, one usually studies problems related to epidemic spreading, opinion formation, signaling and synchronization and so on. In the second approach, which is grounded mainly in a graph-theoretical approach, nodes are treated as passive elements. Instead, the main focus is on dynamics of link formation and network growth. Specifically, one is interested in algorithmic methods to build graphs formed by passive elements (nodes) and links, which evolve according to pre-specified, often local rules. This approach has produced important results on topological features of social, technological and biological networks.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

More recently, however, it has been realized that modeling individual and network dynamics separately is too limited to capture realistic behavior of networks. Indeed, most real–world networks are inherently complex dynamical systems, where both attributes of individuals (nodes) and topology of the network (links) can have inter–coupled dynamics. For instance, it is known that in social networks, nodes tend to divide into groups, or communities, of like-minded individuals. One can ask whether individuals become like-minded because they are connected via the network, or whether they form network connections because they are like-minded. Clearly, the distinction between the two scenarios is not clear-cut. Rather, the real world self-organizes by a combination of the two, the network changing in response to opinion and opinion changing in response to the network. Recent research has focused on the interplay between attribute and link dynamics (e.g., see [2, 4, 1] for a recent survey of the literature).

Here we suggest a simple model of co–evolving network that is based on the notion of interacting adaptive agents. Specifically, we consider network–augmented multi–agent systems where agents play repeated game with their neighbors, and adapt both their behaviors and the network ties depending on the outcome of their interactions. To adapt, agents use a simple learning mechanism to reinforce (punish) behaviors and network links that produce favorable (unfavorable) outcomes. Thus, the agent strategies and network topology evolve in tandem. We show that the collective evolution of such a system can be described by appropriately defined replicator dynamics equations. Originally suggested in the context of evolutionary game theory (e.g., see [3]), replicator equations have been used to model collective learning and adaptation in systems of interacting self–interested agents [5].

2. MODEL

Let us consider a set of agents that play repeated games with each other. We differentiate agents by indices $x, y, z, \ldots$. The time–dependent mixed strategies of agents can be characterized by a probability distribution over the choice of the neighbors and the actions. For instance, $p^i_{xy}(t)$ is the probability that the agent x will choose to play with agent y and perform action i at time t.

Furthermore, we assume that the agents adapt to their environment through a simple reinforcement mechanism. Among different reinforcement schemes, here we focus on (stateless) Q-learning [6]. In this case, it is known that the evolution of the agent strategies is governed by so called replicator equation [5]:

$$\frac{\dot{p}^i_{xy}}{p^i_{xy}} = \sum_j A^{ij}_{xy} p^j_{yx} - \sum_{i,j,\tilde{y}} A^{ij}_{x\tilde{y}} p^i_{x\tilde{y}} p^j_{\tilde{y}x} + T \sum_{\tilde{y},j} p^j_{x\tilde{y}} \ln \frac{p^j_{x\tilde{y}}}{p^i_{xy}} \quad (1)$$

We now make the assumption that the agents' strategies can be factorized as follows:

$$p^i_{xy} = c_{xy} p^i_x \,, \; \sum\nolimits_y c_{xy} = 1, \; \sum\nolimits_i p^i_x = 1. \tag{2}$$

Here p^i_x is the probability for agent x to select action i, whereas c_{xy} characterizes the strength of the directed link $x \to y$. Note that generally links are asymmetric, $c_{xy} \neq c_{yx}$.

Substituting 2 in 1, then taking summation of both sides in the resulting equation, once over y and then over i, we obtain

$$\begin{aligned} \frac{\dot{p}^i_x}{p^i_x} &= \sum_{\tilde{y},j} A^{ij}_{x\tilde{y}} c_{x\tilde{y}} c_{\tilde{y}x} p^j_{\tilde{y}} - \sum_{i,j,\tilde{y}} A^{ij}_{x\tilde{y}} c_{x\tilde{y}} c_{\tilde{y}x} p^i_x p^j_{\tilde{y}} \\ &+ T \sum_j p^j_x \ln(p^j_x / p^i_x) \qquad (3) \\ \frac{\dot{c}_{xy}}{c_{xy}} &= c_{yx} \sum_{i,j} A^{ij}_{xy} p^i_x p^j_y - \sum_{i,j,\tilde{y}} A^{ij}_{x\tilde{y}} c_{x\tilde{y}} c_{\tilde{y}x} p^i_x p^j_{\tilde{y}} \\ &+ T \sum_{\tilde{y}} c_{x\tilde{y}} \ln(c_{x\tilde{y}} / c_{xy}) \qquad (4) \end{aligned}$$

Equations 3, 4 describe the mutual evolution of the agents' strategies and the network structure. Here we focus on the case $T = 0$.

We should note that generally, the replicator dynamics (and Nash equilibria) in matrix games are invariant with respect to adding any column vector to the payoff matrix. However, this invariance does not hold in the present networked game. The reason for this is the following: if an agent does not have any incoming links (i.e., no other agent plays with him/her), then he always gets a zero reward. This poses a certain problem. For instance, consider the game of Prisoner's Dilemma where the payoff for mutual defection is P: In general, the outcome of the game should not depend on P as long as the structural properties of the payoff matrix is the same. However, in our case the situation is different. Indeed, if $P < 0$, an agent might decide to avoid the game by isolating himself (i.e., linking to agents that do not reciprocate), whereas for $P > 0$ the agent might be better of participating in a game.

To resolve this issue, we assume that every time a partner of agent x refuses to play, x receives a negative payoff $-c_p < 0$, which can be viewed as a *cost of isolation*. The introduction of this cost merely means adding a constant to the reward matrix. The adjusted reward matrix elements a_{ij} are given by $a_{ij} = b_{ij} + c_p$, where B is the game reward matrix and similar for all agents.

3. REST-POINTS AND LOCAL STABILITY

To examine the emergent network structures, we need to study the stable rest points of the replicator equations. Those rest points can be found by nullifying the right hand sides of Equations 3 and 4. Furthermore, the stability of those rest points are characterized by the eigenvalues of the corresponding Jacobian matrix

$$J = \begin{pmatrix} \frac{\partial \dot{c}_{ij}}{\partial c_{mn}} & \frac{\partial \dot{c}_{ij}}{\partial p_m} \\ \\ \frac{\partial \dot{p}_m}{\partial c_{ij}} & \frac{\partial \dot{p}_m}{\partial p_n} \end{pmatrix} = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \tag{5}$$

For two-action games, the diagonal blocks J_{11} and J_{22} are $L \times L$ and $N \times N$ square matrices, respectively, where $L = N(N-2)$. Similarly, J_{12} and J_{21} are $L \times N$ and $N \times L$ matrices, respectively.

We have performed thorough analytical characterization of the above system for three-player two-action games, which is the minimal system that exhibits non-trivial structural dynamics. In particular, we have demonstrated that for this class of games it is possible to characterize all the rest-points of the learning dynamics and examine their stability properties analytically.

We have also examined the behavior of the co-evolving system for large number of agents using both simulations and analytical techniques. We found that in the asymptotic limit, the networks formed by the reciprocated links (i.e., $c_{xy}c_{yx} \neq 0$) consists of $star$ motifs. A star graph S_n is a graph with n nodes and $n-1$ links, connecting one central node with the other $n-1$ nodes. We further observed that the basin of attraction of motifs shrinks as the motif size grows, so that smaller motifs are more prevalent.

As an example, we performed simulations for 100 agents interacting via the following Prisoner's Dilemma (PD) reward matrix:

$$B = \begin{pmatrix} (3,3) & (0,5) \\ (5,0) & (1,1) \end{pmatrix}$$

We run the simulation for 5000 different random initializations. When the cost of isolation is sufficiently large, $c_p \geq -b_{22}$, then the network breaks down into isolated star motifs. We observed that 91.26% of the motifs are S_2, 8.41% are S_3, 0.32% S_4, and 0.02% of S_5 star motifs. Within those stable networks, all the players choose the second action (*defect*). Furthermore, we have shown analytically that in a system with N-agents the star network S_N is in fact a stable rest point of the learning dynamics. Finally, when $c_p < -b_{22}$, the network structure is such that links are not reciprocated, so that the agents effectively do not interact.

4. DISCUSSION

In conclusion, we have presented a replicator–dynamics based framework for studying mutual evolution of network topology and agent behavior in a network–augmented system of interacting adaptive agents. By assuming that the agents' strategies allow appropriate factorization, we derived a system of a coupled replicator equations that describe the mutual evolution of agent behavior and network link structure. For N-player games, we reported both simulation and analytical results, which suggests that star-like structures are the most prevalent motifs emerging in our game-dynamical network formation. As future work, we plan to perform a more thorough analysis of the N-agent systems. Finally, we note that the main premise behind our model is that the strategies can be factorized according to Equations 2. While this assumption seems to be justified for certain games, its limitations need to be studied further.

5. ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation under grant No. 0916534 and the US AFOSR MURI grant No. FA9550-10-1-0569.

6. REFERENCES

[1] S. Castellano, C. Fortunato and V. Loreto. Statistical physics of social dynamics. *Rev. Mod. Phys.*, 81(2):591, 2009.

[2] T. Gross and B. Blasius. Adaptive coevolutionary networks: a review. *J. of the Royal Soc. Interface*, 5(20):259, 2008.

[3] J. Hofbauer and K. Sigmund. *Evolutionary games and Population dynamics*. Cambridge University Press, 1998.

[4] M. Perc and A. Szolnoki. Coevolutionary games–a mini review. *BioSystems*, 99(2):109–125, 2010.

[5] Y. Sato and J.P. Crutchfield. Coupled replicator equations for the dynamics of learning in multiagent systems. *Physical Review E*, 67(1), 2003.

[6] C.J.C.H. Watkins and P. Dayan. Technical note: Q-learning. *Machine learning*, 8(3):279–292, 1992.

A Common Gradient in Multi-agent Reinforcement Learning

(Extended Abstract)

Michael Kaisers, Daan Bloembergen, Karl Tuyls
Maastricht University, P.O. Box 616, 6200MD, Maastricht, The Netherlands
{michael.kaisers, daan.bloembergen, k.tuyls}@maastrichtuniversity.nl

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

General Terms

Algorithms, Theory

Keywords

Multi-agent learning, Evolutionary game theory, Dynamical Systems, Gradient Learning

1. INTRODUCTION

This article shows that seemingly diverse implementations of multi-agent reinforcement learning share the same basic building block in their learning dynamics: a mathematical term that is closely related to the gradient of the expected reward. Specifically, two independent branches of multi-agent learning research can be distinguished based on their respective assumptions and premises. The first branch assumes that the value function of the game is known to all players, which is then used to update the learning policy based on *Gradient Ascent*. Notable algorithms in this branch include Infinitesimal Gradient Ascent (IGA) [7], the variation Win or Learn Fast IGA (WoLF) [3] and the Weighted Policy Learner [1]. The second branch of multi-agent learning is concerned with learning in unknown environments, using interaction-based *Reinforcement Learning*, and contains those algorithms which have been shown to be formally connected to the replicator equations of *Evolutionary Game Theory*. In this case, the learning agent updates its policy based on a sequence of $\langle action, reward \rangle$ pairs that indicate the quality of the actions taken. Notable algorithms include Cross Learning (CL) [4], Regret Minimization (RM) [6], and Frequency Adjusted Q-learning (FAQ) [5]. This article demonstrates inherent similarities between these diverse families of algorithms by comparing their underlying learning dynamics, derived as the continuous time limit of their policy updates. These dynamics have already been investigated for algorithms from each family separately [1, 2, 3, 5, 6, 7], however, they have not yet been discussed in context of the relation to each other, and the origin of their similarity has not been discussed satisfactorily. In addition to the formal derivation, directional field plots of the learning dynamics in representative classes of two-player two-action games illustrate the similarities and strengthen the theoretical findings.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ANALYSIS

This section presents an overview of the dynamics of the different algorithms, and highlights their similarities. The discussion is limited to the domain of two-player normal form games for sake of clarity. In these games the payoff function can be represented by a bi-matrix. A general payoff matrix for two-action games, and two specific examples can be denoted as follows, where player 1 and 2 select row i or column j and receive a reward of A_{ij} or B_{ij} respectively:

$$\begin{pmatrix} A_{11}, B_{11} & A_{12}, B_{12} \\ A_{21}, B_{21} & A_{22}, B_{22} \end{pmatrix} \quad \begin{array}{c|cc} & C & D \\ \hline C & \frac{3}{5}, \frac{3}{5} & 0, 1 \\ D & 1, 0 & \frac{1}{5}, \frac{1}{5} \end{array} \quad \begin{array}{c|cc} & H & T \\ \hline H & 1, 0 & 0, 1 \\ T & 0, 1 & 1, 0 \end{array}$$

Payoff bi-matrix   Prisoner's Dilemma Matching Pennies

The dynamics within two-player two-action games have been studied individually for several algorithms. Let x and y denote the probability of selecting the first action by the first and second player respectively. Furthermore, α is the learning rate, $h = (1, -1)$, V is the value function and x^e is a probability belonging to a Nash equilibrium. FAQ also has a temperature parameter τ that controls the balance between exploration and exploitation. Unifying the notation from literature and factoring out the common gradient $\eth = \left[yhAh^T + A_{12} - A_{22} \right]$ the learning dynamics $\dot{x}$ for player 1 can be summarized as follows:

Alg.	$\dot{x}$ (change in probability of first action)
IGA	$\alpha\eth$
WoLF	$\eth \cdot \begin{cases} \alpha_{min} & \text{if } V(x,y) > V(x^e, y) \\ \alpha_{max} & \text{otherwise} \end{cases}$
WPL	$\alpha\eth \cdot \begin{cases} x & \text{if } \eth < 0 \\ (1-x) & \text{otherwise} \end{cases}$
CL	$\alpha x(1-x)\ \eth$
FAQ	$\alpha x(1-x) \left[\eth \cdot \tau^{-1} - \log \frac{x}{1-x} \right]$
RM	$\alpha x(1-x)\ \eth \cdot \begin{cases} (1 + \alpha x \eth)^{-1} & \text{if } \eth < 0 \\ (1 - \alpha(1-x)\eth)^{-1} & \text{otherwise} \end{cases}$

More generally, the first and second players' policy will be denoted as π and σ (i.e., $\pi_1 = x$ and $\sigma_1 = y$ in two-action games). The dynamics are illustrated in Figure 1 and are

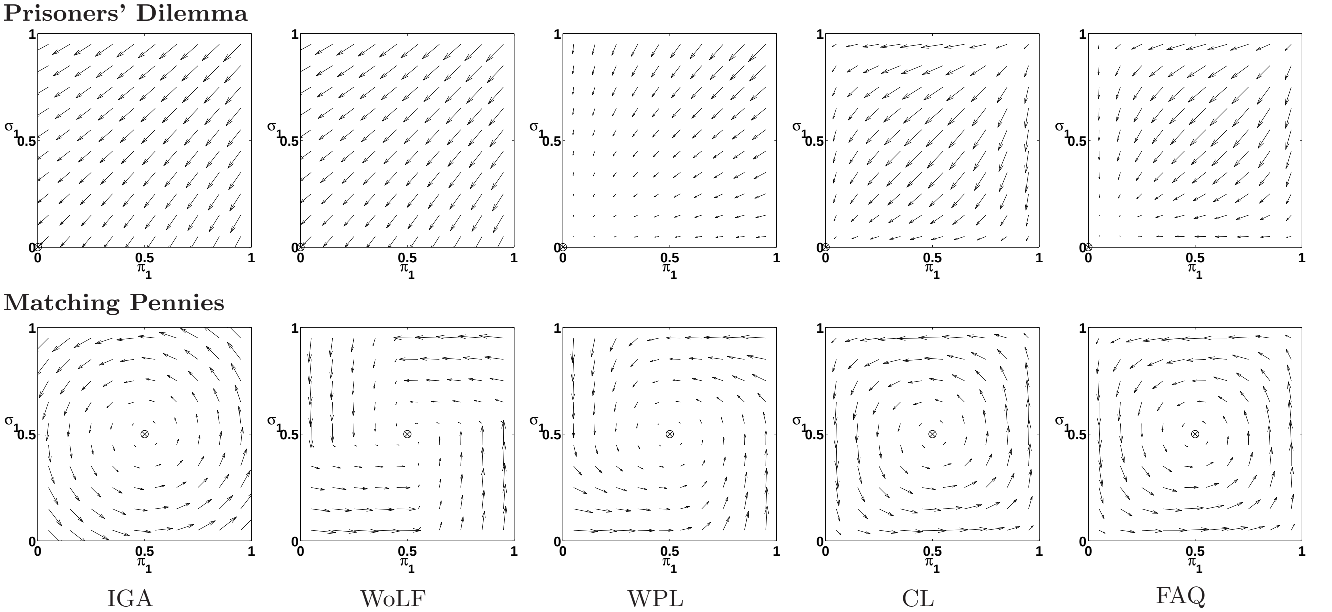


Figure 1: This figure shows the learning dynamics of the various algorithms in the Prisoners' Dilemma and Matching Pennies. The dynamics of RM are visually indistinguishable from CL in this scenario. The Nash Equilibria are indicated with $\otimes$.

now decomposed qualitatively for the class of two-agent normal form games. Let e_i denote the i^{th} unit vector and let n be the number of actions. Gradient Ascent is defined using the orthogonal projection function Φ, which projects the gradient onto the policy simplex thereby ensuring a valid policy (i.e., $\forall \pi_i : 0 \leq \pi_i \leq 1$).

$$\begin{aligned}\Delta\pi_i \leftarrow \alpha\frac{\partial V(\pi,\sigma)}{\partial \pi_i} &= \alpha \lim_{\delta\to 0}\frac{[\pi+\Phi(\delta e_i)]\,A\sigma^T - \pi A\sigma^T}{\delta}\\ &= \alpha\Phi(e_i)A\sigma^T = \alpha\left(e_iA\sigma^T - \frac{1}{n}\sum_j e_jA\sigma^T\right)\end{aligned}$$

In contrast, CL, FAQ and RM ensure validity of the policy update by making the update rule proportional to π. Incorporating proportional updating into the gradient-based policy update rule yields $\pi_i(t+1) \leftarrow \pi_i(t) + \pi_i\frac{\partial V(\pi,\sigma)}{\partial \pi_i}$.

In order to incorporate this different approach, the projection function Φ needs to change as well in order to properly map the weighted gradient. Intuitively, this can be achieved by using a weighted mean instead of a standard mean, such that $\hat{\Phi}(\zeta, w) = \zeta - \sum_j w_j\zeta_j$ where w is a normalized weight vector. Using $w = \pi$, this leads to the following dynamics:

$$\begin{aligned}\dot{\pi}_i &= \pi_i \lim_{\delta\to 0}\frac{\left[\pi+\hat{\Phi}(\delta e_i,\pi)\right]A\sigma^T - \pi A\sigma^T}{\delta}\\ &= \pi_i\hat{\Phi}(e_i,\pi)A\sigma^T = \pi_i[e_iA\sigma^T - \pi A\sigma^T]\end{aligned}$$

These resulting dynamics are exactly the dynamics of Cross Learning, showing that it is equivalent to Gradient Ascent with proportional updates. This provides a strong link between the two families of algorithms, gradient ascent on the one hand and independent multi-agent reinforcement learning on the other.

3. CONCLUSIONS

The main contributions can be summarized as follows: First, it is shown that the replicator dynamics are a prime building block of various types of independent reinforcement learning algorithms, such as Cross Learning, Regret Minimization, and Q-learning. Second, the replicator dynamics are shown to relate to the gradient of the expected reward, which forms the basis of Gradient Ascent. Both the replicator dynamics and gradient ascent have an update rule that is based on the difference between the expected reward of an action and the average expected reward over all actions. The only difference is how each action's update is weighted: gradient ascent assumes uniform weights as given by the gradient, whereas the replicator dynamics use the action-selection probabilities as weights.

4. REFERENCES

[1] S. Abdallah and V. Lesser. A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, 33(1):521–549, 2008.

[2] T. Börgers and R. Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1), November 1997.

[3] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[4] J.G. Cross. A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2):239, 1973.

[5] M. Kaisers and K. Tuyls. Frequency adjusted multi-agent Q-learning. In *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315, May, 10-14, 2010.

[6] T. Klos, G. van Ahee, and K. Tuyls. Evolutionary dynamics of regret minimization. *Machine Learning and Knowledge Discovery in Databases*, pages 82–96, 2010.

[7] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 541–548, 2000.

Combining Independent and Joint Learning: a Negotiation based Approach

(Extended Abstract)

Reinaldo A. C. Bianchi
Centro Universitario FEI
São Bernardo do Campo, Brazil.
rbianchi@fei.edu.br

Ana L. C. Bazzan
Instituto de Informática / PPGC
UFRGS, Brazil.
bazzan@inf.ufrgs.br

ABSTRACT

This work presents a new class of multiagent reinforcement learning algorithms that takes advantage of negotiation in order to improve the process of action selection. In this class of algorithms, agents use communication to cooperate and negotiate over the joint actions, thus enhancing the process of action selection. In this paper a new algorithm in this class is proposed: the Negotiation-based Q-Learning (NQL), which uses negotiation in the context of the Q-Learning algorithm. Results show that allowing negotiation between agents significantly enhances the performance of the multiagent learning process.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Theory

Keywords

Multiagent Reinforcement Learning, Negotiation

1. INTRODUCTION

In their work, Claus and Boutilier [1] have defined two forms of multiagent reinforcement learning (MARL): Independent learners (ILs), which apply Q-learning in the classic sense, ignoring the existence of other agents, and the Joint Action Learners (JALs) that, in contrast, learn the value of their own actions in conjunction with those of other agents.

The main problem of the JAL algorithm is the size of the representation of joint actions and states, which is a key factor that limits the use of algorithms for MARL in complex problems. Another known issue of the JAL is that it is not guaranteed that the chosen set of actions is coordinated with those of other agents. This may in turn lead to agents converging to different targets. Even in cooperative games, two agents could end up with two different, possibly uncoordinated (and hence inefficient), policies.

To cope with these problems, this work presents a new class of MARL algorithms that uses negotiation to choose the actions agents execute. Negotiation is employed by independent learning agents to implement cooperative actions when it is better than to act individually. Since a centralized solution is usually not feasible in large state-action spaces, decomposing the problem into subproblems using cooperation between independent agents in some parts of the environment is a way to reduce the complexity of the problem.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. COMBINING NEGOTIATION AND MULTIAGENT RL

To describe the class of algorithms that can be implemented by extending any MARL algorithm using negotiation, we propose a meta algorithm that is a high level description of how negotiation should be used in MARL, and serves as a template for extending traditional algorithms.

The main characteristic of the meta algorithm is that, before selecting what actions to perform, agents (when acting independently) negotiate with the aim of deciding which actions to take. This is shown in Algorithm 1, where s_i is the state that describes the system at a defined moment, as seen by agent i, and $\mathcal{A}$ is the set of actions to be used.

The negotiation algorithm used in this work is based on the one proposed by Fabregues and Sierra [2], which consists of "repeat a sequence of: a number of negotiation rounds up to the time limit, a selection of actions from the set of agreed upon joint plans and their execution. When new messages arrive, the algorithm check if it is a proposal. If it is, the message is stored in the set of proposals. This set is

Algorithm 1 The Negotiation MARL Meta-algorithm

```
Initialise Q̂_i arbitrarily.
repeat
   Observe the state s_i.
   Negotiate with other agents the actions A to be used.
   Execute action a_i.
   Receive the reinforcement r_i.
   Observe the next state s'_i.
   Update the values of Q̂_i.
   s_i ← s'_i.
until some stopping criteria is reached.
```

periodically checked to select a subset of proposals that can be jointly acceptable. The rest of proposals are rejected. If none of the proposals is good enough, a new deal is selected and the negotiation round is finished. Every proposal is stored until an answer is received or a timeout fires" [2].

We use their algorithm for the purpose of negotiating in a learning scenario as follows: an agent i uses an $\epsilon - greedy$ strategy to choose a set of actions $\vec{a}$. This set may contain actions for all agents, for some of the agents, or only for agent i itself. We remark that the latter occurs especially in the beginning of the learning process. If $\vec{a}$ involves two or more agents, i formulates a proposal regarding a joint action and sends this to other agents. This proposal also contains the expected utility of taking the joint action. During a certain period of time (set by a variable called "patience"), the agents collect proposals from the others.

Following this phase, i.e., after all agents have collected a set of proposal, each agent chooses the proposal that maximizes its expected return, and informs others which action was selected. It may occur that agent i decides to act as an IL, because the individual action a_i has the best utility for this agent when compared to those that were proposed. In this case, i must at least inform other agents of its action selection, so that the others can update their Q-values using the correct action for every agent.

It is important to notice that if the action the agent chooses to execute was a random exploration move, then the agent will not negotiate. It only informs other agents about this action. This is a characteristic that enables existing convergence proofs to hold for this new class of algorithms.

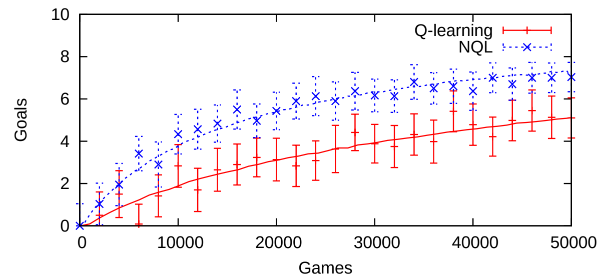
3. VALIDATION OF THE ALGORITHM

To validate the mentioned new class of MARL algorithms, the algorithm called Negotiation based Q-Learning (NQL) is proposed, which uses negotiation in the well-known RL algorithm Q-Learning. By means of negotiation, NQL can implement cooperative actions only when it is better than to act individually. Therefore, it can be seen as an IL that acts as a JAL in some situations.

Empirical evaluations were carried out in a simulator for the robot soccer domain that extends the one proposed by Littman in [3]. In this domain two teams, A and B, with 2 players each, compete in a 4×5 grid (agents are cooperative inside one team, competitive between the teams). The allowed actions are: move (north, south, east and west) or pass the ball to another agent. The action "pass the ball" from agent a_i to a_j is successful if there is no opponent between them. If there is an opponent, it will catch the ball and the action will fail. A complete description of this domain can be found in [3].

In the first experiment, two teams using the algorithms Q-learning and NQL play against an opponent team in which agents move randomly. Thirty training sessions were run for each team, with each session consisting of 5000 games of 10 trials. A trial finishes whenever a goal is scored by any of the agents. The parameters used in the experiments are identical to those used by Littman [3]).

Figure 1(top) shows the learning curves for the algorithms, presenting the average goal difference in each game (i.e., the goals scored by the learning team minus the goals scored by the opponent - in this case, the random team). It is possible to verify that the Q-Learning is outperformed by the NQL at the initial learning phase, and that as the games proceed, the



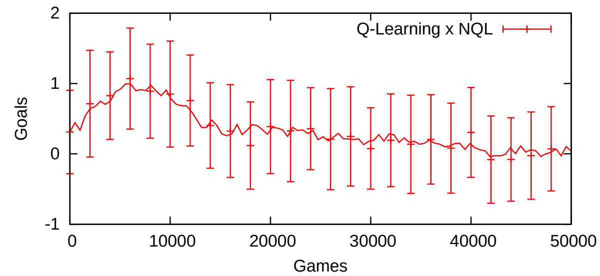


Figure 1: (top) Average goal difference for the Q-Learning and NQL learning against a random opponent and (bottom) for the Q-Learning versus NQL.

performance of both algorithms become similar, as expected. Student's t–test was used to verify the hypothesis that the use of negotiation speeds up the learning process. The value of T was computed for every game and the results showed that NQL is better than Q-learning when both are playing against a random opponent up to the 5000^{th} game, after which the results are comparable, with a level of confidence greater than 95%.

A second experiment tested the NQL when learning while playing against an opponent using Q-learning. Figure 1 (bottom) presents the learning curve (average of 30 training sessions, for $50,000$ games) for this experiment, where it can be clearly seen that the NQL algorithm is better at the beginning of the learning process and that after the a certain number of games the performance of this team becomes similar to the Q-learning, since all algorithms converge to equilibrium.

4. CONCLUSION

The experimental results obtained showed that the algorithm that use negotiation learned faster than in the case in which negotiation is not use. Future works include working on obtaining results in more complex domains, such as RoboCup Simulation and Small Size League robots, and applying this technique to other MARL algorithms.

5. REFERENCES

[1] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI '98*, pages 746–752,1998. AAAI.

[2] A. Fabregues and C. Sierra. An agent architecture for simultaneous bilateral negotiations. In *8th European Workshop on Multi-Agent Systems*, Paris, 2010.

[3] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML'94*, pages 157–163, 1994.

Modeling Difference Rewards for Multiagent Learning (Extended Abstract)

Scott Proper
Oregon State University
Corvallis, OR 97331, USA
proper@eecs.oregonstate.edu

Kagan Tumer
Oregon State University
Corvallis, OR 97331, USA
kagan.tumer@oregonstate.edu

ABSTRACT

Difference rewards (a particular instance of reward shaping) have been used to allow multiagent domains to scale to large numbers of agents, but they remain difficult to compute in many domains. We present an approach to modeling the global reward using function approximation that allows the quick computation of shaped difference rewards. We demonstrate how this model can result in significant improvements in behavior for two air traffic control problems. We show how the model of the global reward may be either learned on- or off-line using a linear combination of neural networks.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Experimentation

Keywords

Multiagent Coordination, Reward Shaping, Scaling, Air Traffic Control, Function Approximation, Neural Networks

1. INTRODUCTION

Reinforcement learning in large multiagent systems is particularly challenging because the agents in the system provide a constantly changing environment in which each agent needs to learn its task. Difference rewards which encourage good agent behavior by rewarding actions that are closely aligned with the desired overall system behavior. Difference rewards have been shown to perform very well in multiagent domains [1]. However it is not always possible to calculate the value of the difference reward, or even approximate it, due to complex system dynamics.

We mitigate this problem by using function approximation techniques to approximate the global reward signal, which we may then use to calculate an approximate difference reward. We use Tabular Linear Functions [2] to model the value of the global (system) reward. This model is then be used to calculate the difference reward. Our results show that we can greatly improve performance over learning on the system reward directly, and in some cases even outperform the true model of the reward signal.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. AIR TRAFFIC SIMULATION

We developed FEATS (Fast Event-based Air Traffic Simulator) to quickly simulate thousands of aircraft of different characteristics taking off from airports, navigating via waypoints and airways to their destination airport, and landing. This simulator is optimized for speed, simulating 26,000 flights/second. Individual simulations require a fraction of a second, allowing efficient experimentation with machine learning techniques. As in [3] we choose to make "meter fixes", rather than aircraft, into learning agents. We manage traffic by controlling aircraft separation distances – called "Miles in Trail" (MIT) separations – at meter fixes surrounding busy airports.

We used a linear combination of terms for measured congestion and delay to calculate the global (system) reward $G(z) = -(B(z) + \alpha C(z))$ where $B(z)$ is the delay penalty for all aircraft in the system, and $C(z)$ is the total congestion penalty. The relative importance of these two penalties is determined by the value $\alpha = 5$. $B(z)$ is the sum of minutes of delay suffered by all aircraft. $C(z)$ is given by $C(z) = \sum_{p \in P} \int_T \Theta(k_{p,t} - c_p)(k_{p,t} - c_p)^2 dt$, where P is the set of airports monitored by the simulation, $k_{p,t}$ is the number of aircraft that have landed in the past 15 minutes (a rolling time window), c_p is the capacity of airport p as defined by the FAA, and $\Theta(\cdot)$ is an indicator function that equals 1 when its argument is greater or equal to zero, and has a value of zero otherwise. Thus $C(z)$ penalizes states where airports become overcapacity. The quadratic penalty provides strong feedback to return the airport to FAA mandated capacities. We use an integral over time due to the fact that our simulation occurs in real time.

3. REWARD MODELING

The learning algorithm for each agent is a simple reinforcement learner using standard TD-update. Following [3], we use the **difference reward** $D_i(z) = G(z) - G(z - z_i)$ to provide the reward signal to each agent, where $z - z_i$ is a modified version of the normal action vector z in which agent i takes a "default" action (in our case, setting its "Miles in Trail" value to zero). As it is not possible to analytically compute this value for the air traffic domain, we learn an approximation $D_i(z) \approx v(z) - v(z - z_i)$, where $v(z) \approx G(z)$. We adapt "Tabular Linear Functions" (TLFs) from previous work [2] to approximate $G(z)$ in this manner.

TLFs have been shown to provide a simple, flexible framework to consider and incorporate different assumptions about the functional form of an approximated function and the set of relevant features. TLFs have previously been used for value function approximation. In this work, we use it to approximate the reward model.

A TLF is a sum over several terms. Each term is given by multiplying a weight and feature value, as with any linear function. Unlike standard linear functions, the weight of each term is given by an arbitrary function of other discretized (or "nominal") features.

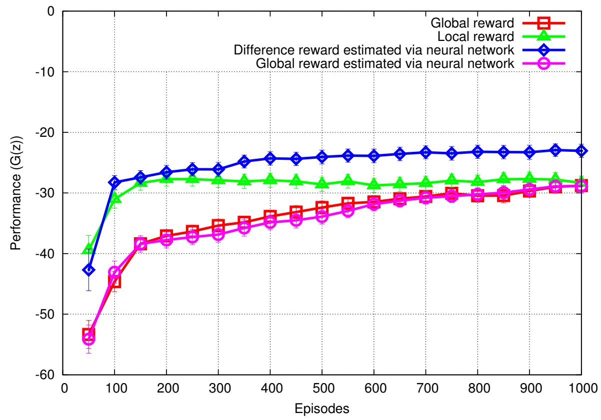


Figure 1: NAS simulation results: The approximated D reward outperforms other approaches, while the approximated global reward show that the approximation used is very accurate.

More formally, a tabular linear function is represented by Equation 1, which is a sum of n terms. Each term is a product of a linear feature ϕ_i and a weight θ_i. The features ϕ_i need not be distinct from each other. Each weight θ_i is a function of m_i nominal features $f_{i,1}, \ldots, f_{i,m_i}$.

$$v(z) = \sum_{i=1}^{n} \theta_i(f_{i,1}(z), \ldots, f_{i,m_i}(z))\phi_i(z) \qquad (1)$$

A TLF using tables to store the value of θ reduces to a linear function when there are no nominal features, i.e. when $\theta_1, \ldots, \theta_n$ are scalar values. However, this is effective only when each table is indexed by just a few nominal features. If this is not the case, we must also approximate the tables themselves. We thus applied TLFs with neural networks to approximate the reward model $G(z)$ for the air traffic simulation, using backpropogation to learn the model:

$$v(z) = \sum_{p \in P} (\theta_B^p(z_p) + \theta_C^p(z_p)) \qquad (2)$$

where z_p are the actions for the agents surrounding airport p, $\theta_B^p(\cdot)$ is a neural network approximating $B_p(z) = \sum_{a \in A_p} B_a(z)$, the sum of delays over all aircraft approaching p, and $\theta_C^p(\cdot)$ is a neural network approximating $C_p(z)$, the congestion penalty for a single airport. Each network has an input node for each action taken by the n_p agents (meter fixes) surrounding that airport, $n_p + 1$ hidden units, and 1 output. Given access to the above terms of $G(z)$, we can train each network separately, allowing a more accurate approximation. Note that a meter fix may control incoming traffic to multiple airports.

4. EXPERIMENTAL RESULTS

We performed experiments testing global and difference rewards, as well as a local reward based only on information available to individual agents. Each episode simulated a single traffic "rush" from start to finish. The actions taken by the meter fixes controlled the delay each aircraft suffered as it was routed through that fix. We used TLFs with neural networks approximating each term to estimate $G(z)$ and thus $D_i(z)$. We train each network offline using 10,000 randomly-generated examples.

Figure 1 shows that the estimated $D(z)$ significantly outperforms both local and global rewards. Learning using the estimated $G(z)$ compared to the true $G(z)$ shows that the estimate is very accurate.

In addition, we scaled up our experiments to 400 airports and 14,295 flights using a generic air traffic domain in a space about four times the size of the NAS. Figure 2 shows that the performance of the estimated difference reward greatly outperforms the other methods, particularly at this huge scale. Local reward performs very poorly: it does not allow for coordination between agents, which is critical in this domain. The estimated global reward does well in comparison to the true global reward, but performance does degrade slightly at this large scale. Difference rewards handle the increase in scale far better than any other method, despite the fact that it is using a learned model rather than the "true" difference reward.

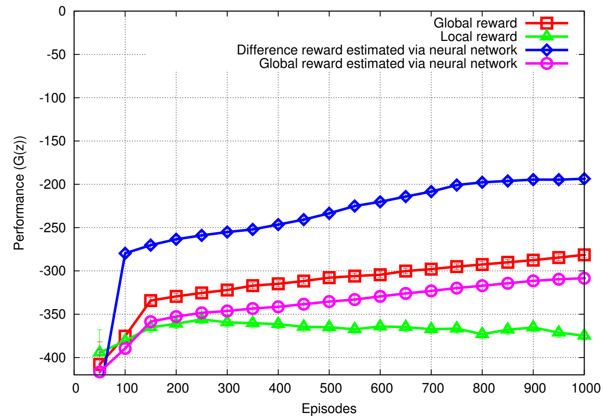


Figure 2: Results for 400 airports and 395 agents in the generic air traffic domain show that D does even better at larger scales (four times the size of the NAS).

5. DISCUSSION

We have shown that although calculating the difference reward for some multiagent domains may be impractical or impossible, it may still be possible to *estimate* D by learning a reward model of $G(z)$ using function approximation. We found that a sufficiently accurate model of $G(z)$ does in fact allow us to estimate D well enough to obtain improved behavior over learning on either the local or global rewards. In the case of air traffic control, a vast database of states and actions already exists, or may be generated via sufficiently sophisticated simulations. This makes learning a model of the reward function offline a practical approach for many domains. Future work includes continued experiments with model learning and the addition of states to our air traffic simulation, allowing agents to learn how to manage and route traffic by dynamically adapting to changing conditions.

Acknowledgements

This work was partially supported by NSF Grant CNS-0931591.

6. REFERENCES

[1] A. K. Agogino and K. Tumer. Analyzing and visualizing multiagent rewards in dynamic and stochastic environments. *Journal of Autonomous Agents and Multi Agent Systems*, 17(2):320–338, 2008.

[2] S. Proper and P. Tadepalli. Scaling model-based average-reward reinforcement learning for product delivery. In *ECML '06: Proceedings of the 17th European Conference on Machine Learning*, pages 735–742, 2006.

[3] K. Tumer and A. Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 330–337, Honolulu,HI, May 2007.

Revenue prediction in budget-constrained sequential auctions with complementarities

(Extended Abstract)

Sicco Verwer *
Radboud University Nijmegen, The Netherlands
S.Verwer@cs.ru.nl

Yingqian Zhang
Erasmus University Rotterdam, The Netherlands
yqzhang@ese.eur.nl

ABSTRACT

When multiple items are auctioned sequentially, the ordering of auctions plays an important role in the total revenue collected by the auctioneer. This is true especially with budget constrained bidders and the presence of complementarities among items. It is difficult to develop efficient algorithms for finding an optimal sequence of items. However, when historical data are available, it is possible to learn a model in order to predict the outcome of a given sequence. In this work, we show how to construct such a model, and provide methods that finds a good sequence for a new set of items given the learned model. We develop an auction simulator and design several experiment settings to test the performance of the proposed methods.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Design, Economics, Experimentation

Keywords

Sequential auctions, revenue maximization, learning

1. INTRODUCTION

Auctions are becoming increasingly popular for allocating resources or items in business-to-business and business-to-customer markets. Often sequential auctions are adopted in practice. Previous research has shown the revenue is heavily dependent on the ordering of items in sequential auctions [1], especially when bidders have budget constraints or when they have preference over bundles of items.

Much of the existing work that studies optimal ordering in auctions focuses on theoretical analysis on bidders' strategy and conditions when the optimal ordering exits. However, it is difficult to apply these results to actual auctions as they rely on strong assumptions which rarely hold in practice. In this paper, we develop a novel method for finding revenue-maximizing orderings that can generalize to real-world auctions. Our method is based on techniques from machine learning. It uses historical auction data in order to quickly learn which orderings have high expected revenue.

Our main contribution is providing a method that first transforms this information into a data set, then learns models for predicting the revenue of orderings, and finally uses a best-first search algorithm in order to find a good ordering for a new set of items. We implement an auction simulator, and design several experiment settings to test the performance of the proposed learning method.

2. LEARNING GOOD ORDERINGS

We assume there is a set of agents A who have budget constraints on purchasing items. Let $R = \{r_1, \ldots, r_l\}$ denote the collection of the item types, and the quantity of each item type can be more than 1. Each bidding agent A_i has a valuation for each type of item or each bundle of different item types $v_i : R \rightarrow \mathbb{R}^+$. In one round of auction, a set of items S with type set $R' \subseteq R$ will be auctioned sequentially with an order that is announced before the auction starts. We assume that the auction is repeated over time, and each auction sells possibly different items, with possibly different set of agents. At the end of each sequential auction, we have the following information at our disposal: (1) the ordering of auctioned items; and (2) the revenue of each sold item. Our goal of the sequential auction design is that given a set of items $r_1, \ldots, r_i$ for sale, deciding the ordering of items such that the revenue collected is maximized.

In order to simplify the learning problem, we make the following two modeling assumptions: (1) *Bidder independence*: in every round of sequential auctions, the set of participating bidders and their valuation functions are similar. (2) *Ordering independence*: the expected revenue for an item depends on which items were sold before and which items are still to be sold, but not on their ordering. The assumptions effectively reduce the difficulty of the learning problem to that of a standard machine learning setting: learn a single model from orderings and their rewards for predicting the expected reward for a given new input ordering.

We view the prediction of the revenue of an auction as a regression problem. Like an MDP, we split this problem into the subproblems of predicting the revenue of the auc-

*This work was performed while the first author worked at KU Leuven, Belgium.
The full version of the paper can be found in [2].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Algorithm 1 Computing a good ordering

Require: A set of items S, historical data on orderings and their revenues D, a maximum number of iterations m
Ensure: Returned is a good (high expected revenue) ordering

```
Transform D into a data set
for every item type r_T do
  Learn a regression model from D for predicting the revenue
  of item type r_T
end for
Initialize a hashtable H and a priority queue Q
Add the empty data row to Q
while Q is not empty and the size of H is less than m do
  Pop the row of features F with highest value v from Q
  if H does not contain F with a value >= v then
    Add F with value v to H
    Let L be the set of remaining items in F
    for every item type r_T of items in L do
      Let i_k be an item of Type r_T in L
      Let L' be a random ordering of L - i_k
      Use the learned models to evaluate the value v' of auc-
      tioning the ordering i_k L' after F
      Create new features F' for auctioning i_k after F
      Add F' to Q with value v + v'
    end for
  end if
end while
return The highest evaluated ordering
```

tioned items. We then sum these up to obtain the overall objective function: $V(r_1 \dots r_n) = \sum_{1 \leq k \leq n} R(r_k, \{r_j \mid j < k\}, \{r_l \mid k < l\})$, where $R(r_k, J, L)$ is a regression function that determines the expected revenue of r_k given that J was auctioned before and L will be auctioned afterwards. We use regression trees as a regression function and train it using features based on the items auctioned before and after the current item r_k. Currently, we provide the following features: (1) For every item type r_T, the amount of r_T items already auctioned; (2) For every item type r_T, the amount of r_T items still to be auctioned; (3) For every pair of item types r_T and $r_{T'}$, the difference between the amount of r_T and $r_{T'}$ items already auctioned. (4) For every item type r_T, the amount of revenue obtained from auctioning r_T items. These features model the influence of utility functions with complementarities and of budget constraints.

We transform an ordering and its obtained revenues into a data set using these 4 types of features. The data set obtained in this way can be given as input to any standard regression method from machine learning. In our case, we learn a regression tree for every item type using recursive partitioning techniques. The result is a set of predictive models for the expected revenue of items, and by summing these revenues we obtain the expected revenue of an auction.

To test an ordering, we employ a best-first search strategy that can be terminated at anytime in order to return the best solution found so far. We show how to compute a good ordering in Algorithm 1.

3. EXPERIMENTS

We developed an auction simulator and created three types of agents who have different bidding strategies: (1) *myopic* agents bid as soon as the asking price reaches their true value; (2) *smart* agents know all the valuation functions of all agents; and (3) *simulator* agents have access to the auction simulator. They bid what the smart agents bid, only if a run of the simulator results in a higher utility.

We first generate a set of agents, and run simulations of 250 random orderings of randomly selected items. These 250 orderings and their obtained revenues are transformed to a data set and provided to the regression tree learner. The resulting models are used to provide an ordering for a new set of randomly generated items using Algorithm 1 with a maximum number of 1000 iterations. We then run 10 simulations of this ordering and average the resulting revenues. We use this average revenue of different sets of items for the learned ordering method as a performance indicator.

We compare our method with two ordering strategies: (i) a random ordering, and (ii) a fixed ordering that auction the most valuable item first. In addition, we include a lower bound on the average revenue of an optimal ordering. This is computed by running simulations of 250 random orderings and selecting the one with the highest revenue.

Table 1: The performance with (top to bottom): 8 myopic agents, 8 smart agents, and 8 simulator agenst.

ordering strategy	1	2	3	4	5	sum
random	192	252	216	246	230	1136
most valuable first	176	214	184	251	223	1048
best first	204	247	225	245	229	1150
best first with sum	196	243	232	249	231	1151
best auction found	208	284	235	255	232	1214

ordering strategy	1	2	3	4	5	sum
random	260	237	231	158	262	1148
most valuable first	259	235	223	180	253	1150
best first	269	242	235	172	264	1182
best first with sum	269	228	232	176	266	1171
best auction found	278	259	237	183	279	1236

ordering strategy	1	2	3	4	5	sum
random	225	183	203	225	192	1028
most valuable first	212	182	204	225	180	1003
best first	240	181	209	234	193	1057
best first with sum	237	181	211	234	194	1057
best auction found	239	191	216	323	197	1166

The results with myopic agents, smart agents, and simulator agents are shown in Table 1.[1] Our preliminary results are encouraging as they show that our method is able to learn a good ordering for a complex auction setting (4 resource types with random values, popularities, and complementarities, of which 2 to 5 of each are auctioned every auction) from a small amount of examples (250 historical orderings along with their revenues). In addition, surprisingly they show that a random ordering strategy also performs well in complex auction settings. This is counter-intuitive. Further investigations are required to explain this result.

4. REFERENCES

[1] Wedad Elmaghraby. The importance of ordering in sequential auctions. *Manage. Sci.*, 49:673–682, May 2003.

[2] Sicco Verwer and Yingqian Zhang. Revenue prediction in budget-constrained sequential auctions with complementarities. Technical report, August 2011.

[1] We refer to [2] for more results.

An RL Approach to Common-Interest Continuous Action Games

(Extended Abstract)

Abdel Rodríguez[†‡]
abrodrig@vub.ac.be

Peter Vrancx[†]
pvrancx@vub.ac.be

Ricardo Grau[‡]
rgrau@uclv.edu.cu

Ann Nowé[†]
ann.nowe@vub.ac.be

[‡]Center of Studies in Informatics
Universidad Central "Marta Abreu" de Las Villas
Santa Clara, Villa Clara, Cuba

[†]Computational Modeling Lab.
Vrije Universiteit Brussel
Plainlaan 2 1050 Brussels, Belgium

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Multi-agent learning, Teamwork, coalition formation and coordination, Implicit Cooperation

1. ABSTRACT

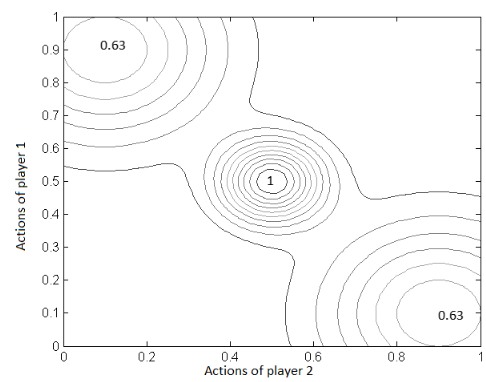
In this paper we present a reinforcement learning technique based on Learning Automata (LA), more specific Continuous Action Reinforcement Learning Automaton (CARLA), introduced by Howell et. al. in [2]. LA are policy iterators, which have shown good convergence results in discrete action games with independent learners. The approach presented in this paper allows LA to deal with continuous action spaces.

Recently, Rodríguez et al. [3] performed an analysis of the CARLA algorithm. The result of this analysis was an improvement of the CARLA method in terms of computation effort and local convergence properties. The improved automaton performs very well in single agent problems, but still has suboptimal performance with respect to global convergence in multi-agent settings.

The CARLA algorithm has successfully been applied to control problems [2, 1]. However in real world applications systems can be coupled and each subsystem is to be controlled by an individual controller. The interaction of these controllers can be considered as a common interest game. The interacting dynamics will have the learners converging to a suboptimal solution if the subsystems are controlled ignoring the existence of each other. In such a situation a better exploration of the joint-action space is required.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Figure 1: A two players game with three local optima. The contours represents combination of actions with the same reward.



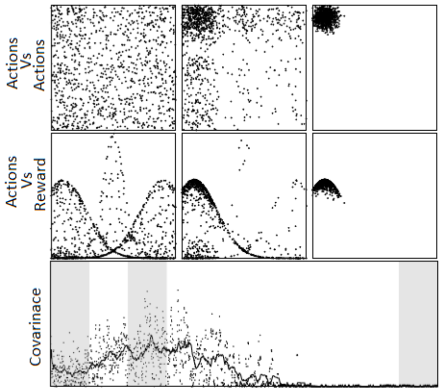
Exploring Selfish Reinforcement Learning (ESRL), introduced by [4], is an exploration method for independent LA playing a repeated discrete action game guaranteeing convergence to the optimal Nash equilibrium. The supporting idea of this method is that a set of independent LA will converge to one of the Nash equilibria of the game, but not necessarily one from the Pareto front. ESRL proposes that once the agents converge to a Nash equilibrium, at least two learners should delete the selected action from their action spaces and restart learning. This allows the agents to find all dominant equilibria and agree on the best one. As the more interesting Nash equilibria are often also stronger attractors, the agents can quite efficiently reach Pareto optimal Nash equilibria.

This paper introduces Exploring Selfish Continuous Action Reinforcement Learning Automaton (ESCARLA), an extension of the ESRL method to continuous action games.

The supporting idea of ESRL is to exclude actions after every exploration phase. The problem with applying this approach in continuous action games, is that it makes no sense for the agents to delete a single action. Instead, a vicinity around the action should be identified and excluded. Now the agent must estimate when it crosses the boundary of the basin of attraction of the local attractor.

In order to solve this problem we propose to use the absolute

Figure 2: Relation between covariance and exploration.



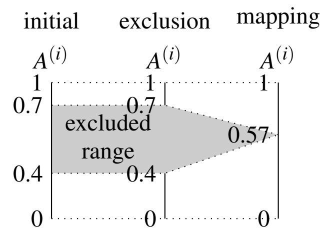
value of the covariance between the actions and rewards as a metric. Figure 1 shows the contour representation of a 2-players game example. There are three attractors in this example. The two local maxima located in the top left and bottom right corners have larger basins of attraction while the global maximum at the center has a narrower basin of attraction. Figure 2 shows the relation between the exploration and the covariance between actions and rewards from a single agent point of view. The first row shows a global view of the exploration. Three time intervals are shown. The first interval is the start of learning process (time-steps from 0 to 1000). The second interval is when the learners are reducing the global exploration (time-steps from 2000 to 3000). Notice this is a good time for deciding on which neighborhood to exclude. The last interval selected is when agents have converged to the local attractor (time-steps from 9000 to 10000). The second row shows the local information that the independent agents can access. The same time-steps are represented on each column but in this case we are plotting the selected actions on the horizontal axis and the corresponding reward on the vertical axis. The bottom row shows the absolute value of the covariance between actions and rewards over the whole learning process. Additionally, in order to have a better idea of how this covariance is evolving, the solid curve represents its average. The time-steps corresponding to the three moments introduced above are shaded in gray. This covariance reaches a low value at the beginning of learning since lots of explorations are performed by both agents. When the agents are exploring within the basin of attraction of a local attractor then the noise in the rewards observed by each agent is minimal so the covariance reaches its maximum. As agents converge to a locally superior strategy, less exploration is performed so therefore the covariance value drops down to zero. The safe region to exclude after the agent's actions have converged to the local optimum, can therefore be estimated at the moment when the absolute value of the covariance reaches its maximum value.

A good way of estimating this region is using the percentiles of the probability density function of the actions. For a given confidence value c we can define a region as shown in expression (1) where $\text{percentile}(p, f)$ represents the value where the probability density function f accumulates the probability p.

$$\left[\text{percentile}\left(\frac{1-c}{2}, f\right), \text{percentile}\left(1-\frac{1-c}{2}, f\right)\right] \qquad (1)$$

The proposal here is to let the agents start learning until they all converge. Then each agent should delete the region defined by (1) from its action space. Deleting the action range implies modifying

Figure 3: Mapping process. Actions from the non-deleted $\{[0,0.4], [0.7,1]\}$ range will be mapped into the original action space



the action space so we need to map the new one into a compact set again as shown in Figure 3. Then all agents must restart the learning process to converge to another attractor. After enough exploration the agents should compare all the results and pick up the strategy that gave them the highest score. In engineering applications we may either know the total amount of maxima of the problem or the desired performance. In such applications the agents could understand by enough exploration by finding all different maxima of the problem or by achieving the desired performance. Please note that we are assuming a common interest game, so therefore the agents can agree on a best combination of actions. The general algorithm is given next.

ESCARLA algorithm
```
repeat
    explore
    synchronize
until enough exploration
select best strategy
```

Exploration phase
```
initialize parameters
repeat
    sample action
    update strategy
    if maximum covariance
    then mark interval
until convergence
```

Synchronization phase
```
exclude marked interval
```

2. REFERENCES

[1] M. Howell and M. Best. On-line pid tuning for engine idle-speed control using continuous action reinforcement learning automata. *Control Engineering Practice*, 8(2):147 – 154, 2000.

[2] M. N. Howell, G. P. Frost, T. J. Gordon, and Q. H. Wu. Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics*, 7(3):263 – 276, 1997.

[3] A. Rodríguez, R. Grau, and A. Nowé. Continuous action reinforcement learning automata. performance and convergence. In J. Filipe and A. Fred, editors, *In Proceedings of the Third International Conference on Agents and Artificial Intelligence*, pages 473–478. SciTePress, January 2011.

[4] K. Verbeeck. *Coordinated Exploration in Multi-Agent Reiforcement Learning*. PhD thesis, Vrije Universiteit Brussel, Faculteit Wetenschappen, DINF, Computational Modeling Lab, September 2004.

Selecting judgment aggregation rules for NAO robots: an experimental approach

(Extended Abstract)

Vijayalakshmi Ganesan
University of Luxembourg
vijayalakshmi.ganesan.001@student.uni.lu

Marija Slavkovik
University of Luxembourg
marija.slavkovik@uni.lu

Sergio Sousa
University of Luxembourg
sergio.sousa.001@student.uni.lu

Leendert van der Torre
University of Luxembourg
leon.vandertorre@uni.lu

ABSTRACT

Social choice rules can be used to reach group decisions in multiagent systems. We consider judgment aggregation, the problem of aggregating answers to binary logically related questions. In general "fairness" is usually considered to be the main concern when selecting a social choice rule, however we believe that in judgment aggregation often a more relevant property is how efficient the rule is in truth tracking, that is, how often does it return the correct answer to the binary questions. Whereas "fairness" can be studied axiomatically, truth tracking efficiency needs to be studied experimentally. We accomplish the experimental analysis by constructing a multi-robot system.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

ExperimentationVerification

Keywords

Judgment aggregation, Truth tracking, multi-robot systems

1. BACKGROUND AND MOTIVATION

Social choice develops and studies methods for reaching group decisions, by aggregating individual opinions. Social choice is used in society in formal contexts, for instance in political elections, in informal context in everyday cooperation when preferences are aggregated, as well as in multiagent systems [1]. How individual opinions should be aggregated depends on the aggregation problem. In preference aggregation and voting the concern is to construct socially "fair" aggregation rules. Judgment aggregation theory is a social choice discipline concerned with the particular problem of aggregating individual binary answers, namely *judgments*, to a set of logically related questions. In some judgment aggregation contexts, when the questions admit objectively correct answer, a more important concern is to design *truth tracking efficient rules* [2]. For example the question "is there a red ball in the box?" has a correct answer, but an agent cannot know it if it cannot look inside the box and can only judge whether the answer is "yes" or "no". An aggregation rule is truth tracking efficient if it generates group decisions equivalent to the correct answers.

"Fairness" conditions can be studied axiomatically. Starting with the work of Kenneth Arrow, social choice theory is marked by impossibility results, which show that no preference aggregation rule exists that satisfies a minimal set of "fairness" conditions. Similar results have been shown to hold for judgment aggregation [3]. Truth tracking efficiency cannot be studied in this manner. One way to analyze this property is trough a probabilistic analysis, as done in [6]. However, to obtain a realistic estimate of the truth tracking efficiency of a judgment aggregation rule one needs to study this property experimentally: using judgment making robots in a setting where the rule is used.

The biggest challenge in experimentally analyzing social choice rules is the technical setup. Unlike a probabilistic analysis, where to add an agent or a question one needs to increase the value of a variable, in a realistic setting each addition of agents and questions is non-trivial. Our aim is to establish the foundations for experimental analysis of judgment aggregation rules. We develop a multi-robot system which aggregates judgment and use it to compare two judgment aggregation rules with respect to truth tracking.

2. JUDGMENT AGGREGATION RULES

Consider as an example the case when three robots (Lucy, Rosy and Jempi) need to determine if a sound is coming from a box (question x). They can make the conclusion x or not x by considering whether sound is heard (question p) and if a box is seen in the direction of sound (proposition q). These questions are related, namely $x \leftrightarrow (p \wedge q)$. Consider the *profile of judgments* given by the robot in Table 1 (white field). A basic question in judgment aggregation is whether to establish the simple majority supported answers on the reasons p and q and then use these to deduce an answer on

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

q (so called *premise-based procedure*), or to establish only the simple majority supported answers on the conclusion x (so called *conclusion-based procedure*). The two procedures can lead to different result on x as is in Table 1. In society, in particular in legal contexts where the efficiency in truth tracking is of higher importance than "fairness", the choice between a premise-based and conclusion-based procedure should be made based on the context of the problem [5]. In multiagent systems, experiments can be used to choose the better procedure.

Robots	p	q	x
Lucy	no	yes	no
Rosy	yes	no	no
Jempi	yes	yes	yes
Majority	yes	yes	no

Table 1: An example of judgment aggregation.

3. METHODOLOGY AND RESULTS

Our system consists of five NAO robots[1]. We extend the given example of establishing whether sound is coming from a box so that it is feasible given the sensors of the robots and using the *sound pressure inverse distance law*. This gives us the list of binary questions:

p: NAO can hear sound (with energy value E_1).
q: NAO can see a box at distance L (possibly in the direction of the sound).
r: NAO can hear sound second time (at $L/2$ distance with energy value E_2).
s: Sound energy value E_2 increases in proportion to distance covered, depending on L.
x: Sound is coming from inside the box.

The logical connection rules as the following: $\{(p \wedge q \wedge r \wedge s) \leftrightarrow x, q \rightarrow r, r \rightarrow s\}$.

We execute the technical framework by interfacing the agent programing language GOAL [4] and the NAO's robotic framework NaoQi through an Environment Interface Standard (EIS)-compliant Java interface. The robots send their judgments to an aggregator program, which then determines the group decisions by using either the premise-based or the conclusion-based procedure.

We obtained fifty multi-agent profiles. Of the tested profiles, we observed that thirty of them displayed a different result when the premise-based and conclusion-based procedure was used. This can be seen in Table 2, where I ="Inside" and O ="Outside" and $O(*)$ ="Outside, box close to sound source and in same line of vision".

No. of profiles	Inconsistency	Truth	PBP	CBP
25	Y	I	I	O
5	N	I	I	I
10	N	O	O	O
5	N	O(*)	I	I
5	Y	O(*)	I	O

Table 2: Aggregation Results for Profiles

The results of our experiment indicate that the premise-based procedure is best at truth tracking. When the sound actually comes from inside the box, this procedure scores 100% for almost every case of majority inconsistency seen (row 1 of the Table 2). When the robots hear sound directly without reflections,*i.e.,* when sound source is outside the box and the box is not in the line of vision of sound direction, the robots get accurate results on the direction of the sound and there is no inconsistency between the premise- and conclusion-based procefure(row 3 of the Table 2). When the sound comes from outside the box and in the line of vision of the box, the experiment fails by design and in some cases the conclusion based procedure is close to the truth (due to the nature of the experiment), but this is hard to quantify (row 4 and 5 of the Table 2).

[1]http://www.aldebaran-robotics.com/

4. DISCUSSION

Truth tracking efficiency has been probabilistically analyzed in [6], where the authors find that, if the agents have a low probability of making the objectively true judgment, another procedure, the *distance-based procedure*, outperforms the premise-based procedure. We calculated off-line the group decisions obtained from the collected profiles of judgments if the distance-based procedure was to be used and obtained that this procedure performs worse than the premise-based procedure. Since we do not have any estimates on the probability of a robot to report an objectively true judgment, our results are not strictly comparable.

The questions in our experiment, which are the reasons that support the conclusion, are not logically independent, so the probabilistic analysis such as the Condoret jury theorem cannot be used to explain the excellent truth tracking qualities of the premise-based procedure. Instead we consider this property to be due to the fact that all reasons in our experiment, unlike the conclusion x, are sensor-read values. It is our assumption that when the conclusion is such that can be read by a sensor, the premise-based procedure would lose its primacy.

The construction of the multi-robot system in which the robots can form judgments needs to be done once, and we have accomplished this step. A reoccurring challenge is to find questions on which there is more than one way to form a judgment by a robot. We have worked with only one set of questions. To the best of our knowledge, ours is the first effort in experimentally testing social choice rules on robots and more work is needed to establish the properties of an experiment that leads to a conclusive evaluation of rule truth tracking efficiency.

5. REFERENCES

[1] Boella, G. and Pigozzi, G. and Slavkovik, M. and van der Torre, L. Group Intention Is Social Choice with Commitment. In *COIN VI*, volume 6541 of *LNCS*, pages 152–171. Springer, 2011.

[2] Bozbay, I. and Dietrich, F. and Peters, H. Judgment aggregation in search for the truth. Technical Report 040, METEOR, Maastricht Research School of Economics of Technology and Organization, 2011.

[3] Dietrich, F. and List, C. Arrow's theorem in judgment aggregation. *Social Choice and Welfare*, 29(1):19–33, 2007.

[4] Hindriks, K. *GOAL Programming Guide(Draft)*. Delft University of Technology, 2010.

[5] J. R. Nash. A context-sensitive voting protocol paradigm for multimember courts. *Stanford Law Review*, 56:75–159, 2003.

[6] Pigozzi, G. and Hartmann, S. Aggregation in multiagent systems and the problem of truth-tracking. In *Proceedings of AAMAS'07*, pages 97:1–97:3. ACM, 2007.

Distance-Based Rules for Weighted Judgment Aggregation

(Extended Abstract)

Marija Slavkovik
University of Luxembourg
Computer Science and Communication
marija.slavkovik@uni.lu

Wojciech Jamroga
University of Luxembourg
Computer Science and Communication
wojtek.jamroga@uni.lu

ABSTRACT

Cooperating agents need to reach group decisions on several logically related issues. These decision-making problems are studied in social choice theory by the discipline of judgment aggregation. Judgment aggregation produces group decisions by aggregating individual answers to binary questions, however existing aggregation rules are defined for a very restricted setting, insufficient for aggregating opinions in a computer science contexts. We propose a family of distance-based judgment aggregation rules and study their 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

judgment aggregation, distance-based merging

1. BACKGROUND AND MOTIVATION

Social choice develops and analyzes methods for reaching group decisions by aggregating individual information. Judgment aggregation in particular explores how the truth-values, called judgments, that individuals assign to logically connected issues can be aggregated into a consistent set of truth values [7]. Judgment aggregation problems occur in computer science contexts, *e.g.,* [1], as well is in society in committee decision making contexts, such as juries and expert panels. There is a notable difference between problems in the two contexts. In society problems it can always be assumed that each individual is capable of making and stating an opinion on each issue, namely that the sets of individual judgments are *complete*. It is further assumed that each individual is equally competent to give an informed opinion on each issue, when compared to the other individuals, i.e., the judgments are non-weighted. These two assumptions cannot be plausibly made in computer science settings. For instance, a robot that does not have a microphone cannot produce opinions regarding sound issues. Moreover, one facial recognition program can be much better than another.

A judgment aggregation rule is a function that assigns a consistent set of truth values to a collection of individually assigned truth values. Like for other social choice methods, it is impossible to construct a judgment aggregation rule that satisfies a minimal set of desirable criteria [7]. There is a small number of rules developed in the judgment aggregation literature all of which are only applicable to complete judgment sets, see [6] for an overview.

Here, we develop and study the properties of a family of judgment aggregation rules for aggregating incomplete sets of judgments in the presence of weights. The family is constructed by generalizing the distance-based merging procedure of [3], inspired by belief merging [5]. Although multi-valued rules are considered in belief merging, weights associated with pairs of agents and issues have not been considered. Furthermore, the properties considered in belief merging are not identical to the properties of interest in judgment aggregation which we define and analyze here, with the exception of the unanimity property.

2. RULES AND PROPERITES

Let $\mathcal{L}_3$ be a ternary propositional logic, and $\models_3$ the entailment operator for this logic. A judgment aggregation problem is specified by: a set of agent names N, a consistent set of formulas $\mathcal{A} \subseteq \mathcal{L}_3$ called the agenda, a set of formulas $\mathcal{R} \subseteq \mathcal{L}_3$ called constraints and a set of truth values $T = \{0, \frac{1}{2}, 1\}$. We use the value $\frac{1}{2}$ to represent the case when no judgment has been assigned to an issue.

A judgment for $a \in \mathcal{A}$ is an assignment of a truth value from T. The collection of judgments assigned to each $a \in \mathcal{A}$ is called a judgment sequence A. We use A_j to denote the judgment in the sequence regarding $a_j \in \mathcal{A}$. Let n be the cardinality of N and m the cardinality of $\mathcal{A}$. A *profile of judgments* is a $n \times m$ matrix π with elements $\pi(i, j) \in T$.

A judgment sequence A is consistent if and only if it is a truth assignment such that $\mathcal{A} \cup \mathcal{R} \not\models_3 \perp$. A judgment sequence is complete if and only if for each $a \in \mathcal{A}$ either $a \in \hat{A}$ or $\neg a \in \hat{A}$. For each $\mathcal{A}, \mathcal{R}$ and $\models_3$ we can construct the set of all consistent judgment sequences $\Phi(\mathcal{A}, \mathcal{R}, \models_3)$. We write only Φ when the arguments are clear from the context. In defining the judgment aggregation problem we do not limit ourselves to a particular ternary logic that has to be used to represent the judgments. In theory, any ternary logic can be used. The choice of logic specifies the domain and co-domain of the judgment aggregation rule.

An aggregation function is a function $\odot : (\mathbb{R}^+)^n \mapsto \mathbb{R}^+$

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

which is non-decreasing (if $x \leq y$ then $\odot(x_1, \ldots, x, \ldots, x_n) \leq \odot(x_1, \ldots, y, \ldots, x_n)$) and satisfies the boundary condition: the infimum of $\odot(\mathbf{x})$ is 0 [4, p.3]. An aggregation function is: *symmetric* iff $\odot(\mathbf{x}) = \odot([\mathbf{x}]_\sigma)$ for every $\mathbf{x} \in (\mathbb{R}^+)^n$ and permutation σ [4, p.22]; *associative* iff $\odot(x) = x$ for all $x \in \mathbb{R}^+$ and $\odot(\mathbf{x}, \odot(\mathbf{x}'), \mathbf{x}'') = \odot(\mathbf{x}, \mathbf{x}', \mathbf{x}'')$ for all $\mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \bigcup_{n\in\mathbb{N}^0}(\mathbb{R}^+)^n$ [4, p.22].The most commonly used aggregation functions are the $\sum$, $\prod$, max and min. All of these functions are all symmetric and associative.

A function $\delta : \Phi \times \Phi \to \mathbb{R}^+$ is called a distance if, for all $x, y, z \in \Phi$, there holds: $\delta(x, y) \geq 0$ (non-negativity), $\delta(x, y) = \delta(y, x)$ (symmetry) and $\delta(x, x) = 0$ (reflexivity). A distance δ is called a *metric* on Φ if, for all $x, y, z \in \Phi$, there holds:$\delta(x, y) = 0$ if and only if $x = y$ (identity of indiscernible) and $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$ (triangle inequality) [2, p.3-4].

A collection of judgment weights W is also an $n \times m$ matrix whose elements $w(i, j) \in \mathbb{R}^+$, $\mathbb{R}^+$ being the interval of reals $[0, +\infty)$. If no weights are given, then $W = U$, where U is such that for each i and j, $w(i, j) = 1$. If only the weights associated with an agent are given then for each i, $w(i, 1) = w(i, 2) = \cdots = w(i, m)$; this kind of weights have been usually considered in belief merging. If only the weights associated with the relevance of each agenda issue are given then for each j, $w(1, j) = w(2, j) = \cdots = w(n, j)$. If agent i's judgment on issue a_j is useless, then $w(i, j)$ should be set to 0.

We can now define the weighted distance-based judgment aggregation rule Δ as follows.

Definition 1. *Let $\odot$ be an aggregation function, $\circledast$ a symmetric aggregation function, and δ a distance metric. The distance-based aggregator $\Delta^{\delta,\circledast,\odot}$ is a weighted judgment aggregation rule specified as*

$$\Delta^{\delta,\circledast,\odot}(\pi, W) = \underset{A\in\Phi}{\arg\min}\ \odot\left(\circledast\left(w(i,j)\cdot\delta(A_j, \pi(i,j))\right)_{j=1}^{m}\right)_{i=1}^{n}.$$

The well known distances: Hamming distance d_H, Taxicab distance d_T, and Drastic distance d_D can be defined as: $d_H(A, A') = \sum_i^m \delta_H(A_i, A'_i)$, $d_T(A, A') = \sum_i^m \delta_T(A_i, A'_i)$, $d_D(A, A') = max(\delta_H(A_i, A'_i)$, where $\delta_T(A, A') = |A_i - A'_i|$ while $\delta_H = 0$ when $A(a_i) = A'_i$ and $\delta_H = 1$ when $A_i \neq A'_i$.

A judgment aggregation rule for a particular aggregation problem in multi-agent systems can be selected by looking at the properties which that rule satisfies. Typically in social choice theory one does not study how to select a rule, but which properties can be satisfied at the same time by a rule. We show which types of $\odot, \circledast$ and δ guarantee that the resulting rule would satisfy a social-theoretic property and study the properties of some specific $\odot, \circledast$ and δ.

We introduce two auxiliary concepts and then define the social-theoretic properties for $\Delta^{\delta,\circledast,\odot}$ desirable for virtually any judgment aggregation context.

Let $M_{n\times m}$ and $M'_{n\times m}$ be matrices. M' is a σ-permutation of M, if it is obtained by permuting the rows of M using a permutation σ.

We consider the function m, a simple majority function which, for a given $a \in \mathcal{A}$, returns the judgment for a supported by a strict majority of agents, taking into account also the weights. Namely, for $\pi \in T^{n\times m}$, $W \in (\mathbb{R}^+)^{n\times m}$ and $V \in T^n$, let $N_v \subseteq N$ be the set of agents i for which, for a given $a_j \in \mathcal{A}$, $\pi(i, j) = v$. For $T = \{0, \frac{1}{2}, 1\}$, the function $m : T^n \times (\mathbb{R}^+)^{n\times m} \mapsto T$ is $m(\pi\nabla j, W) = v$ when $\sum_{i\in N_v} w(i, j) > \sum_{i\in N\setminus N_v} w(i, j)$ and $\frac{1}{2}$ otherwise. $Maj(\pi, W)$ is the sequence obtained by applying m to each pair of columns of π and W.

Since all the properties in the judgment aggregation literature, with the exception of [6] have been defined for a different function type than our Δ we need to construct corresponding definitions of the most interesting properties.

Definition 2. *Let Δ be a distance-based judgment aggregation rule specified by $\delta, \circledast, \odot$. Δ satisfies* **anonymity** *iff $\Delta(\pi, W) = \Delta([\pi]_\sigma, [W]_\sigma)$ for every $\pi \in T^{n\times m}$, every $W \in (\mathbb{R}^+)^{n\times m}$, and every permutation σ. Δ satisfies* **unanimity** *iff for every $W \in (\mathbb{R}^+)^{n\times m}$ and every $\pi \in T^{n\times m}$ such that $\pi_1 = \cdots = \pi_n = A$, $\Delta(\pi, W) = \{A\}$. Δ satisfies* **majority-preserving** *iff for every $\pi \in T^{n\times m}$ and every $W \in (\mathbb{R}^+)^{n\times m}$, $Maj(\pi, W) \in \Phi$ implies that $Maj(\pi, W) \in \Delta(\pi, W)$. Δ satisfies* **separability** *iff for every two profiles $\pi^1_{[n_1\times m]}$ and $\pi^2_{[n_2\times m]}$ (and corresponding $W^1_{[n_1\times m]}$, $W^2_{[n_2\times m]}$), the $[A \in \Delta(\pi^1, W^1)$ and $A \in \Delta(\pi^2, W^2)]$ implies that $A \in \Delta(\pi, W)$. Matrices π and W are the concatenations of π^1, π^2 and W^1, W^2 correspondingly.*

Proposition 2.1. *If $\odot$ and $\circledast$ are symmetric and associative then $\Delta^{\delta,\odot,\circledast}$ satisfies anonymity, unanimity and separability. $\Delta^{d_T,\Sigma}$ is majority-preserving.*

The proof of anonymity follows from the definition of symmetry. The proof for unanimity follows from the boundary condition of an aggregation function and the reflexivity property of δ. The proof of separability follows from the definition of associativity and the non-decreasing property of the $\odot$ function.

The property of majority-preservation is not satisfied by almost all distance-based rules. To prove that $\Delta^{d_T,\Sigma}$, one needs to swap the order of the sums in the rule definition. The property follows from the triangular inequality property of the metric. $\Delta^{d_H,\Sigma}$, $\Delta^{d_T,max}$, or even $\Delta^{d_D,max}$ are not majority-preserving. We can construct a function $d_e(A, A') = \Pi_{j=1}^m \delta_e(A(j), A'(j))$, where $\delta_e(x, y) = k^{\delta_H(x,y)}$, for which $\Delta^{d_e,\Pi}$ is a majority-preserving rule. However, $\Delta^{d_e,\Pi}(\pi, W) = \Delta^{d_h,\Sigma}(\pi, W)$ for each π, W. It is our conjecture that $\Delta^{d_T,\Sigma}$ is the only majority-preserving operator.

3. REFERENCES

[1] G. Boella, G. Pigozzi, M. Slavkovik, and L. van der Torre. A satisficing agreements model. In *Proceedings of COIN@WI-IAT*, 2011.

[2] M.M. Deza and E. Deza. *Encyclopedia of Distances.* Springer, 2009.

[3] U. Endriss, U. Grandi, and D. Porello. Complexity of winner determination and strategic manipulation in judgment aggregation. In *Proceedings of the 3rd COMSOC*, 2010.

[4] M. Grabisch, J-L. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions.* Cambridge University Press, 2009.

[5] S. Konieczny and R. Pino-Pérez. Logic based merging. *Journal of Philosophical Logic*, 40(2):239–270, 2011.

[6] J. Lang, G. Pigozzi, M. Slavkovik, and L. van der Torre. Judgment aggregation rules based on minimization. In *TARK*, pages 238–246, 2011.

[7] L. List and B. Polak. Introduction to judgment aggregation. *Journal of Economic Theory*, 145(2):441 – 466, 2010.

Bribery in Voting Over Combinatorial Domains Is Easy (Extended Abstract)

Nicholas Mattei
University of Kentucky
Lexington, KY, USA
nick.mattei@uky.edu

Maria Silvia Pini
University of Padova
Padova, Italy
mpini@math.unipd.it

Francesca Rossi
University of Padova
Padova, Italy
frossi@math.unipd.it

K. Brent Venable
University of Padova
Padova, Italy
kvenable@math.unipd.it

ABSTRACT

We investigate the computational complexity of finding optimal bribery schemes in voting domains where the candidate set is the Cartesian product of a set of variables and agents' preferences are represented as CP-nets. We show that, in most cases, the bribery problem is easy. This also holds for some cases of k-approval, where bribery is difficult in traditional domains.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Theory, Algorithms

Keywords

Voting Protocols, Bribery, Manipulation, Complexity

1. INTRODUCTION

Making collective decisions is a challenging task for both humans and autonomous agents. Computational social choice focuses on computational questions regarding group decision making [3]. In this document we consider a scenario where a collection of agents use the CP-net formalism to compactly represent their preferences over a common set of issues that may have conditional dependencies [2].

When voting [1] is structured as the combination of several decisions, one natural method to determine a winner is to decide on an issue by issue basis, while the other natural approach is to aggregate the agents' votes over complete combinations of issues. We consider both approaches and we study elections via sequential (that is, issue by issue) majority (SM), plurality (OP), veto (OV), and k-approval (OK).

In this setting, we study the bribery problem, which is when an outside agent with a limited budget attempts to affect the outcome of an election by paying some of the agents to change their preferences [6]. We consider several cost schemes to compute the cost of changing a vote of an agent in response to a briber's request, which are based on the actual changes to be made in the CP-net. We show that, in most cases, bribery in combinatorial domains is easy. This also holds for some cases of OK, where bribery is difficult in traditional domains.

2. VOTING WITH CP-NETS

Our setting consists of a set of n agents with preferences over a common set of candidates with a combinatorial structure: there is a common set of m binary issues and the set of candidates is the Cartesian product of their domains. Each candidate (or outcome) is an assignment of values to all issues, thus we have 2^m candidates.

We assume each agent expresses its preferences over the candidates via an acyclic CP-net [2]. CP-nets are sets of *conditional preference statements* (cp-statements) each stating a total order over the values of a variable (say X), possibly depending on each combination of values of a set of other variables (say $X_1,\ldots,X_n$). X is said the dependent variable and $X_1,\ldots,X_n$ are the parents of X. Acyclic CP-nets are CP-nets where the dependency graph (with arcs from parents to dependent variables) does not have cycles.

Voting theory [1] provides many voting rules to aggregate agents' preferences. Each rule takes, as input, a partial or complete preference ordering of the agents and gives, as output, the "winner" outcome (the best outcome according to the rule). When there are more than two candidates, there are many voting rules one could use and we consider three. In **plurality** the candidate ranked in first place by the largest number of voters wins. When there are two candidates, plurality coincides with majority. In **veto** each voter chooses a candidate to veto and the candidate with the least number of vetoes wins. In ***k*-approval** each voter labels k candidates as approved or not and the candidate with the most approvals wins.

When we use a sequential approach to voting, we require a total ordering O over the issues so that, in each CP-net, each issue must be independent of all issues following it in the ordering O. A *profile* (P,O) is a collection P of n CP-nets over m common issues and a total ordering O over the issues that satisfies the above property. This is called an O-legal profile in [7]. The CP-nets appearing in such profiles do not necessarily have the same dependency graphs.

3. COMBINATORIAL BRIBERY

The bribery problem we consider is parametric with respect to three items: the way a winner is chosen from the given profile, the

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

	SM	SMw	OP(IV) OV(IV) OK*(IV)	OP(DV,IV+DV) OV(DV,IV+DV) OK*(DV,IV+DV)
C_{EQUAL}	NP-c	NP-c	P	P
C_{FLIP}	P	NP-c	P	P
C_{LEVEL}	P	NP-c	P	?

Table 1: Our complexity results for the combinatorial bribery problem. OK* stands for OK when k is a power of 2.

allowed bribery actions, and the cost scheme for such actions.

In our domain, agents have a CP-net instead of an explicit outcome ordering. Therefore, we define the bribery actions as changes made directly to the cp-statements within the CP-net of an agent. Since we consider binary issues, changing a cp-statement means flipping the positions of the two values of an issue. In a CP-net, a cp-statement is associated to a certain issue, and issues are of two kinds: independent and dependent. We distinguish bribery actions on these two kinds of issues denoting with IV (resp. DV) the situation in which the briber asks for flips only in cp-statement related to independent (resp. dependent) issues. When both are allowed, we write IV+DV.

Also the cost schemes we consider are defined in terms of the amount of change the agents have to make on their CP-nets in order to comply with the briber's request. In particular, we consider:

$\mathbf{C_{EQUAL}}$: A unit cost allows any number of flips in a CP-net.

$\mathbf{C_{FLIP}}$: The cost is the total number of flips in the CP-net.

$\mathbf{C_{LEVEL}}$: An issue which is closer to be independent is regarded as more important. We may link this importance to the cost of a flip: the cost of changing a CP-net is the total number of flips performed in the cp-statements, each weighted according to the level of the relevant issue. More precisely: $\sum_x flip(x) \times (k+1-level(x))$, where x ranges over the issues, k is the number of levels in the CP-net, $flip(x)$ is the number of flips performed in cp-statements associated to x, and $level(x)$ is defined recursively as: $level(x) = 1$ if x is an independent issue; otherwise, $level(x) = i+1$ if all parents of x are in levels $\{1,\dots,i\}$ and there is a parent in level i.

We can now state the **Combinatorial bribery problem**: We are given a profile (P,O) where P is a collection of n compact CP-nets with m binary issues and O is a total ordering of the m issues, a budget B, an outcome p, and bribing cost vector $\vec{Q}$ (each voter has its own cost, to be multiplied by the cost of the bribery actions according to the cost scheme). With this input, we want to know if there is a way for an outside actor to make p win in profile (P,O) with winner determination rule $D \in \{SM, OP, OV, OK\}$, by using bribery actions according to $A \in \{IV, DV, IV+DV\}$, and by paying according to scheme $C \in \{C_{\text{EQUAL}}, C_{\text{FLIP}}, C_{\text{LEVEL}}\}$ and bribing cost vector $\vec{Q}$, without exceeding B.

Table 1 shows the computational complexity of this problem, considering all possible combinations of winner determination rules, bribery actions, and cost schemes.

Starting from the first column, the NP-completeness result regarding SM with C_{EQUAL} is obtained via a reduction from the OPTIMAL LOBBYING (OL) problem [4]. Bribery with SM is instead easy with cost schemes C_{FLIP} and C_{LEVEL}: since we are working level by level, at each level we can select the agents to bribe by starting from the cheapest ones (according to $\vec{Q}$). The resulting preferred value for this issue can then be propagated in all CP-nets.

The second column shows SM with weighted voters. The results in this case are obtained via polynomial reductions from plurality-weighted-$bribery which was shown to be NP-complete in [6].

All the other entries of Table 1 relate to the extension to a combinatorial setting of the result by Faliszewski [5], which shows that plurality bribery in single issue elections with nonuniform cost functions is in P through the use of flow networks. The algorithm requires the enumeration of all possible elements of the candidate set as part of the construction of the flow network. In our model, the number of candidates can be exponential in the size of the input, so we cannot use that construction directly. We show that a similar technique works for OP and for all costs (except C_{LEVEL} when the briber can act on both dependent and independent issues) by considering only a polynomial number of candidates. For C_{LEVEL}, enumerating a polynomial number of cheapest (non-voted) alternatives for a voter becomes difficult making the flow-based approach non-applicable and, therefore, we conjecture this case is difficult.

The corresponding results for OV are obtained by the same line of reasoning and by noting that the worst outcome of a CP-net is the optimal outcome of the "reversed" CP-net, that is, the CP-net obtained by reversing the total orderings in all the cp-statements.

In OK, each agent gives its top k outcomes according to some linearization. When k is a power of two, it is possible to prove that if the top outcome is fixed, the next k outcomes must follow in some unique order. Therefore, it is possible to treat the top k outcomes as one bundle and to apply the flow-based approach in order to decide the cheapest bribery scheme to elevate the bundle that includes p into the winning set. We note that this is in contrast with the traditional bribery domain where the bribery problem for k-approval, when $k \geq 3$, is NP-complete even when all the bribery costs are equal [6].

4. FUTURE WORK

We are studying the open question left in our result table. We also plan to study non-binary domains, other scoring and voting rules, additional bribery actions that can also add dependencies, and the combination of weights with other voting rules.

5. ACKNOWLEDGMENTS

Nicholas Mattei was supported by NSF CCF-1049360 and NSF IIS-1107011, which, as part of the IJCAI-11 Doc. Consortium, funded his visit to Padova. This work has been partially supported by the MIUR PRIN 20089M932N project, "Innovative and multi-disciplinary approaches for constraint and preference reasoning."

6. REFERENCES

[1] K. J. Arrow, A. K. Sen, and K. Suzumura. *Handbook of Social Choice and Welfare.* North-Holland, Elsevier, 2002.

[2] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *JAIR*, 21(1):135–191, 2004.

[3] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. *SOFSEM 2007*, pages 51–69, 2007.

[4] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[5] P. Faliszewski. Nonuniform bribery. In *AAMAS08*, pages 1569–1572, 2008.

[6] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *JAIR*, 35:485–532, 2009.

[7] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.

On the benefits of argumentation schemes in deliberative dialogue

(Extended Abstract)

Alice Toniolo
University of Aberdeen
King's College
Aberdeen, UK
a.toniolo@abdn.ac.uk

Timothy J. Norman
University of Aberdeen
King's College
Aberdeen, UK
t.j.norman@abdn.ac.uk

Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
katia@cs.cmu.edu

ABSTRACT

We present a model of argumentation-based deliberative dialogue for decision making in a team of agents. The model captures conflicts among agents' plans due to scheduling and causality constraints, and conflicts between actions, goals and norms. We evaluate this model in complex collaborative planning problems to assess its ability to resolve such conflicts. We show that a model grounded on appropriate argumentation schemes facilitates the sharing of relevant information about plan, goal and norm conflicts. Our results show also that this information-sharing leads to more effective conflict resolution.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-Agent Systems

General Terms

Experimentation

Keywords

Argumentation, Collective decision making, Planning

1. INTRODUCTION

Collaborative decision making among a team of agents is a complex activity, particularly when agents have different but interdependent objectives. When agents need to collaborate to accomplish a task, they must form an agreement on a plan to enact and coordinate together [3]. Agents may, however, have conflicting opinions on what to include in a shared plan due to differing commitments.

Argumentation-based models of dialogue enable agents to provide justifications for differing positions regarding a joint problem, which is useful in complex collaborative situations. Using such an approach, agents can identify conflicts in joint plans, and explore and identify alternative solutions that are more favourable for the team. Atkinson and Bench-Capon [1] present an argumentation-based dialogue for practical reasoning based on argumentation schemes focussed on finding a common plan for a joint goal. In related research, Toniolo et al. [4] present a formalisation of argumentation schemes appropriate for agents engaging in deliberative dialogue where conflicts may also arise due to differing objectives and normative constraints. A study of the benefits of using argumentation schemes for agents negotiating the allocation of limited resources is presented in Karunatillake et al. [2]. In existing research, however, the utility of using argumentation schemes is demonstrated through extended examples where the possible solutions are pre-established. How information shared during dialogue influences conflict resolution has not been rigorously assessed.

In this paper, we consider complex problems of collaboration where agents have differing objectives, norms and plan constraints. We empirically evaluate the model of argumentation schemes presented in [4] within symmetric and asymmetric dialogue systems and present some evidence for how such a model facilitates the identification and resolution of conflicts between interdependent plans.

2. DIALOGUE SYSTEM

We consider agents that prepare plans to achieve individual objectives, but must collaborate to coordinate interdependent tasks or to inform the team about critical commitments. The dialogue system includes a language for discussing agents' plans, a model of arguments, a set of defeasible relations among arguments, and a dialogue protocol. The planning language is based on *situation calculus* extended for norms that define what an agent is obliged or forbidden to do. The structure of the arguments is grounded upon the argumentation schemes for practical reasoning presented in [4]. An action, described through its preconditions, effects and the goal that it contributes to, may be proposed by one of the parties. Agents can, then, formulate arguments that deal with potential conflicts between the proposed action and other actions, norms and goals through the use of critical questions including "**CQ1:** Is the action possible given other concurrent actions in the plan?", "**CQ2:** Is the action possible according to causal plan constraints?", "**CQ3:** Is there any conflicting norm that regulates actions or states of the world?", "**CQ4:** Is the goal justified?". The model defines an argumentation scheme for each of these conflicts and the critical questions identify defeat and support relations among arguments. A support relation justifies an agent's commitment, and a defeat relation describes

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

a conflict between a task of an agent and a task, a norm or a goal of the opponent's plan.

The dialogue system is built for two-agents discussions. Initially each agent creates an individual plan that is locally norm-consistent. The proponent starts the dialogue proposing an action from its individual plan to the other agent, and the dialogue progresses in a turn-taking fashion. When an agent passes, the proponent withdraws its proposal or the opponent accepts it and the dialogue terminates. On termination, agents may re-plan taking into account new information acquired during the dialogue. This, we claim, will lead agents to identify better collaborative plans. To test this claim, we consider three protocols for communication that correspond to different degrees of freedom in moving arguments. Protocol $\mathcal{P}_{ctrl}$ is a control condition where agents are not permitted to exchange arguments other than accepting or rejecting the claim. The argumentation-based protocols are symmetric protocol ($\mathcal{P}_{sym}$) where proponent and opponent may use defeat or support relations to form arguments, and asymmetric protocol ($\mathcal{P}_{asym}$) where the opponent explores its objections to the proposed action which are defended by the proponent.

3. EVALUATION

Design. The metric for evaluation is the feasibility of the resulting plans; i.e. the number of conflicts of different types between individual plans that can hamper execution of interdependent tasks. The agents' planning domain concerns operations of a local authority and a humanitarian organisation for evacuating people following a disaster. We ran 450 experiments for each protocol, starting from randomly generated initial plans. The conflicts were analysed before discussion to measure the total number of conflicts among the two plans (complexity of the problem) and post discussion for the conflicts solved.

Results. Figure 1.A shows that the number of conflicts solved is higher when argumentation schemes are used to guide the dialogue ($\mathcal{P}_{sym}$ and $\mathcal{P}_{asym}$) than in the control condition ($\mathcal{P}_{crtl}$). We plot here the percentage of solved conflicts as the complexity of the problem increases. In protocols $\mathcal{P}_{asym}$ and $\mathcal{P}_{sym}$ the conflict resolution trend stabilises (at around 33% and 45% respectively) showing an approximately linear relation between solved conflicts and plan complexity. In the control condition the trend falls, demonstrating that agents solve fewer conflicts as plan complexity increases. This result provides evidence for the claim that many conflicts can only be discovered through the exchange of arguments and, hence, sharing relevant information about existing plan, norm and goal commitments.

Although $\mathcal{P}_{asym}$ and $\mathcal{P}_{sym}$ show a similar trend, there is a difference in the performance of the two protocols. The total number of arguments exchanged in $\mathcal{P}_{sym}$ tends to be higher than with $\mathcal{P}_{asym}$ (Figure 1.B). However, the proportion of conflicts resolved with $\mathcal{P}_{sym}$ is more than 10% higher than with $\mathcal{P}_{asym}$ (Figure 1.A). The difference here is that $\mathcal{P}_{sym}$ permits additional information to be exchanged; i.e. justifications for an agent's commitment as well as identification of conflicts. This result provides evidence for there being a tradeoff in practice between the complexity of the dialogue and the number of conflicts that can be solved. We conclude that, although $\mathcal{P}_{sym}$ leads to more complex dialogues, it is more effective in resolving more complex interdependencies between agents' plans.

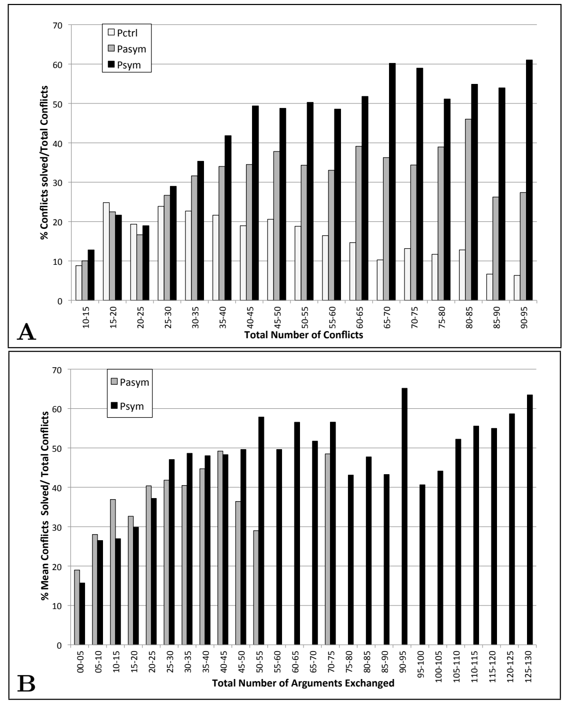


Figure 1: Results.

4. CONCLUSIONS

In this research we have considered problems where collaboration among agents is hampered by a wide number of conflicts related to individual objectives, norms and plan constraints. We have evaluated an argumentation-based model of deliberative dialogue grounded upon argumentation schemes that identify the causes of conflicts in collaborative planning. Our study has shown that the use of argumentation schemes leads to an effective exchange of relevant information. We have also demonstrated that focussed information sharing supports agents in creating more favourable collaborative plans.

5. ACKNOWLEDGMENTS

This research is supported by the award made by the RCUK Digital Economy programme to the dot.rural Digital Economy Hub; award reference: EP/G066051/1.

6. REFERENCES

[1] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10-15):855–874, 2007.

[2] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and P. McBurney. Dialogue games that agents play within a society. *Artificial Intelligence*, 173(9-10):935–981, 2009.

[3] Y. Tang, T. J. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In *Proceedings of the 8th International Conference on Autonoumous Agents and Multiagent Systems*, 2009.

[4] A. Toniolo, T. J. Norman, and K. Sycara. Argumentation schemes for collaborative planning. In *Proceedings of the 14th International Conference on Principles and Practice of Multi-Agent Systems*, 2011.

Testing the Benfits of Structured Argumentation in Multi-Agent Deliberation Dialogues

(Extended Abstract)

Eric M. Kok, John-Jules Ch. Meyer, Henry Prakken and Gerard A. W. Vreeswijk
Department of Information and Computing Sciences
Utrecht University, The Netherlands
{erickok,jj,henry,gv}@cs.uu.nl

ABSTRACT

Work on argumentation-based dialogue systems often assumes that the adoption of argumentation leads to improved efficiency and effectiveness. Several studies have taken an experimental approach to prove these alleged benefits, but none has so far supported the expressiveness of a logic for structured argumentation. This paper shows how the use of argumentation in deliberation dialogues can be tested while supporting goal-based agents that use the ASPIC framework for structured argumentation.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*

General Terms

Experimentation, Design

Keywords

Argumentation, Multi-agent dialogues, Experimental evaluation

1. INTRODUCTION

To improve communication and shared decision making in multi-agent systems it is often proposed to allow for argumentation in inter-agent dialogues. Throughout the years many frameworks and protocols have been developed and the theoretical reachability of ideal and intuitive outcomes has often been proved formally. However, since not all properties can be studied formally at least three works have experimentally explored the benefits of argumentation in dialogues. [2, 5, 1] On the other hand, none of these studies have captured the expressivity of formal models of argument based inference. They particularly lack a language in which arguments with internal structure can be used to cover realistic argumentation dialogues.

2. DELIBERATION MODEL

This paper tests the benefits of argumentation in multi-agent deliberation dialogues. Agents aim to reach agreement on a course of action, while considering a mutual goal. This type of dialogue is of particular interest because of the mix of competitive and cooperative elements. A slightly simplified version of the framework for deliberation dialogues of Kok et al. [3] is used, which models dialogues as a series of moves in which proposals can be made and questioned and where arguments can be stated, constructed using options, goals and beliefs. Figure 1 shows an example of a proposal tree. By making proposals and replying to these the agents influence the dialogical status of the moves and ultimately of the dialogue outcome.

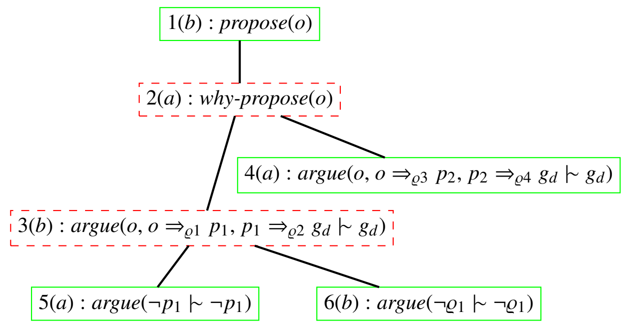


Figure 1: Example of a proposal tree for an option o

Arguments are formed using a simple instantiation of the abstract ASPIC framework for argumentation with structured arguments [6], which is an instance of the Dung abstract argumentation model. An argument can be attacked by rebutting a conclusion of a defeasible inference, by undermining one of its premises or by undercutting one of its defeasible inferences.

3. SCENARIO GENERATION

In the experiment of this paper, agents engage in a dialogue according to a scenario, which represents the underlying deliberation problem. It describes the mutual goal and the beliefs, goals and options known to the agents. Consequently, the structure of a scenario heavily influences the dialogue and the outcome. It is therefore important that scenarios reflect the characteristics of real deliberation problems.

The process to generate these scenarios is explained in detail in [4]. The idea is that agents are assigned beliefs, goals and options in a systematic way through three steps. First, roles are specified from which most of an agent's goals and known options originate. Second, chains of inference rules are generated between an agent's goals and known options. One chain ensures that an agent can potentially form arguments for its options so it can propose and de-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

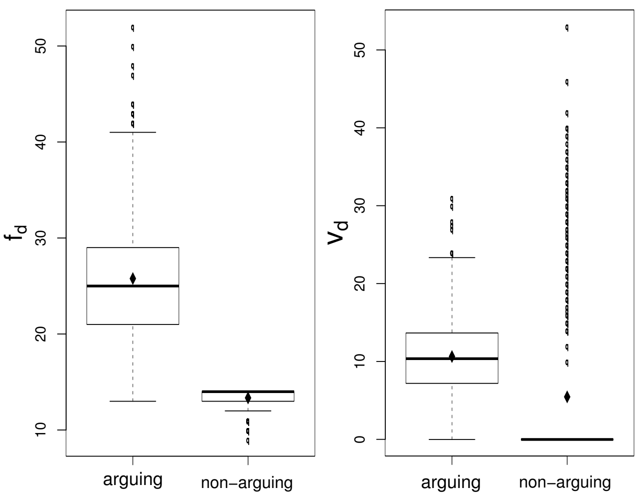


Figure 2: Efficiency f_d and effectiveness v_d of the arguing and non-arguing strategies, with averages ◆

fend them in the dialogue. Third, a set of potential conflicts is generated for every rule chain. These are negated facts that allow agents to undercut, undermine or rebut arguments. Finally, a subset of these generated options, goals and beliefs is assigned to the agent, along with some personal additional goals and beliefs. The resulting scenarios provide a structure that results in interesting dialogues with potential for arguments and counter-arguments.

4. EXPERIMENTAL EVALUATION

To test the benefits of argumentation in dialogues an experiment was conducted in which arguing and non-arguing strategies are tested. In both strategies the agents evaluate their known options based on the utility of defensible goals, but the non-arguing strategy will solely propose options and not play arguments or counter arguments. Since scenarios are generated in a structural fashion, the arguing strategy is likely to be able to form arguments for and against dialogue proposals.

Dialogical effects are tested in terms of dialogue *efficiency* by counting the played number of moves, and dialogue *effectiveness* by measuring the shared utility of the dialogue outcome. Arguing and non-arguing agents engage in a dialogue given a generated scenario and the effectiveness and efficiency is tested at termination. In the final experiment the average efficiency and effectiveness of both strategies were compared over 1000 scenarios.

In Figure 2 the average efficiency f_d, the number of dialogue moves, of the arguing and non-arguing strategies is shown on the left. Clearly the average number of moves when arguing ($f_d \approx 26$) is much higher than when the agents do not argue ($f_d \approx 14$). This is simply because all the non-arguing agents do is propose or reject options, while the arguing agents actually discuss claims. While argumentation may possibly prevent unnecessary moves, improving efficiency, this is clearly not true for this model of this paper.

In Figure 2 the effectiveness v_d, the total utility the agents have for the dialogue outcome, is shown on the right. Clearly, the average effectiveness is much higher ($v_d \approx 10$) for the arguing strategy than for the non-arguing strategy ($v_d \approx 5$). In many dialogues the non-arguing agents reject all proposals, leaving no dialogue outcome and hence a utility of 0. Because the arguing agents can move arguments giving a motivation, they can defend proposals, making them available to select as dialogue outcome again.

Finally, a comparison was made between the arguing strategy and a baseline strategy that never evaluates and rejects options, but proposes all the options known to the agent. Since there is then no selection over preferred outcomes, the strategy was expected to result in a lower effectiveness, that is, a lower shared utility. However, it was found that the average effectiveness betweent he baseline and arguing strategies was very similar. It might seem then that arguing in deliberation dialogues might not be beneficial after all, but there is still a difference in the way the results came to be. The arguing strategy empowers rational and self-interested agents and the dialogues they produce contain useful information like which proposals were clearly not the right choice. Further research is needed to investigate how this additional information can best be utilized.

5. CONCLUSIONS

Existing work on the experimental evaluation of the benefits of argumentation in agent dialogues makes use of very simple models of argumentation, in which arguments have no or very little structure. This paper has improved the state-of-the art by carrying out an experimental evaluation of deliberation with arguments that have considerably more structure and can be attacked by undercutters, underminers or rebuttals. Our work partly confirms findings of earlier work [2, 5, 1] that the use of argumentation in inter-agent dialogues may be beneficial to the agents.

A second contribution of our paper is a methodology for carrying out evaluation experiments using inter-agent dialogues with structured arguments. Since this kind of research is still rare, a new method needed to be developed, based on the generation process for realistic scenarios (presented in further detail in [4]) and a strategy model for goal-directed agents, with the aim to support future experimental research.

6. ACKNOWLEDGMENTS

This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 612.066.823.

7. REFERENCES

[1] E. Black and K. Bentley. An empirical study of a deliberation dialogue system. In *Proceedings of the 1st International Workshop on the Theory and Applications of Formal Argumentation*, Barcelona, Spain, 2011.

[2] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and P. McBurney. Dialogue games that agents play within a society. *Artificial Intelligence*, 173(9-10):935–981, 2009.

[3] E. M. Kok, J.-J. C. Meyer, H. Prakken, and G. A. W. Vreeswijk. A Formal Argumentation Framework for Deliberation Dialogues. In P. McBurney, S. Parsons, and I. Rahwan, editors, *Proceedings of the 7th International Workshop on Argumentation in Multi-Agent Systems*, Berlin, Germany, 2010. Springer-Verlag.

[4] E. M. Kok, J.-J. C. Meyer, H. van Oostendorp, H. Prakken, and G. A. W. Vreeswijk. A Methodology for the Generation of Multi-Agent Argumentation Dialogue Scenarios. In *9th European Workshop on Multi-agent Systems*, Maastricht, The Netherlands, 2011.

[5] P. Pasquier, R. Hollands, I. Rahwan, F. Dignum, and L. Sonenberg. An empirical study of interest-based negotiation. *Autonomous Agents and Multi-Agent Systems*, 22(2):249–288, 2010.

[6] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

Knowing Each Other in Argumentation-based Negotiation

(Extended Abstract)

Elise Bonzon
LIPADE
Paris Descartes University
France
elise.bonzon@parisdescartes.fr

Yannis Dimopoulos
Dep. of Computer science
University of Cyprus
Cyprus
yannis@cs.ucy.ac.cy

Pavlos Moraitis
LIPADE
Paris Descartes University
France
pavlos@mi.parisdescartes.fr

ABSTRACT

Argumentation-based negotiation has emerged as an important topic in multi-agent systems over the last years. Although there are many studies of frameworks that enable agents to negotiate through the exchange of arguments, there is a lack of reasoning methods that employ the (usually incomplete) knowledge an agent may have about his opponent. This work addresses this issue by providing a reasoning mechanism that allows negotiating agents to take into account information about their counterparts. Thus an agent may support his own decisions by using arguments that are meaningful for his opponent. Experimental results highlight the impact of the proposed approach in the negotiation process.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Argumentation, Negotiation, Collective decision making

1. INTRODUCTION

Over the last years argumentation-based negotiation (ABN) has gained an increasing interest in the multi-agent field (see e.g. [2]). Two important underlying hypotheses shared by all works in ABN are (a) the selection of arguments that an agent uses to justify his offer to his opponent or to attack or defend another argument, is based solely on his knowledge about the world and his self-interest (b) the knowledge that an agent has about his opponent comes exclusively from their interaction during the negotiation.

The above assumptions seem rather counterintuitive. Consider, for instance, a simple scenario where a car salesperson negotiates with a rich potential buyer over the purchase of a car. Driven by his self-interest to maximize profit, the salesperson suggests the new top of the range Ferrari model. However, the arguments that he will possibly use to justify the offer to the customer, are quite different than the high profit this sale carries, and would rather argue about the very strong motor, the exceptional handling, etc. Moreover, a competent salesperson is expected to use arguments that are appropriate for the customer, even without any prior interaction between them.

This work presents a new perspective to ABN that captures these intuitions in an argumentation based reasoning mechanism for negotiation, where agents use both the knowledge they have about the world (as in the existing works) as well as the (usually incomplete) knowledge they have about the other agents in order to make the crucial decisions at any time. More precisely this new perspective considers that agents use their own arguments for choosing the offers to propose but, whenever possible, use arguments that are *meaningful* for their opponents to support those offers. This policy is also applied for the arguments that agents use for attacking the opponent's arguments.

This paper provides a brief, high-level, description of the new ABN reasoning mechanism, along with a selection of experimental results that confirm what one might intuitively expect: knowledge on the opponent may have a positive impact on the length of the negotiation as well as the quality of the obtained solutions.

2. THE NEGOTIATION MECHANISM

The negotiation framework of this work is the one of [3]. We assume two agents, α and β, who are involved in a bilateral negotiation over a set of offers (options) $\mathcal{O} = \{o_1, ..., o_n\}$ which are identified from a logical language $\mathcal{L}$. As in [3] it is assumed that an agent α has a *theory* represented in an abstract way, that consists of a set of arguments; a function that returns the arguments which support a given offer, and a defeat relation between arguments. This defeat relation is computed by combining a conflict relation between arguments and a preference relation on the set of arguments.

Moreover, we assume that each agent has also knowledge about the other agent he could negotiate with. The theory agent α has on β has the same structure as the agent's α own theory, but we suppose it to be incomplete, as the knowledge α has on β is partial. The important part of this theory is the set of arguments agent α knows. This set can be empty if α does not know anything about β, or contains a subset of β's arguments. We must note that the knowledge an agent has about his opponent is incomplete but accurate (i.e. as far as arguments, preferences on these arguments and conflicts).

In [4], Rubinstein introduced the *Alternating Offers pro-*

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

tocol for bargaining between agents. This protocol has been adapted in the argumentation-based negotiation context in [3]. In this work we adapt the *negotiation strategy* that is used in [3] by considering the case where agents have some partial knowledge about their opponent.

The new reasoning mechanism we have implemented realizes this new strategy which corresponds to the main idea proposed in this paper. According to this idea, agents use their own negotiation theory in order to find the best offer to propose to their opponent. This offer is supported by the current "strongest" acceptable (wrt the defeat relation) argument in the agents' theory. Then they use the partial knowledge they have on their opponent in order to find whether this offer is supported by an acceptable argument in the opponent's argumentation theory. If this is the case, this argument is sent for supporting the proposed offer. Otherwise they are looking whether there exists an argument that supports this offer in the opponent's theory, that is not currently acceptable, but which could be defended by their own theories in order to become an acceptable one. The same policy is also applied for choosing the arguments that are used for attacking the arguments of the opponent. However, if such arguments do not exist, agents use the arguments of their own theories for supporting or defending an offer as it is done in the frameworks where agents have no knowledge on the opponent.

3. EXPERIMENTAL EVALUATION

The experimental evaluation is based on two systems. The first implements the method of [3], that does not utilize any form of knowledge about the opponent agent, whereas the second system is an implementation of our approach.

Agent theories have been generated randomly, as sizeable real-life argumentation theories are not readily available. Random theory generation also facilitates the process of creating structurally diverse theories. Indeed, the experimental suite used in this work includes a variety of agent theories with up to 230 arguments, that differ regarding the relation between the preferences on the epistemic arguments of the negotiating agents, as well as the knowledge an agent possesses about his opponent. The experimental suite contains test cases that are generated by assigning values to two parameters. The first parameter concerns the percentage of common preferences between epistemic arguments ($\succeq_e$) shared by the agents, with values 100% and 50%. The second parameter concerns the portion of the knowledge (i.e. arguments) each agent has on his opponent ($\mathcal{A}^{\alpha,\beta}$), with values 0%, 25%, 50% and 100%.

In the following, R_K denotes the round where an agreement is found by using our system (agents have some knowledge K about each other) and $R_{\neg K}$ the round where an agreement is found with the system of [3] (without knowledge about each other). D_K (resp. $D_{\neg K}$) is the distance between the outcome of the negotiation found with our system (resp. with the system of [3]) and the optimal (or ideal) solution for each agent (see [1]). Then, nR_K is the number of negotiations where our system found an agreement in less rounds than the system of [3]; $nR_{\neg K}$ is the number of negotiations where the system of [3] found an agreement in less rounds than our system; nD_K denotes the number of negotiations where the distance of the outcome of the negotiation from the optimal solution is smaller for at least one agent and not worse for the other agent in our system than in the one of [3]; $nD_{\neg K}$ is the number of negotiations where the distance of the outcome of the negotiation from the optimal solution is smaller for at least one agent and not worse for the other agent in [3] than in our system. Table 1 presents the comparative results for the experiments where *both systems have found an agreement* (the number of such negotiations over the 180 experimented per test is given in column $nAgr$). Each test (row) consists of 180 negotiations. The number of arguments involved is between 60 and 230 for each agent's theory.

Table 1: Comparison of the systems

	nR_K	$nR_{\neg K}$	$nAgr$	nD_K	$nD_{\neg K}$
$\succeq_e$: 100%, $\mathcal{A}^{\alpha,\beta}$: 100%	45	0	152	5	1
$\succeq_e$: 100%, $\mathcal{A}^{\alpha,\beta}$: 50%	20	2	152	0	0
$\succeq_e$: 100%, $\mathcal{A}^{\alpha,\beta}$: 25%	0	0	152	0	0
$\succeq_e$: 100%, $\mathcal{A}^{\alpha,\beta}$: 0%	0	0	152	0	0
$\succeq_e$: 50%, $\mathcal{A}^{\alpha,\beta}$: 100%	47	1	141	3	2
$\succeq_e$: 50%, $\mathcal{A}^{\alpha,\beta}$: 50%	4	2	141	1	0
$\succeq_e$: 50%, $\mathcal{A}^{\alpha,\beta}$: 25%	0	0	141	0	0
$\succeq_e$: 50%, $\mathcal{A}^{\alpha,\beta}$: 0%	0	0	141	0	0

The analysis of the experimental results summarized in Table 1 gives us useful information about: (1) the usability in practice of argumentation based negotiation and the way it computationally behaves while scaling in a bilateral negotiation context (2) the performance of our approach. Concerning the first point, our work is (as far as we know) the first one to empirically show that a Dung-based abstract preference-based argumentation framework behaves computationally well while scaling in a bilateral negotiation context. We ran 1440 negotiation experiments which, when resulted in agreement, did so in reasonable execution times. More precisely the average time for an agrement was between 10s and 15s for a size of 60 arguments for each agent theory and 45s for a size of 230 arguments for each agent theory. Concerning the second point, the results show that our system improves the performance of the system of [3] regarding two important criteria, namely the *length of the negotiation* when there is an agreement and the *quality* of the agreement. More precisely concerning the criterion of length, the use of knowledge about the other agent has, (no matter what the % of knowledge about the other agent is), a significant positive impact on the negotiation shortening. This can be important especially for time constraint negotiations. Finally it is worth noting that both systems find exactly the same solution and in the same round when in our system there is no knowledge at all on the opponent.

4. ACKNOWLEDGMENTS

We would like to thank Sonia Akroune for her great work on the experimental part.

5. REFERENCES

[1] L. Amgoud and S. Vesic. A formal analysis of the role of argumentation in negotiation dialogues. *JLC*, 2011.

[2] Y. Dimopoulos and P. Moraitis. Advances in argumentation based negotiation. *Chapter 4, Negotiation and Argumentation in Multi-Agent Systems*, 2011.

[3] N. Hadidi, Y. Dimopoulos, and P. Moraitis. Argumentative alternating offers. In *AAMAS'10*, pages 441–448, 2010.

[4] A. Rubinstein. Perfect Equilibrium in a Bargaining Model. *Econometrica*, 50:97–110, 1982.

Branch and Bound for Negotiations in Large Agreement Spaces

(Extended Abstract)

Dave de Jonge
Artificial Intelligence Research Institute,
IIIA-CSIC
Campus de la UAB
08193, Bellaterra, Catalonia, Spain
davedejonge@iiia.csic.es

Carles Sierra
Artificial Intelligence Research Institute,
IIIA-CSIC
Campus de la UAB
08193, Bellaterra, Catalonia, Spain
sierra@iiia.csic.es

ABSTRACT

We introduce a new multiagent negotiation algorithm for large and complex domains, called NB^3. It applies Branch & Bound to search for good offers to propose. To analyze its performance we present a new problem called the Negotiating Salesmen Problem. We have conducted some experiments with NB^3 from which we conclude that it manages to decrease the traveling cost of the agents significantly, that it outperforms random search and that it scales well with the complexity of the problem.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Negotiation, Search, Negotiating Salesmen Problem

1. INTRODUCTION

Previously proposed negotiation algorithms have mostly focused on the utility space. They assume that given a utility aspiration level it is always possible to find a proposal that would fit that level. In this paper we focus on complex problems for which these classical continuity assumptions do not apply and thus solutions have to be found directly at domain level. Also, we address a number of realistic assumptions that make the application of current negotiation algorithms unfeasible: the space of solutions is huge, utility is non-linear and therefore difficult to calculate, the environment is only partially observable, decisions have to be made within a limited time frame and solutions may involve many agents, possibly human.

We introduce a new family of Branch and Bound algorithms, namely NB^3 (Negotiation Based Branch & Bound), that use negotiation as the key element in the exploration of the joint space of solutions for multiple agents.

As far as we know there are no algorithms implemented that are comparable with the work we present here. Most work on negotiations involves very simple scenarios with only two agents, a small space of solutions and linearly additive utility functions. Although non-linear utility has been studied in for example [1] they still assume that the utility is simple to calculate. The combination of search and negotiation has been studied in [2], but they assume a mediator that must be trusted and severely limits the freedom of the agents, making it more useful for cooperative scenarios rather than competitive ones. In [3] there is also search, but once again with simple utility functions. In [4] it was suggested to use genetic algorithms to explore the agreement space, but no implementation or results were given.

2. NB^3 BASIC CONCEPT

We assume a multiagent scenario in which each agent has a personal cost function to minimize. Each agent has a set of possible actions that it can execute in order to change the state of the world into a new state for which its personal cost function has a lower value. The result of an action depends on the actions executed by the other agents, so they have to negotiate over which joint plan of actions to execute. Their interests are however conflicting: a certain world state might yield low costs for one agent, but high costs for another agent. The agents are assumed to be selfish: they are only interested in minimizing their personal cost function. This means that the agents must compromise: each agent should propose plans that lower his own cost as much as possible, but that at the same time lower the costs of the other agents sufficiently to make them accept these plans.

We designed an algorithm, which we call NB^3, to run on such an agent. The other agents present might also run this algorithm, or any other negotiation algorithm, or they might be human, but that is irrelevant to us. NB^3 applies a Branch and Bound tree search to explore the space of all possible plans under the above mentioned assumptions. Each node in the tree represents a partial plan, and maintains a lower- and upper-bound for the cost function of the agent as well as an estimation of the lower- and upper-bounds of the costs of all the other agents. Whenever for a certain node the lower bound of one of the agents is higher than the reservation value of that agent, that node can be pruned, because it

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

means that this partial plan can never yield a lower cost for the agent than its reservation value. In order to determine which nodes to expand, NB^3 uses a heuristic that is based on the offers that the other agents have made previously. In this way, the search can be directed towards a solution that is acceptable to all agents. For more details we refer to [5].

3. NEGOTIATING SALESMEN PROBLEM

To test the algorithm we have defined a new problem, called the Negotiating Salesmen Problem (NSP). It is a variant of the Traveling Salesman Problem, but with multiple salesmen, each only interested in minimizing its own path.

The idea is that there is a set of cities and a set of salesmen and each city needs to be visited by at least one agent. There is one home city where each agent should start and finish its trajectory. Every other city is assigned to one salesman that has to visit it. However, the salesmen are allowed to exchange their cities amongst one another, so that the agents can decrease the distances they have to travel. For example: if a city v is assigned to agent α, but α prefers to visit another city v', which is assigned to agent β, then α will propose β to exchange v for v'. If β however doesn't want v he will not accept this deal. And if no other agent wants to accept v either, then α is obliged to travel along city v. However, we impose the restriction that some cities cannot be exchanged. The cities that can be exchanged are called the *interchangeable cities*, while the cities that cannot be exchanged are the *fixed cities*. Each agent therefore prefers to visit cities that are close to any of his fixed cities.

4. EXPERIMENTAL RESULTS

We have implemented an agent that applies NB^3 to the NSP and conducted a number of experiments with this implementation. Ideally, we should test our algorithm against other negotiation algorithms but, as mentioned, we don't know of any such algorithm that could handle the hard conditions we are considering.

Note that comparing the algorithm against existing search algorithms will not work, since they do not apply negotiation. Pure search algorithms might find the most selfish solution, or the socially optimal solution, but are not able to find the best compromise between these two extremes, given the offers made by other agents. Moreover we do not claim that our search algorithm is better than any existing search algorithm, we only claim that we have made the first algorithm that successfully combines search and negotiation in large and complex agreement spaces.

In order to do useful experiments anyway, we have tested the algorithm against a simplified version of itself that expands the search tree randomly, i.e. without using smart heuristics. Also we have done some tests in which all agents were running NB^3 and repeated these tests with different problem sizes to see how the algorithm scales. Finally we have compared the results with the socially optimal solution.

For any agent we determined a score by comparing its path length after negotiations with its path length before the negotiations. So if an agent scores for example 40% it means that its final path length was 40% shorter than its initial path length. The scores presented here are each averaged over all agents and 25 problem instances.

From the results we can conclude that our algorithm significantly outperforms random search. In the NB^3 vs. random search experiments the NB^3 agents were able to decrease their path lengths by 30%, while the random search agents did not score higher than 10% to 20% depending on the number of NB^3 agents present.

From the experiments with varying complexity of the NSP-instances we conclude that NB^3 scales very well with increasing complexity. When increasing the number of agents from 6 to 16 while holding the number of interchangeable cities per agent fixed at 10, the results stay stable between 25% and 30%. Increasing the number of cities per agent from 6 till 16 while holding the number of agents fixed at 10, the average score decreased from 38% to around 25%.

It is impossible for all the agents to decrease their path lengths with 100% because that would mean they are not traveling at all anymore. Therefore we also compared the results with the length of the paths of the socially optimal solution, and it turned out that the agents are able to decrease their costs by 65% relative to the social optimum (so 0% indicates no decrease of path length, while 100% indicates the agents have reached the social optimum). Note however that NB^3 was designed for selfish agents, and not for agents that want to reach the social optimum. Therefore, even if the social optimum is found by some of the agents, they might not propose it because they might try to reach more selfish solutions.

5. CONCLUSIONS

We have developed a new algorithm that successfully combines search with negotiation under hard, realistic conditions. We applied it to the Negotiating Salesmen Problem and conclude that its results are significantly better than random search and scale well with increasing problem size.

6. ACKNOWLEDGMENTS

Supported by the Agreement Technologies CONSOLIDER project, contract CSD2007-0022 and INGENIO 2010 and CHIST-ERA project ACE and the Spanish Ministry of Education and Science TIN2010-16306 project CBIT.

7. REFERENCES

[1] Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, and Enrique de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *AAMAS '09*, pages 1057–1064, 2009.

[2] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *Intelligent Systems, IEEE*, 18(6):32 – 38, nov-dec 2003.

[3] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make negotiation trade-offs. pages 119–126, 2000.

[4] Angela Fabregues and Carles Sierra. An agent architecture for simultaneous bilateral negotiations. In *Proceedings of the 13è Congrés Internacional de l'Associació Catalana d'Intel·ligència Artificial (CCIA 2010)*, pages 29–38, Espluga de Francolí, Tarragona, October 2010.

[5] Dave de Jonge and Carles Sierra. Branch and bound for negotiations in large agreement spaces. Technical report, IIIA-CSIC, Bellaterra (Barcelona), Spain, 2012. `http://www.iiia.csic.es/files/mon_library/TR-IIIA-2012-01.pdf`.

Collaborative Job Processing on a Single Machine – A Multi-Agent Weighted Tardiness Problem

(Extended Abstract)

Fabian Lang
Institute of Computer Science
Helmut Schmidt University
Holstenhofweg 85
Hamburg, Germany
fabian.lang@hsu-hamburg.de

Andreas Fink
Institute of Computer Science
Helmut Schmidt University
Holstenhofweg 85
Hamburg, Germany
andreas.fink@hsu-hamburg.de

ABSTRACT

We present a multi-agent variant of the Single Machine Total Weighted Tardiness Problem with Sequence-Dependent Setup Times. Since, i.a., agents have an incentive to lie, central planning is not feasible and decentralized methods such as automated negotiations are needed. Hereto, we propose and evaluate an iterative quota-based negotiation protocol.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]:
Distributed Artificial Intelligence - Multiagent systems

General Terms

Algorithms, Economics, Experimentation

Keywords

Automated negotiation, Autonomous agents, Machine scheduling, Interorganizational system, Agent coordination

1. PROBLEM DESCRIPTION

In the following, we discuss the problem of single machine scheduling with several self-motivated, non-cooperative agents – with applications such as allocation of processing power, satellite data transmission , or terminal scheduling in harbors – and present a negotiation protocol constituting a coordination mechanism to find beneficial agreements. The *Multi-Agent Single Machine Total Weighted Tardiness Problem with Sequence-Dependent Setup Times* (MA-SMTWTP-SDST) is a scheduling problem where a set of jobs $\mathcal{J} = \{1, ..., j, ..., n\}$ has to be processed by a single machine, which can process only a single job at a time. Each job j is assigned to an agent i of the set $\mathcal{I} = \{1, ..., i, ..., m\}$ by an assignment variable a_j. The agents aim at minimizing their individual total weighted tardiness TWT_i (see (1b)). Each job comprises a processing time $p_j(> 0)$, a weight of relative importance $w_j(> 0)$, and a due date D_j. A setup time $s_{k,j}$ occurs between a job j and its preceding job k (see (3)) and c_j denotes the completion time of job j. If a job is not finished before its due date, a tardiness T_j arises (see (2)). The objective of the problem is to find a job sequence $\pi = \{\pi_1, \ldots, \pi_n\}$ (with π_j as processing position) minimizing the collective total weighted tardiness TWT (see (1a)). Lawler [4] shows that the centralized problem (without SDST) is already strongly $\mathcal{NP}$-hard. Since the agents have an incentive to lie (e.g., by declaring their jobs as more important than they really are), revealed information about due dates as well as job weights are worthless and not utilizable. Hence, centralized planning is not feasible here.

$$\min \quad \sum_{i=1}^{m} TWT_i \tag{1a}$$

$$\min \quad TWT_i = \sum_{j \in \mathcal{J} | a_j = i} w_j T_j \ , \ \forall i \in \mathcal{I} \tag{1b}$$

$$T_j = \max\{c_j - D_j; 0\} \tag{2}$$

$$c_j = \sum_{k \in \mathcal{J} | \pi_k \leq \pi_j} s_{\tilde{k} | (\pi_{\tilde{k}} = \pi_k - 1), k} + p_k \tag{3}$$

2. THE PROPOSED PROTOCOL

Here, we present our proposed negotiation protocol for multi-agent coordination (see algorithm 1). The basic idea is that agents overcome local optima by accepting deteriorating contract proposals [2][3]. The protocol is generic. At first, a mediator generates a random initial contract c_o^* that represent the active contract (=current draft). In every iteration t, an acceptance quota p_t for the set of proposals is determined, which declines over time (first round: p_o; last round: $p_{T-1} \stackrel{!}{=} \frac{1}{L}$). Subsequently, the mediator creates $L-1$ mutations $c_t^{'}$ of the active contract c_t^*. Those mutations and the active contract constitute the set of contract proposals $\mathcal{C}^{'}$. Afterwards, the agents decide whether to accept (=1) or reject (=0) the proposals, but have to accept at least $q_t(= p_t * L)$ contracts. We suppose that they accept the $q_t * L$ best contracts as well as possible improving contracts. The mediator selects one randomly from the overall accepted contracts $\mathcal{C}^c$; if there is none ($\mathcal{C}^c = \emptyset$), the active contract remains for the next iteration. Thereafter, the process starts over and new proposals are generated using the (new) active contract. Finally, after T iterations, the last active contract c_{T-1}^* becomes the final contract c^*.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Algorithm 1 An Iterative, Quota-Based Protocol

$c_0^* \leftarrow GenerateInitialContract()$
for $t = 0, 1, \ldots, T-1$ **do**
  $p_t \leftarrow p_0 * \beta^t$; $q_t \leftarrow p_t * L$
  $\mathcal{C}^c \leftarrow \emptyset$; $\mathcal{C}' \leftarrow \{c_t^*\}$
  for $l = \{1, 2, \ldots, L-1\}$ **do**
    $c_t' \leftarrow Mutate(c_t^*)$
    $\mathcal{C}' \leftarrow \mathcal{C}' \bigcup \{c_t'\}$
  end for
  for all $i \in \mathcal{I}$ **do**
    $\overrightarrow{Z_i} \leftarrow AcceptOrReject(\mathcal{C}', q_t)$
  end for
  for all $c_t' \in \mathcal{C}'$ **do**
    if $\sum_{i \in \mathcal{I}} Z_i[c_t'] = m$ **then**
      $\mathcal{C}^c \leftarrow \mathcal{C}^c \bigcup \{c_t'\}$
    end if
  end for
  if $\mathcal{C}^c = \emptyset$ **then**
    $c_{t+1}^* \leftarrow c_t^*$
  else
    $c_{t+1}^* \leftarrow RandomlySelect(\mathcal{C}^c)$
  end if
end for
$c^* \leftarrow c_{T-1}^*$

3. COMPUTATIONAL RESULTS

For the evaluation, we have used the 120 problem instances of the SMTWTP-SDST benchmark library from [1] and assigned the jobs to multiple agents in sequence. Since there are loose and tight problem sets, we have normalized the results: 100% represents the best found value of a problem set in the respective simulation data set.

There are three adjustable parameters: the number of iterations T, the number of proposals $L-1$, and the initial acceptance ratio p_0. Another decision parameter is the mediator's way of generating proposals such as, firstly, shifting a single item to another position in the sequence and, secondly, swapping two items' positions. Table 1 shows the results of different parameterizations of p_0 and L assuming $m = 5$ agents and 100,000 iterations.

Table 1: Configuration of the Quotas

Mutation	*Shifting*			*Swapping*		
L	10	25	50	10	25	50
$p_0 = 10\%$	429%	259%	246%	288%	159%	157%
$p_0 = 33\%$	156%	151%	148%	126%	116%	111%
$p_0 = 50\%$	138%	125%	134%	121%	112%	112%
$p_0 = 67\%$	129%	127%	127%	127%	112%	112%
$p_0 = 90\%$	134%	123%	127%	127%	113%	116%

As shown, the quota protocol needs a sufficiently high absolute value of accepted contracts to succeed. The results tend to be better if there are more proposals as well as higher demanded quotas so we have chosen $\{L = 25; p_0 = 0.67\}$ for the remainder of the paper. Concerning the mutation method, swapping appears to perform better.

Regarding the number of agents, table 2 shows that the TWT increases with more agents, although the jobs are the very same. We trace this finding to a more difficult coordination process between the agents.

Table 3 shows a comparison between the quota-rule and free decision making for different negotiation lengths. The performance is slightly increasing with a rising number of iterations (but converging) and the quota-rule outperforms free decision making by far.

Table 2: Number of Agents

Agents	2	3	4	5	10	15	20
Tardiness (%)	102	114	119	126	135	141	150

Table 3: Iterations and Comparison

Iterations	10K	50K	100K	500K	1,000K
Quota	122%	110%	105%	108%	106%
No Quota	252%	244%	237%	247%	241%

Finally, we have approximated the Pareto frontier using a multi-objective simulated annealing procedure (MOSA, see [5]) with 5,000 runs. Figure 1 depicts the history of an exemplary negotiation between two agents over one million negotiation rounds as well as the Pareto frontier. The negotiation moves intensively through the contract space. At the end, the negotiation has converged and is moving up and down in the neighborhood of the Pareto frontier. The protocol finds even better results than the centralized MOSA.

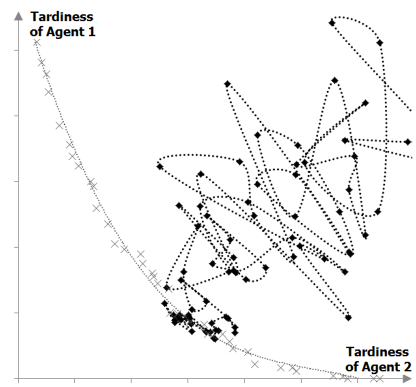


Figure 1: Neg. History & Approx. Pareto Frontier

4. REFERENCES

[1] V. Cicirello. Weighted tardiness scheduling with sequence-dependent setups: a benchmark library. Technical Report, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2003. *http://loki.stockton.edu/∼cicirelv/benchmarks.html*

[2] A. Fink. Supply chain coordination by means of automated negotiations between autonomous agents. In B. Chaib-draa and J. Müller, editors, *Multiagent based Supply Chain Management*, pages 351–372. Springer, Berlin / Heidelberg, Germany, 2006.

[3] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):111–125, 2003.

[4] E. Lawler. A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1:331–342, 1977.

[5] E. L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: a survey. *Journal of Multi-Criteria Decision Analysis*, 3(2):83–104, 1994.

Determining the Willingness to Comply With Norms (Extended Abstract)

N. Criado
Universidad Politécnica de Valencia
Valencia (Spain)
ncriado@dsic.upv.es

E. Argente
Universidad Politécnica de Valencia
Valencia (Spain)
eargente@dsic.upv.es

P. Noriega
IIIA-CSIC
Campus de la UAB, Bellaterra, Catalonia (Spain)
pablo@iiia.csic.es

V. Botti
Universidad Politécnica de Valencia
Valencia (Spain)
vbotti@dsic.upv.es

ABSTRACT

In this paper, we propose that agents make decisions about norm compliance based on three different factors: self-interest, enforcement mechanisms and internalised emotions. Different agent personalities can be defined according to the importance given to each factor.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Legal Aspects, Algorithms, Experimentation

Keywords

Norm Compliance, Norms, BDI Agents

1. INTRODUCTION

Despite the efforts that have been made to develop agents endowed with capabilities for taking into account norms in their decisions, the development of procedures for making autonomous decisions about norm compliance is an important issue that requires more attention [2].

Proposals on normative agent architectures can be mainly classified into *norm-oriented* and *goal-oriented*. The behaviour of norm-oriented agents is completely determined by norms and they do not make decisions about norm compliance. The behaviour of goal-oriented agents is determined by both norms and goals. Up to now the decisions about norm compliance consider the impact of norms and their enforcement mechanisms (i.e., sanctions and rewards) on the agents' goals. Obviously, these reasons are relevant for making decisions about norm compliance. However, there are works on the psychology field [3] that claim that norm compliance is not only explained by rational reasons that consider the impact of norms and their enforcement procedures (sanctions and rewards) on the agent's goals. Besides that, there are emotional reasons, which are related to emotions such as shame, that have not been considered yet in the development of norm-autonomous agents. In this paper we analyse how agents can determine their willingness to comply with norms according to rational and emotional factors.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. DETERMINING THE WILLINGNESS TO COMPLY WITH NORMS

As stated by Conte et al. in [1] "*the decision to comply with a norm is made considering: the value of the violation (probability and weight of punishment), the importance of the goal and feelings related to norm violation*". To calculate this willingness we have mainly considered the works of Elster [3] that analyse factors that sustain norms in human societies. In these works, Elster claims that compliance with norms can be explained by three factors: (i) *self-interest* motivations (f'_w), which consider the influence of norm compliance and violation on agent's goals; (ii) the *expectations* (f''_w) of being rewarded or sanctioned by others; and (iii) *emotional* factors (f'''_w) that are related to internalised emotions such as honour (vs. shame) and hope (vs. fear). The agent's willingness to follow a concrete norm is calculated as a weighted average as follows:

$$\frac{w' \times f'_w + w'' \times f''_w + w''' \times f'''_w}{w' + w'' + w'''}$$

where the weights w', w'' and w''' are defined within the $[0, 1]$ interval.

We have assumed that the weighted average is a suitable method to derive the central tendency of these three functions. The weights that each agent gives to these factors characterise the agent's personality and do not depend on the norm that is considered.

2.1 Self-interest

The *self-interest* factor (f'_w) evaluates the consequences of a given norm from a utilitarian perspective; i.e., the utility is the good to be maximized. The utility of a norm is defined by considering the direct positive or negative consequence of the norm fulfilment. In case of an obligation, the direct consequence of the fulfilment of the obligation is the obliged condition. In case of a prohibition, obeying this prohibition implies that the forbidden condition will be avoided.

2.2 Expectations

The *expectation* factor (f''_w) models the impact of the external enforcement on agents. Specifically, the enforcement mechanism considered in this work consists in a material system of sanctions and rewards that modify the utility that agents obtain when they violate or fulfil norms. This factor considers how much the agent loses from being penalised and how much it gains from being rewarded. The violation of the norm implies that the agent will be sanctioned and not rewarded. Thus, the *expectation* factor is defined as the combination of the undesirability of the sanction and the negation of the reward. For simplicity, we assume that there is a perfect enforcement that always punishes offenders and rewards obedience. However, if agents are able to perceive the probability of being punished or rewarded, then the desirability of sanctions and rewards should be pondered with these probabilities.

2.3 Anticipated Emotions

The *emotional* factor (f'''_w) models the emotions triggered when the agent violates a given norm. We use the term emotion for representing the valued reaction of agents (i.e., the agent's cognitive interpretation) with respect to some aspect of the world (i.e., the reality) [4]. Specifically, agents are capable of anticipating, exhibiting and explaining those human emotions that are involved with the normative decisions. Thereby, the decisions about norm compliance are based on other criteria beyond utility.

As argued by Elster in [3], in humans norms are sustained by the desire to avoid the disapproval of others. Following Elster's proposal, when the violation of norms is greeted with condemnation self-*attribution* emotions (i.e., shame) are triggered on the offender. Moreover, the situations that are predicted to occur when norms are violated may cause *prospect* emotions (i.e., hope and fear) on the offender.

To estimate the value of these two emotions an emotional model susceptible of being implemented in a software agent is required. Specifically, we consider one of the emotional models that have made a deeper impact on the *Multi-Agent System* (MAS) field; the OCC model developed by *Ortony, Clore and Collins* in [4]. Thus, the OCC model has been used for establishing the intensity of the emotions that are involved in the norm-reasoning process as follows:

- *Self-Attribution Emotions.* According to the OCC model, shame is a self-attribution emotion that is elicited by the evaluation of the actions that have been performed by the agent itself. Specifically, the shame that the agent will feel if it violates a given norm is defined by considering the salience of this norm. Therefore, self-attribution emotions only sustain norm obedience.

- *Prospect Emotions.* According to the OCC model, the hope (vs. fear) emotion is triggered when a desirable (vs. undesirable) event is predicted. The fear and hope emotions that may be triggered if a norm is violated are defined by considering the desirability and probability of the consequences of violating and norm.

3. MAIN AGENT TYPES

The decisions about norm compliance are made by considering three willingness factors (i.e., f'_w, f''_w and f'''_w) that are combined as a weighted average. Therefore, different agent personalities can be modelled according to the definition of the weights w', w'' and w'''. The three basic personalities are:

- *Egoist agents* ($w' = 1$, $w'' = 0$ and $w''' = 0$) are the lest prone to comply with norms, since they only consider whether the norm condition favours or hinders their goals.

- *Cautious agents* ($w' = 0$, $w'' = 1$ and $w''' = 0$) are more prone to comply with norms than egoist agents. This can be explained by the fact that cautious agents consider whether either the sanction or the negation of the reward favour their goals.

- *Emotional agents* ($w' = 0$, $w'' = 0$ and $w''' = 1$) are the most willing to obey norms; i.e., they are the most norm-oriented. This is explained by the fact that attribution emotion only sustains norm obedience.

4. CONCLUSIONS

This paper answers a main question that is related to the possibility of developing norm-autonomous agents that consider emotional criteria in their decisions about norm compliance. In response to this issue, this paper describes how agents can consider both their preferences and the norm repercussions when they determine their willingness to comply with norms. The repercussion of norms is not only defined in terms of the utility of norms and the economic cost (vs. benefit) of the sanctions (vs. rewards), but also in terms of the social repercussion of norms (i.e., emotional factors). Specifically, agents are endowed with mechanisms for anticipating the emotions that will be elicited if the norms are transgressed. Moreover, the way in which agents combine rational and emotional factors allow different personalities to be modelled. As future work we plan to evaluate whether the use of agents that consider emotional criteria obtains better results in norm-governed scenarios.

5. ACKNOWLEDGMENTS

This paper was partially funded by the Spanish government under grants CONSOLIDER-INGENIO 2010 CSD2007-00022, TIN2009-13839-C03-01, TIN2008-04446 and by the FPU grant AP-2007-01256 awarded to N. Criado. This research has also been partially funded by the Generalitat de Catalunya under the grant 2009-SGR-1434 and Valencian Prometeo project 2008/051.

6. REFERENCES

[1] R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm acceptance. In *Proc. of ATAL*, pages 99–112, 1999.

[2] N. Criado, E. Argente, and V. Botti. Open Issues for Normative Multi-Agent Systems.

[3] J. Elster. Social norms and economic theory. *Journal of Economic Perspectives*, 3(4):99–117, 1989.

[4] A. Ortony, G. Clore, and A. Collins. *The cognitive structure of emotions.* Cambridge Univ Pr, 1990.

The Dutch eat at 5:30 pm: Shared Strategies for Agent Reasoning

(Extended Abstract)

A. Ghorbani, H.M. Aldewereld, M.V. Dignum
Faculty of Technology, Policy and Management
Delft University of Technology, The Netherlands
a.ghorbani@tudelft.nl

P. Noriega
Artificial Intelligence Research Institute of the Spanish National Scientific Research Council
Barcelona, Spain
pablo@iiia.csic.es

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Languages and structures

General Terms

Language

Keywords

shared strategies, norms, multi-agent systems

1. INTRODUCTION

In the Netherlands, almost all people have dinner around 5:30pm. As a foreigner in that country, it is almost impossible to plan a (working) meeting around this time, which would be a 'normal' time in many other countries. On the other hand, having dinner that early is not an obligation. No one will be offended or would even care if you choose to eat later. This is an example of a *shared strategy*, i.e. an institutional arrangement where different actors have the intention of performing the same task at a certain time or setting [4].

Even though the concept of shared strategy is socially and computationally very instrumental, it has not yet been implemented nor formalized in the MAS literature. First, it determines the general behaviour of the system thus providing expectations that accommodate the behaviour. For example, restaurants should start preparing meals early since there will be many people coming at that time. Second, this notion adds a new dimension to the deontic classical concept where there is no obligation, permission or prohibition, yet a shared behaviour takes place.

In MAS research, shared strategies can be a new way of expressing conventions that cannot easily be fitted into norms as they have no deontic 'flavour' to it. Moreover, shared strategies are an expectation on individual behavior, rather that an individual plan or goal. Shared strategies are also different from collective intentions [2]. A collective intention is a goal shared by everyone in a team. Moreover, members of the team are aware of other agents intention to meet the common goal. For a shared strategy however, while all agents possibly have the same goal, their execution of tasks to fulfil the goal are independent of each other and if one agent does not perform the task, the general goal can still be met. In other words, a shared strategy does not necessarily have to be activated for all the agents every time.

Regarding the benefits of implementing the concept of shared strategies in MAS, in this paper we take inspiration from the Institutional Analysis and Development framework (IAD), an institutional economic framework developed by the Nobel laureate Elinor Ostrom [4]. IAD is an analysis framework for understanding social systems with the purpose of (re)designing social rules (i.e. norms). The ADICO structure, part of the IAD framework, provides a language for institutional statements, such as shared strategies, institutional rules and norms [1].

2. TOWARDS A DEFINITION

According to E. Ostrom, a shared strategy is a social concept that refers to a type of behavioural pattern that is observed by a significant number of individuals although it is, prima facie, neither associated with any deontic modality, nor having a reward or punishment linked to its performance.

Ostrom, in [4], pg. 143, proposes as an example of shared strategy, the rule of "calling back when a telephone conversation is cut". This strategy is a conditional that under objective circumstances triggers an action. It does not explicitly entail an obligation or a prohibition and no reward or punishment ensues. On a closer look, however, it does entail an *expectation*, that, depending on the context in which the interruption took place, may be a strong, possibly asymmetrical and, if not fulfilled may be consequential. Strategy:"When in Rome, do as Romans do", is an ostensible *directive for action* whose —relatively inconsequential— deontic component may guide the adaptive behaviour of foreigners, on one hand, and the leniency of natives towards non-standard behaviour of foreigners, on the other. Strategy, "Dutch eat at 5:30", asserts a *factual regularity* but it

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

also hides a directive for action whose compliance by an individual is indifferent to the rest of the world; nevertheless, it creates expectations that under certain circumstances, may have practical consequences (in Holland, for an individual's eating plans or for the operation of restaurants). These three strategies may be deemed shared strategies only if we make explicit some assumptions about the expectations involved, otherwise they would be examples of *common* and *collective* strategies. Thus, the third strategy would be not a shared strategy but a "common strategy" if we understand it as a prevalent behaviour. However, it becomes a "shared strategy" when we understand it as an expectation of common behaviour; for instance, saying that most people believe that most Dutch eat at 5:30. Finally, the first strategy also fails to be a shared strategy when the two parties expect that both parties should follow the rule, or technically, when there is collective belief.

3. SHARED STRATEGIES IN MAS

The intuition of the formal definition of shared strategy is that each agent expects that under certain conditions, other agents will behave in a certain way. Based on this expectation, we assume that agents can take two approaches to use shared strategies in their planning, referred here as an *optimistic* and a *pessimistic* approach.

In order to discuss the difference between these two approaches, we take as example the shared strategy: drivers will break when there is an obstacle in the road. An optimistic pedestrian agent will assume that all drivers will break when she crosses the road, and therefore will plan to cross the road even if she sees a car approaching. On the other hand, a pessimistic pedestrian will assume that you cannot know which drivers will adhere to the shared strategy, since not all have to follow it, and therefore will plan to stop at the curb when she sees a car approaching.

From an institutional perspective there are two issues worth identifying. The relationships between shared strategies and institutional design and evolution, and the role of shared strategies in agent-based simulation.

Since shared strategies constitute a regularity of the aggregate behaviour, institutional conventions may be designed to promote or to control the consequences of that regularity. The approach is straightforward when the existence of a shared strategy is known in advance and it is likely that its execution affects institutional objectives. In this case, it is reasonable to include specific evaluation mechanisms to monitor the effects of the strategy, and use these to assess transaction costs that would in turn guide the adaptation of the institution to actual performance (see [3]). When the existence of a shared strategy is not known in advance, ordinary performance monitoring does not necessarily identify the behavioural regularity, even when performance indicators might signal a hidden cost. In such case, institutional reaction may be untimely and ineffectual. To contend with such eventuality, one may attempt to foresee undesirable outcomes and, at the risk of overregulation, legislate against them. The opacity of undesirable outcomes, however, may sometimes be appropriately addressed with conventional mechanism-design techniques or by a clever use of modelling and simulation methodologies.

In addition to their value for visualizing the effect of shared strategies on institutional performance. In this context, the modeler deals with the system as a regulated MAS, making a shared strategy a feature of individual agents and harnessing individual actions through institutional conventions of different sorts. The use of shared strategies may be fruitful for some forms of agent-based simulation. One relevant form is to use shared strategies as a salient part of the agents' internal decision models. This way, the designer may study different aspects of normative, motivational and goal-directed attitudes (for example the interplay of norms and strategies in different agent architectures, norm internalization processes, norm emergence, norm compliance vs. conflict resolution approaches, value formation, achievement degrees). Another form of using shared strategies in agent-based simulation is to factor the analysis of aggregate behavior by designing populations partitioned by shared strategies, thus measuring cost and value of interactions within populations with pure and mixed strategies, rational or spontaneous triggering of the shared strategies, etc.

Finally, as Ostrom remarks, the ADICO structured is meant to be for the analysis of institutional evolution. i.e. one type of statement becoming another type (e.g. passing from shared strategy to norm, etc.)

4. CONCLUSION AND FUTURE WORK

In this paper we presented the concept shared strategy as an alternative concept to that of norm in MAS. Based on the work of Ostrom, namely the notion of ADICO institutional statement, we presented an integrated formalism to describe the semantics of norms and shared strategies, based on a temporal epistemic logic.

A shared strategy is a low priority statement leading to action among a group of agents. Since the expectation is *shared*, each agent believes that *most* other agents will perform the action but does not necessarily know who. Therefore, agents don't have expectations for a particular other agent to perform shared strategies because they cannot know whether that particular agent follows the strategy or not, even though as a group, most will. This yields that no deontic type and no sanction can be assigned to a shared strategy.

Shared strategies are a crucial part of agent societies as they result in global behaviors that may need to be taken into consideration by other agents who may be part of the system or merely global viewers. A shared strategy can change into norm and vice versa depending on the level of norm internalization and the context which facilitates the implementation of norm emergence and evolution.

For future work, we are further extending the formalization of shared strategy. We are also exploring how shared strategies can be implemented into BDI architecture.

5. REFERENCES

[1] S. Crawford and E. Ostrom. A grammar of institutions. *American Political Science Review*, pages 582–600, 1995.

[2] B. Dunin-Keplicz and R. Verbrugge. Collective intentions. *Fundamenta Informaticae*, 51(3):271–295, 2002.

[3] D. North. *Institutions, institutional change and economic performance*. Cambridge university press, 2009.

[4] E. Ostrom. *Understanding institutional diversity.* Princeton Univ Pr, 2005.

Specifying and reasoning about normative systems in deontic logic programming

(Extended Abstract)

Ricardo Gonçalves
rjrg@fct.unl.pt

José Júlio Alferes
jja@fct.unl.pt

CENTRIA & Dep. Informática, Faculdade Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal

ABSTRACT

In this paper we propose the usage of a framework combining standard deontic logic (SDL) and non-monotonic logic programming – deontic logic programs (DLP) – to represent and reason about normative systems.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Languages, Theory

Keywords

Norms, Knowledge representation, Organisations and institutions, Logic-based approaches and methods, Design languages for agent systems

1. INTRODUCTION

Normative systems have been advocated as an effective tool to regulate interaction in multi-agent systems. Essentially, norms encode desirable behaviours for a population of a natural or artificial society. In general, they are commonly understood as rules specifying what is expected to follow (obligations, permissions, ...) from a specific set of facts. Moreover, in order to encourage agents to act according to the norms, normative systems should also be able to specify the application of rewards/sanctions.

Deontic logic [20] deals precisely with the notions of obligation and permission, and it is, therefore, a fundamental tool for modeling normative reasoning. The modal logic KD has emerged as the Standard Deontic Logic (SDL) [3].

Although necessary, SDL has shown not to be sufficient for the task of representing norms [4]. For instance, it is well known its inability to deal with some paradoxes, namely those involving the so-called contrary-to-duty obligations. The main difficulty of SDL is the fact that classical implication does not provide a faithful representation for the conditional obligations that usually populate a normative system.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Works using dyadic modal logics [19, 9] and input-output logic [11] are examples of approaches that model conditional obligations in order to have a behavior more reasonable than SDL in the face of the aforementioned paradoxes.

Another fundamental ingredient for modeling norms is the ability to express defeasible knowledge. This is important for representing exceptions, which are very common in normative rules. Several approaches using non-monotonic logics where applied to the problem of representing and reasoning about norms [16, 14, 1, 2].

For all the reasons aforementioned, the representation and reasoning about normative systems would greatly benefit from a framework combining deontic logic and rule based non-monotonic reasoning. We thus propose a language for representing and reasoning about normative systems that combines deontic logic with non-monotonic logic programming. Several features distinguish our approach from other formalisms, viz. [13, 12, 15, 6, 7, 10], that combine deontic operators with non-monotonic reasoning. First of all, we have a rich language, which allows complex deontic logic formulas to appear in the body and in the head of rules, combined with the use of default negation. Moreover, at the level of the semantics, we endow the normative systems with a purely declarative semantics, which stems from a well-known semantics: the stable model semantics of logic programs.

The fundamental notion in our framework is that of a deontic logic program. This is composed by rules that resemble usual logic program rules but where complex SDL formulas can appear in the place where only atoms were allowed.

Definition 1. A *deontic logic program* is a set of rules $\varphi \leftarrow \psi_1, ..., \psi_n, not\, \delta_1, ..., not\, \delta_m$ (1) where each of $\varphi, \psi_1, ..., \psi_n$, $\delta_1, \ldots, \delta_m$ is an SDL formula.

As usual, the symbol $\leftarrow$ represents rule implication, the symbol "," represents conjunction and the symbol *not* represents default negation. A rule as (1) has the usual reading that φ should hold whenever $\psi_1, ..., \psi_n$ hold and $\delta_1, ..., \delta_m$ are not known to hold.

Note that, contrarily to some works in the literature [6, 10, 7], deontic formulas can appear both in the head and in the body of a rule, and they can be complex formulas rather than just atomic formulas. This extra flexibility is fundamental, for example, to deal with non-compliance and application of sanctions.

A normative system is usually understood as a set of rules that specify what obligations and permissions follow from a given set of facts, and, moreover, that specify sanction and/or rewards. In our approach, we use the deontic logic programs to represent normative systems.

Definition 2. A normative system $\mathcal{N}$ is a deontic logic program.

In order to allow agents and institutions to reason about a normative system, it is very important that it has a rigorous formal semantics which, at the same time, should be clean and as simple as possible. We endow our rich normative language with a declarative semantics, by defining a stable model based semantics [21] for deontic logic programs. The definition of such a semantics for deontic logic programs is not straightforward due to their complex language where, instead of atoms, we can have complex SDL formulas in the head and body of rules. The problem is that, contrarily to the case of atoms, these formulas are not independent. To overcome this difficulty we need to define a notion of interpretation that accounts for such interdependence between these "complex atoms". The key idea of taking theories of SDL as interpretations, contrasted with the usual definition of an interpretation as any set of atoms, allows the semantics to cope with the interdependence between the SDL formulas appearing in the rules. This construction of a stable model semantics for deontic logic programs can be seen as a special case of the general construction of [5] for *parametrized logic programs* in which SDL is taken as the parameter logic.

The thus obtained normative language is quite expressive, and can be shown to embed extant approaches such as an important fragment of input-output logic [10]. The fact that our language has a purely declarative semantics also allows us to have several interesting properties. First of all, the agents (the ones that are subject to the normative system), the modeler (the one that writes down the norms) and the electronic institution (the one responsible for monitoring the agents and applying the sanctions/rewards) can all reason about the normative system in a simple and clear way. Moreover, in this semantics we can define the fundamental notion of equivalence between normative systems, and, what is the more, we are able to define a logic in which we can verify equivalence of normative systems using logical equivalence.

The results achieved open very interesting paths for future research. An example is the use of abductive reasoning over our stable model semantics to allow agents to plan their interaction with the normative system, in order, for example, to avoid sanctions. Being declarative, our normative framework could easily be integrated in normative multi-agent system that use declarative languages for modeling norms [17, 8], allowing an important increasing of expressivity of these norm languages. Although this is not the main focus of such systems, it was realized, viz. [18], the need for more expressive declarative norm languages.

Other interesting topics for future work include the study of how tools for updating logic programs could be used for the fundamental problem of updating normative systems, and how to define a well-founded based semantics for DLP, that is a sound skeptical approximation of the stable model semantics with more favorable computational complexity.

2. ACKNOWLEDGMENTS

This work was partially supported by FCT Project ERRO PTDC/EIA-CCO/121823/2010. R. Gonçalves was partially supported by FCT Grant SFRH/ BPD/ 47245/ 2008.

3. REFERENCES

[1] G. Boella, G. Governatori, A. Rotolo, and L. van der Torre. A logical understanding of legal interpretation. In F. Lin, U. Sattler, and M. Truszczynski, editors, *KR*. AAAI Press, 2010.

[2] G. Brewka. Well-founded semantics for extended logic programs with dynamic preferences. *J. Artif. Intell. Res. (JAIR)*, 4:19–36, 1996.

[3] B. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.

[4] R. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24(2):33–36, 1963.

[5] R. Gonçalves and J. Alferes. Parametrized logic programming. In T. Janhunen and I. Niemelä, editors, *Logics in AI – JELIA*, volume 6341 of *LNCS*, pages 182–194. Springer, 2010.

[6] G. Governatori and A. Rotolo. Bio logical agents: Norms, beliefs, intentions in defeasible logic. *Autonomous Agents and Multi-Agent Systems*, 17(1):36–69, 2008.

[7] J. F. Horty. Deontic logic as founded on nonmonotonic logic. *Ann. Math. Artif. Intell.*, 9(1-2):69–91, 1993.

[8] J. Hubner, J. Sichman, and O. Boissier. Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. *Int. J. Agent-Oriented Softw. Eng.*, 1:370–395, 2007.

[9] D. Lewis. *Semantic analyses for dyadic deontic logic*. Cambridge University Press, 1999.

[10] D. Makinson, Leendert, and V. D. Torre. Input-output logics. *Journal of Philosophical Logic*, 29:2000, 2000.

[11] D. Makinson and L. van der Torre. Constraints for input/output logics. *Journal of Philosophical Logic*, 30:155–185, 2001.

[12] L. T. McCarty. Defeasible deontic reasoning. *Fundam. Inform.*, 21(1/2):125–148, 1994.

[13] D. Nute. *Defeasible deontic logic*. Springer, 1997.

[14] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1), 1997.

[15] Y. U. Ryu and R. M. Lee. *Defeasible deontic reasoning: a logic programming model*, pages 225–241. John Wiley and Sons Ltd., Chichester, UK, 1993.

[16] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The british nationality act as a logic program. *Com. ACM*, 29:370–386, 1986.

[17] N. A. M. Tinnemeier, M. Dastani, and J.-J. C. Meyer. Roles and norms for programming agent organizations. In C. Sierra, C. Castelfranchi, K. S. Decker, and J. S. Sichman, editors, *AAMAS (1)*, pages 121–128. IFAAMAS, 2009.

[18] N. A. M. Tinnemeier, M. Dastani, J.-J. C. Meyer, and L. W. N. van der Torre. Programming normative artifacts with declarative obligations and prohibitions. In *IAT*, pages 145–152. IEEE, 2009.

[19] L. W. N. van der Torre. Contextual deontic logic: Normative agents, violations and independence. *Ann. Math. Artif. Intell.*, 37(1-2):33–63, 2003.

[20] G. H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.

[21] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. pages 1070–1080. MIT Press, 1988.

Normative Systems require Hybrid Knowledge Bases (Extended Abstract)

Marco Alberti
Ana Sofia Gomes
Ricardo Gonçalves
Matthias Knorr
João Leite
Martin Slota

CENTRIA & Departamento de Informática, Universidade Nova de Lisboa, Portugal
{m.alberti | sofia.gomes | rjrg | mkn | jleite | mslota}@fct.unl.pt

ABSTRACT

In this extended abstract we borrow an example from the Portuguese Penal Code to advocate that norms used to regulate interaction in human societies, just as those used in multi-agent systems, require the joint use of the features based on the Closed World Assumption of rules in Logic Programming and those based on the Open World Assumption of ontologies in Description Logics, all of which are provided by Hybrid MKNF Knowledge Bases.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Languages, Theory

Keywords

Norms, Knowledge representation, Organisations and institutions, Design languages for agent systems

1. INTRODUCTION

Normative systems have long been advocated as an effective tool to regulate interaction in multi-agent systems, and the theory and practice of normative multi-agent systems is a young and very active research area.

Essentially, norms encode desirable behaviours for a population of a natural or artificial society. For example, a (conditional) norm might specify that drivers are expected to stop if so signalled by an authority. In general, norms are commonly understood as a specification of what is expected to follow (obligations, goals, contingency plans, advices, actions, ...) from a specific state of affairs.

Nowadays, many popular organisational models for specification and practical implementation of multi-agent systems are partly based on normative notions (see, e.g., [2] and references therein). Typically, these systems take a formal representation of the normative system and, through automated reasoning, check observable agents' behaviours against the norms, in order, for instance, to detect norm violation and to apply sanctions.

One key problem to implement such practical normative systems involves the representation of, and reasoning with norms. On the one hand, we need a representation language that is expressive enough to represent the norms we wish to encode, on the other hand it must be such that we can reason with it efficiently.

Ever since the formalisation of the British Nationality Act using Logic Programming (LP) by Sergot et al. [7], *non-monotonic* formalisms have been used to deal with many aspects of legal rules and regulations. The non-monotonic features common to existing approaches which implement the Closed World Assumption have been shown necessary in the context of reasoning with norms, laws and regulations, for example, to allow default reasoning needed, e.g., to represent exceptions.

Despite the specificities of multi-agent systems, many of their aspects are inspired by human societies, and an intimate parallel between laws in real-world legal systems and norms in multi-agent systems can often be drawn. Therefore, in this paper, instead of tailoring an artificial multi-agent based scenario to illustrate our points, we exploit the Portuguese Penal Code (PPC), that is filled with examples rich in intrinsic subtleties.

Example 1. To illustrate the need for default reasoning to represent exceptions, consider the following PPC articles:

Article 131. Murder

Who kills another person shall be punished with imprisonment from eight to sixteen years.

However, exceptional circumstances for murder increase the duration of the conviction:

Article 132. Aggravated murder

1 – If death is produced in circumstances which present a special censurability, the agent is punished with imprisonment of twelve to twenty-five years.
2 – Is likely to reveal the special censurability referred to in the preceding paragraph, among others*, the fact that the agent:*

(...)
d) employs torture or cruel act to increase the suffering of the victim;
(...)
h) performs the act with at least two other people;

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Accordingly, killing someone is punished with imprisonment from eight to sixteen years, *except* if some additional facts are established, in which case the penalty is aggravated. In other words, unless one of these aggravating facts is proved, *by default* this crime is punished with imprisonment from eight to sixteen years. The relevant part can easily be captured by LP rules using non-monotonic default negation as follows:

$$\begin{aligned}\mathsf{Murder}(X,Y) &\leftarrow \mathsf{Killing}(X), \mathsf{Guilty}(X,Y),\\ &\qquad {\sim}\mathsf{AggrMurder}(X,Y).\\ \mathsf{AggrMurder}(X,Y) &\leftarrow \mathsf{Killing}(X), \mathsf{Guilty}(X,Y),\\ &\qquad \mathsf{Censurable}(X).\end{aligned}$$

together with the definition of $\mathsf{Censurable}(X)$, and where $\mathsf{Guilty}(X,Y)$ represents the fact that subject Y was found guilty of punishable event X.

However, in legal reasoning, we sometimes need to represent concepts that cannot be handled by the LP approach, namely those involving open world knowledge. Whereas some extensions of LP allow the representation of some open world knowledge, e.g., through the use of strong negation, they cannot deal with cases that require reasoning with existential knowledge and unknown individuals, often required when dealing with norms.

Description Logics (DLs) [1], decidable fragments of classical first-order logic, offer an alternative and are considered the standard logical representation for expressive ontologies. They are based on the Open World Assumption and allow for reasoning with unknown individuals.

Example 2. Going back to the previous example, encoding item h) as a condition to establish special censurability requires that we refer to (at least) two possibly unknown individuals (a witness or a security camera recording could be sufficient to establish that the culprit acted together with two more people, but not their identity). The relevant part can be encoded in DLs as

$$\geq 3\ \mathsf{PerformedBy.Person} \sqsubseteq \mathsf{Censurable}\ ,$$

meaning that special censurability of the act is established if it was committed by at least three people. Such a condition cannot be expressed in the body of an LP rule since it does not permit encoding unknown individuals. Similarly, it would not be possible to assert in a rule that some act was performed by, e.g., five people, but whose identities, besides that of the accused, are unknown.

Not only can DLs be seen as a solution to properly deal with existential knowledge and unknown individuals, they are quite appropriate for taxonomic representations of facts and concepts, and have been acknowledged in the area of legal reasoning as a fundamental tool for modelling and reasoning about the hierarchy of legal concepts [6]. Unfortunately, DLs do not allow for default reasoning.

Despite the fact that the example used was extracted from a human legal system, normative multi-agent systems are not different when it comes to norms requiring exceptions (e.g., agents should not be allowed to access some confidential information *unless* they have specific privileges) and reasoning with existential knowledge and unknown individuals (which is becoming increasingly relevant, now that privacy issues in MAS are receiving more attention, and the identity of the agent that performed an action might not be available).

In order to properly represent and reason with rich norms that include all these features, we need an approach that tightly combines the best of the two families of formalisms – LP rules and DLs – and exhibits, at least, the following key features:

- have a formal rigorous semantics so that norms can be shared by agents and institutions, and both can reason with the norms to determine their actions and sanctions;
- support both the Open and Closed World Assumption, and the ability to represent and reason with existential knowledge and unknown individuals;
- be equipped with efficient operational semantics to be usable in practical multi-agent systems.

To the best of our knowledge, no existing system concurrently provides seamlessly the expressivity of LP and DLs to represent norms.

With these requirements in mind, we propose that normative frameworks use the joint expressivity of LP and DLs. In such frameworks, facts are represented as a DL ABox, and norms as a combination of a DL TBox and LP rules. The semantics can then be rooted on Hybrid MKNF [5], a language that tightly integrates rules and ontologies, or on its well-founded version [4] for which top-down querying procedures have been introduced and an implementation with support for the DL $\mathcal{ALCQ}$ is available [3].

2. ACKNOWLEDGMENTS

M. Knorr was partially supported by FCT Project ERRO PTDC/EIA-CCO/121823/2010, A. S. Gomes by FCT Grant SFRH/BD/64038/2009, R. Gonçalves by FCT Grant SFRH/BPD/47245/2008, J. Leite by FCT Project ASPEN PTDC/EIA-CCO/110921/2009 and M. Slota by FCT Grant SFRH/BD/38214/2007.

3. REFERENCES

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2nd edition, 2007.

[2] V. Dignum, editor. *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models.* Information Sci. Ref., 2009.

[3] A. S. Gomes, J. J. Alferes, and T. Swift. Implementing query answering for hybrid MKNF knowledge bases. In *Procs of PADL'10*, volume 5937 of *LNCS*. Springer, 2010.

[4] M. Knorr, J. J. Alferes, and P. Hitzler:. Local closed world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*, 175(9–10):1528–1554, 2011.

[5] B. Motik and R. Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5), 2010.

[6] G. Sartor, P. Casanovas, M. Biasiotti, and M. Fernandez-Barrera, editors. *Approaches to Legal Ontologies: Theories, Domains, Methodologies.* Law, Governance and Technology series. Springer, 2011.

[7] M. Sergot, F. Sadri, R. Kowalski, F. Kriwaczek, P. Hammond, and H. Cory. The british nationality act as a logic program. *Commun. ACM*, 29:370–386, 1986.

Self-management of Ambient Intelligence Systems: a Pure Agent-based Approach*

(Extended Abstract)

Inmaculada Ayala
Universidad de Málaga
Bulevar Louis Pasteur, 35
Málaga, Spain
ayala@lcc.uma.es

Mercedes Amor
Universidad de Málaga
Bulevar Louis Pasteur, 35
Málaga, Spain
pinilla@lcc.uma.es

Lidia Fuentes
Universidad de Málaga
Bulevar Louis Pasteur, 35
Málaga, Spain
lff@lcc.uma.es

ABSTRACT

Ambient Intelligence systems (AmI) are normally composed of networked heterogeneous devices with critical resource limitations. One of the biggest requirements of AmI systems is that they should be capable of self-management in order to adapt their behavior and resources to environmental conditions and variable device resources. Autonomous agents are a good option to endow AmI systems with self-managing capabilities, but current agent platform implementations do not adequately address the heterogeneity requirements of AmI systems, given the impossibility until now of producing pure agent-based solutions. In this paper we present a pure agent-based solution for self-managing AmI systems, with particular emphasis on defining a working solution considering the diversity of devices and communication protocols through which AmI devices must interoperate.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Design, Experimentation, Management

Keywords

Agent Oriented Software Engineering, Self-management, Ambient Intelligence, Lightweight devices

1. INTRODUCTION

Ambient Intelligence (AmI) environments represent a new generation of computing systems equipped with devices with special capabilities that make people aware of the environment and react to it, in a more natural way [4]. These systems are composed of a large variety of networked heterogeneous devices, such as mobile phones or Wireless Sensors Networks (WSNs). Normally, the majority of AmI devices show symptoms of degradation, such as energy loss or failure of some network nodes, which requires explicit management action, for example saving energy to guarantee the system's survival. Consequently AmI systems demand the reconfiguration of their internal functioning in response to changes in their environment. This means that AmI systems must behave as autonomic systems with a self-managing capacity.

The self-management properties are inspired by the properties of agents [5], such as autonomy, distribution and proactiveness. This leads us to consider agents and Multi-Agent Systems (MASs) as effective metaphors for system design and implementation of AmI scenarios. In this paper we will focus on how an agent-based solution can help to implement the self-management requirement of AmI systems. Several approaches already exploit MAS in the context of self-managing AmI systems [2][3], but the solution proposed by most of them cannot be considered a pure agent-based solution since agents are just used to apply Artificial Intelligence techniques (e.g. learning or planning algorithms), or as autonomic managers of an add-on autonomic system.These solutions are not energy efficient, as the self-management tasks imply an extra traffic between the autonomic agent manager(s) and the managed devices. So, pure agent-based solutions are more energy efficient, since they minimize the self-management traffic, which can be considerable in AmI systems with for example thousands of sensors.

In this paper we introduce *self-StarMAS*, a set of cooperating agents, running in each device of an AmI system, able to communicate and interoperate through heterogenous communication protocols, and with the capacity of self-management adapted to each kind of agent/device.

2. THE SELF-STARMAS SYSTEM

In this section we will present the self-StarMAS system (see fig. 1) focusing on the challenges that pose the use of agent technology to self-manage AmI systems and how they are addressed by our approach. The use of agents to support self-management is not a new research area, but we consider that current solutions do not address all these challenges well.

C1 - Self-management: Much research on self-management is progressing and several self-* properties introduced by IBM can be considered a good starting point (self-configuring, self-optimizing,...). **Achieving:** We propose a MAS composed by a set of cooperating agents with the

*This work has been supported by the Project RAP TIN2008-01942 and the Project FamWare P09-TIC-5231.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

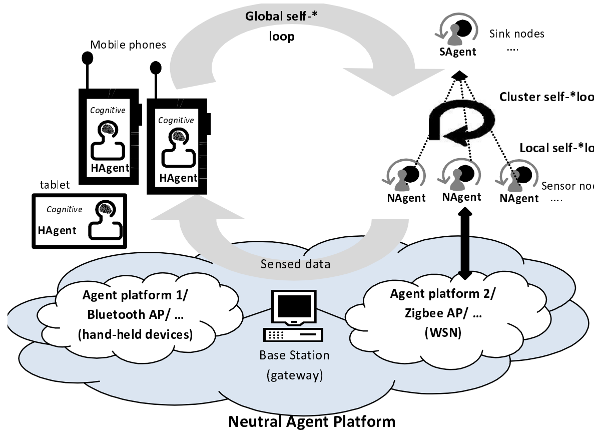


Figure 1: The self-StarMAS global architecture

capacity to provide self-management. Each individual AmI device is endowed with an agent with self-management capacities, so we propose a pure agent-based solution.

C2 - Decentralized adaptation: The highly decentralized and embedded nature of the devices involved makes it hard to enforce some forms of direct control over each of the networked devices [6]. This makes traditional centralized approaches that use one agent, or a fixed set of agents to perform a kind of self-management task, inadequate and economically inviable, leading to a demand for novel decentralized solutions. **Achieving:** Agents in self-StarMAS are structured following a hierarchical organization with different types of agents (see fig. 1), adapted to the resources of each device and its physical organization in the WSN. Firstly, agents running in mote-like sensors include a lightweight implementation of self-managing properties, with only a limited set of previously known reconfiguration actions. They specialize in applying policies to save energy levels but the adaptation is performed locally (*Local self-*loop*). Secondly, sink nodes of the WSN have agents with higher level organization capacities. The policies applied by them have the goal of self-managing a group of agents, representing a group of sensors. Finally, agents running in hand-held devices have the capacity to both self-manage their internal functioning, and act as managers of the self-StarMAS system (*Global self-*loop*).

C3 - Devices heterogeneity: AmI systems are composed of set of heterogeneous lightweight devices, which are continuously being updated. This means that any pure agent-based solution for self-managing AmI systems must consider that agents will be running in different devices, but even so, they must interoperate using the services of a common agent platform (AP). **Achieving:** Self-StarMAS includes different agent designs for different kind of AmI devices, but they are completely interoperable using FIPA-compliant negotiation protocols adapted for self-management in AmI systems. Until now, only a few APs can be executed in mobile phones and only a couple of approaches [1] can be executed in sensors such as Sun SPOT, but in this last case agents in different types of devices are not interoperable.

C4 - Communication heterogeneity: In a ubiquitous environment, agents running in different devices must be able to communicate with each other. The challenge is to support the communication of agents running in devices, using the lightweight devices's own proprietary communication technology. **Achieving:** Self-StarMAS agents are platform-neutral which means that agents are able to support the discovery and interaction between agents running in devices using different communication technologies or APs. As part of our solution, we provide different possibilities in order to support agent interoperation: (i) use an existing AP (e.g. JADE-Leap); (ii) use our infrastructure to support the communication between hand-held devices using WiFi and WSNs using Zigbee; and (iii) use the native communication protocol of the AmI device (e.g. for hand-held devices we implemented an AP using Bluetooth).

To the best of our knowledge, only our work properly addresses all the challenges presented above. We present a workable solution instead of a theoretical one, the results of which sometimes cannot be applied to real AmI systems. The approach proposed here pursues a general goal of making the agent technology a genuine alternative to develop complete AmI applications.

3. CONCLUSIONS

In this paper we present, self-StarMAS, a pure agent-based solution for the self-management of AmI systems. The main characteristic of the self-StarMAS is that it provides an homogenous management of AmI systems, although it is composed of heterogeneous agents running on heterogenous devices, communicating through heterogeneous AP and protocols. The architecture of each type of agent is customized to the resources of each device type, allowing the definition of reconfiguration policies adapted to the role and resources of each agent. Up to now we have defined some control policies focused on energy saving, but the self-StarMAS system is extensible, so new policies can be defined easily.

4. REFERENCES

[1] F. Aiello, G. Fortino, R. Gravina, and A. Guerrieri. A java-based agent platform for programming wireless sensor networks. volume 54, pages 439–454, Oxford, UK, March 2011. Oxford University Press.

[2] B. An, V. Lesser, D. Westbrook, and M. Zink. Agent-mediated Multi-step Optimization for Resource Allocation in Distributed Sensor Networks. In Y. Tumer and S. Sonenberg, editors, *Proc. of AAMAS 2011*, Taipei, Taiwan, May 2011. IFAAMAS.

[3] I. Chun, J. Park, H. Lee, W. Kim, S. Park, and E. Lee. An agent-based self-adaptation architecture for implementing smart devices in smart space. *Telecommunication Systems*, pages 1–12. 10.1007/s11235-011-9547-8.

[4] D. J. Cook, J. C. Augusto, and V. R. Jakkula. Review: Ambient intelligence: Technologies, applications, and opportunities. *Pervasive Mob. Comput.*, 5:277–298, August 2009.

[5] M. C. Huebscher and J. A. McCann. A survey of autonomic computing: degrees, models, and applications. *ACM Comput. Surv.*, 40:7:1–7:28, August 2008.

[6] R. Quitadamo and F. Zambonelli. Autonomic communication services: a new challenge for software agents. *Autonomous Agents and Multi-Agent Systems*, 17:457–475, December 2008.

Enhancing Decentralized Service Discovery through Structural Self-Organization

(Extended Abstract)

E. del Val[1], M. Vasirani[2] M. Rebollo[1], A. Fernandez[2]
[1]Universitat Politècnica de València
{edelval,mrebollo}@dsic.upv.es
[2] Universidad Rey Juan Carlos
{matteo.vasirani,alberto.fernandez}@urjc.es

ABSTRACT

The efficiency of service discovery in distributed systems relies on the collaboration of the agents and the structure of the relations established among them. Structural relations cannot be static, agents should be able to adapt their links as the domain conditions and their interests change. This self-organization considerably improves the performance of the service discovery process. We present a self-organization mechanism that facilitates the task of decentralized service discovery and improves its efficiency in dynamic environments. Each agent has local knowledge about their neighbors and the queries received during the discovery process. With this information, each agent is able to decide when it is more appropriate to modify its structural relations with its direct neighbors and what the most suitable acquaintances to replace them are.

Categories and Subject Descriptors

H. [**Information Systems**]

General Terms

Management, Performance

Keywords

Services, Self-adaption, Self-Organization, Similarity, Homophily

1. STRUCTURAL RELATIONS FOR SELF-ORGANIZING AGENTS

Structural relations define the set of agents with which an agent establishes a relation. One of the criteria considered to establish structural relations is *homophily* [3]. Homophily is present in many complex networks. The idea behind the homophily concept is that individuals tend to interact and establish links with similar individuals through a set of social dimensions. Therefore, at the structural level, communities of similar agents (provide similar services) are created in a decentralized way. The resulting system structure can be considered to be preferential attachment network, which grows according to a simple self-organized process [4].

Each agent in this structure has a local knowledge about its structural relations, as well as knowledge about the environment. Formally, the knowledge model of agent i is a tuple $< \mathcal{A}_i^T, \mathcal{A}_i^K, \mathcal{A}_i^R, \mathcal{C} >$, where $\mathcal{A}$ is the set of agents in a multiagent system and

- $\mathcal{A}_i^T \subseteq \mathcal{A}$ is the set of agents that agent i can communicate with.
- $\mathcal{A}_i^K \subseteq \mathcal{A}$ is the set of acquaintances of agent i. We point out that, in general, $\mathcal{A}_i^T \subseteq \mathcal{A}_i^K$, since an agent is always at least aware of the existence of its neighbors in the communication network.
- $\mathcal{A}_i^R \subseteq \mathcal{A}$ is the set of agents that agent i has a relation with.
- $\mathcal{C}$ is the set of possible categories (relation types). We consider that each agent is associated, at least, to one category that is related to the services it provides.

1.1 Self-Organization in Service Discovery

When an agent needs a service, since there is not a service discovery facilitator nor a registry to be queried, it sends a query q that contains the requirements for a provider agent t, which is unknown. The query contains the service description and its category. The receiver agent performs a matchmaking process of the query against the services it offers. If the best matching service has a degree above a certain threshold, then the search ends successfully. In the case of an unsuccessful matching, the query is forwarded to one of its neighbors $j \in \mathcal{A}_i^R$, which is the most similar neighbor to the target agent t considering the semantic closeness (homophily degree) to the desired service and the degree (number of connections) of the agent [4]. This process is repeated until an agent that offers a service that is 'similar enough' is found or when the TTL (Time To Live) of the query ends. The criterion of 'similar enough' is established by the agent that starts the service search process as a semantic similarity threshold.

Each time an agent forwards a query, it updates the information about its structural relations and adds its identification to the query along with its matching degree. If the query has been successfully solved, the agent that started the process adds the target agent t as an acquaintance agent.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The target agent that has the required service propagates a message to the agents that participated in the search. Each agent analyzes the utility of its links. In the case that the agent has an acquaintance that has a higher utility than the current links, it decides to break useless structural relations with current neighbors and creates new ones with acquaintances.

The criterion to evaluate structural relations is based on their utility. In the context of service discovery, the *Utility* of a structural relation between agents i and j for a category c is defined as:

$$U^c_{i,j} = \frac{\#^c_i}{\#_i} \cdot m^c_j, \quad (1)$$

where $\frac{\#^c_i}{\#_i}$ is the ratio between the number of queries for service category c received by i and the total number of queries received by i so far, and $m^c_j \in (0, 1)$ is the average degree of match for queries of category c performed by agent j. How this degree of match is calculated is explained in more detail in [4].

If a relation with a neighbor is used to address requests of services of a certain category, then it is interesting for the agent to maintain the link. However, if a relation has not not being used, then the agent must decide whether or not to maintain it. The utility of an structural relation decays exponentially according to Eq. 2 [2]:

$$\tau^c_{i,j} = 1 - e^{-\rho \cdot U^c_{i,j}} \quad (2)$$

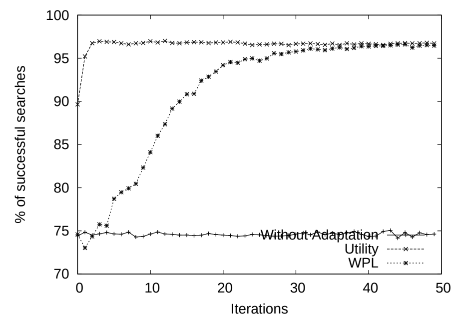
where $\rho \in (0, \infty)$ is an adjustable parameter and $U^c_{i,j} \in \mathbb{R}^+$ is the utility of the established relation between agent i and agent j for the category c .

Each agent i maintains a vector of values $\boldsymbol{\tau}_{i,j} = [\tau^1_{i,j} \ldots \tau^{|\mathcal{C}|}_{i,j}]$ for each one of its neighbors. An element $\tau^c_{i,j}$ of the vector represents, the probability of sending a query of category c through agent j. When agent i establishes a new relation of category $c \in \mathcal{C}$ with agent j, the corresponding value $\tau^c_{i,j}$ is also initialized to 1. New relations with some of the acquaintances can be formed. Thus, for every acquaintance $j \in \mathcal{A}^K_i - \mathcal{A}^R_i$, agent i maintains a vector of values $\boldsymbol{\eta}_{i,j} = [\eta^1_{i,j} \ldots \eta^{|\mathcal{C}|}_{i,j}]$ that represents the probability that agent i will establish a new relation of category $c \in \mathcal{C}$ with agent j. The probability of actually establishing a new relation with agent j is given by an Equation similar to Eq. 2.

2. EXPERIMENTS

We evaluated the influence of the proposed mechanism based on the utility functions for the evaluation of structural links with neighbors and acquaintances, and the criteria to each agent decides when it is more appropriate change current structural relations. We compared the results of our proposal with those obtained without using adaptation mechanisms and with a Q-learning algorithm called Weighted Policy Learner (WPL)[1].

Considering the number of changes in the structural relations between agents, the 'Utility' mechanism initially generates a high number of changes if we compare it with the 'WPL' mechanism. The 'WPL' follows a constant rate of changes, and the adaptation is slower than the 'Utility' mechanism. 'Utility' allows agents to only rewire links when the acquaintance links are significantly better than the current links. This makes agents change a reasonable number of structural relations. The success of the service discovery system is improved with both strategies (see Fig. 1). With both adaptation mechanisms, agents are able to create new links that connect with other agents that offer the most demanded services. The 'Utility' mechanism improves the success rate in the first two iterations. However, the WPL takes more time to achieve a success rate over 95%.



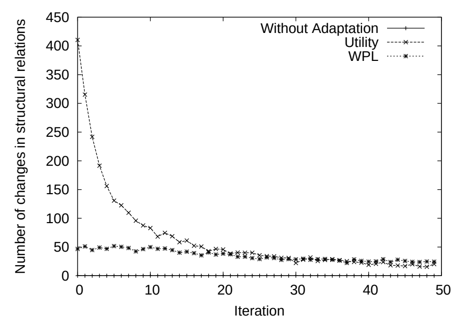


Figure 1: Evolution over time of system measures using Utility and WPL mechanisms. (Top) Percentage of queries that end successfully. (Bottom) Number of structural changes in the system.

3. ACKNOWLEDGEMENTS

Work partially supported by the Spanish Ministry of Science and Innovation through grants TIN2009-13839-C03-01, TIN2009-13839-C03-02 (co-funded by Plan E), CSD2007-0022 (CONSOLIDER-INGENIO 2010), FPU grant AP-2008-00601 awarded to E. del Val.

4. REFERENCES

[1] S. Abdallah and V. Lesser. Multiagent reinforcement learning and self-organization in a network of agents. In *6th AAMAS*, 2007.

[2] E. M. Jin, M. Girvan, and M. E. J. Newman. Structure of growing social networks. *Phys. Rev. E*, 2001.

[3] P. Lazarsfeld. Friendship as a social process: A substantive and methodological analysis. *Freedom and Control in Modern Society*, 1954.

[4] E. D. Val, M. Rebollo, and V. Botti. Self-Organized Service Management in Social Systems. In *45 HICCS*, 2012.

Cloning, Resource Exchange and Relation Adaptation: A Self-organising Multi-Agent Framework

(Extended Abstract)

Dayong Ye
University of Wollongong
NSW 2522 AU
dy721@uowmail.edu.au

Minjie Zhang
University of Wollongong
NSW 2522 AU
minjie@uow.edu.au

Danny Sutanto
University of Wollongong
NSW 2522 AU
danny@elec.uow.edu.au

ABSTRACT

In this paper, a self-organising multi-agent framework is proposed. Different from current related approaches which concerned only a single principle of self-organisation, this framework synthesises the three principles of self-organisation, i.e., agent cloning/spawning, resource exchange and relation adaptation. In this framework, an agent can autonomously generate new agents when it is overloaded, exchange resources with other agents if necessary, and adapt relations with other agents to achieve a better network structure. In this way, agents in this framework can adapt to dynamic environments.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Self-organisation, Adapation, Learning

1. INTRODUCTION

Self-organisation is defined as "the mechanism or the process enabling the system to change its organisation without explicit external command during its execution time [2]". For self-organising sytems design, Mathieu et al. [1] pointed out that a self-organising system should include three principles. The first one is that agents within the system will generate new agents to take part of their load once they are overloaded. The second one is that agents can exchange skills or resources, if necessary, between each other to increase autonomy. The last one is that agents should be able to create new specific relations between agents in order to remove the middle-agents. Currently, each of the three principles has attracted many research efforts. However, to the best of our knowledge, there lacks an attempt which combines the three principles in a single framework in order to achieve better performance compared with those self-organisation approaches which considers only one of the three principles. Towards this end, in this paper, we present a self-organising multi-agent framework which combines the three principles together, i.e., agent cloning/spawning, resource exchange and relation adaptation. Our framework is illustrated within a general platform, i.e., distributed task allocation. By employing a general platform, instead of a particular existing system, our framework can be potentially applied to a wide variety of applications.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. MODEL DESCRIPTION

The task allocation network is modeled as a tuple $\langle A, N, T, \Gamma\rangle$. Each element is described as follows.

- A is a set of collaborative agents in the network, i.e., $A = \{a_1, ..., a_n\}$.
- $N = \{N_1, ..., N_n\}$, where each N_i demonstrates the neighbours of agent a_i.
- $T = \{t_1, ..., t_m\}$ is a set of task types which will arrive at the network.
- $\Gamma = \{\gamma_1, ..., \gamma_l\}$ is a set of resource types which exist in the network.

The neighbour set of each agent N_i consists of three different neighbours, i.e., *peer*, *subordinate* and *superior*, which are formed by two relations, i.e., *peer-to-peer* relation and *subordinate-superior* relation.

A task type defines which resource types are needed and the quantity of each type of resources is required. In addition, a task type also dictates the task benefit, the task service time and the task maximum waiting time before being executed. If a task cannot be completed before the deadline of its service time, the benefit of this task decreases gradually with time elapse till 0.

There is a continuous dynamic stream of tasks which arrive at the network. Each task Φ randomly corresponds to a task type in T. Φ can be divided into several subtasks and each subtask, $\varphi_i \in \Phi$, requires a particular resource type and a specific amount of this type of resources which are indicated by the corresponding task type. In addition, each subtask has a relevant benefit paid to the agent which successfully completes the subtask. In this paper, a subtask φ_i is modeled as a token Δ_i which can be passed in the network to find a suitable agent to complete. Each token consists of not only the information about resource requirement of the corresponding subtask, but also the token traveling path which is composed of those agents that the token has passed.

3. THE SELF-ORGANISING MULTI-AGENT FRAMEWORK

3.1 Agent Cloning/Spawning

When an agent is overloaded (i.e., it cannot complete the subtasks in its subtask waiting list before their respective deadlines or it has too many neighbours to keep with), the agent will create a new agent to handle part of its load. The agent has two options, namely cloning an agent or spawning an agent.

Specifically, for a single agent, spawning is triggered when the task load exceeds the agent's ability to finish it on time, given the agent's current status and resource level. In this condition, the agent spawns some new agents and assigns the most benefit tasks and corresponding resources to them. These spawned agents are as *subordinate*s of the original agent, but they cannot establish relations with other agents. When the spawned agents finish the assigned tasks, they are in an idle status. When the spawned agents keep in an idle status for a pre-defined period, namely that no more such subtasks need to be completed, they will be destroyed by the original agent to save relation management load.

On the other hand, cloning happens when an agent has too many neighbours, which means that the agent has a heavy overhead for managing relations with other agents. In this situation, to avoid possible communication congestion, the agent clones an agent which has the same resources as itself, and assigns some neighbours to the cloned agent. The original agent keeps a *peer* relation with the cloned one. Different from the spawning agents, the cloned agent cannot be destroyed by the original agent. Instead, the original and cloned agents will compose together, once the total number of neighbours of them is less than a pre-defined threshold.

3.2 Resource Exchange

For a single agent, when a resource has not been used for a long time, the agent will transfer the resource to a neighbouring agent which really needs this resource. Here, we devise a Q-learning algorithm to handle it (**Algorithm 1**). An action in this Q-learning algorithm represents transferring a resource to a neighbour.

Algorithm 1: Resource transfer according to a_i

for each $a_i \in A$ **do**
  $Res \leftarrow GetResource()$;
  for each resource $\gamma_i \in Res$ **do**
    a_i initialises Q-values and $\boldsymbol{\pi}$;
    a_i informs its neighbours;
    neighbours respond to a_i;
    $\mathbf{r} \leftarrow$ the reward vector for neighbours;
    for each neighbour of a_i, i.e., $n_j \in N_i$ **do**
      $Q(n_j) \leftarrow (1-\alpha_1)Q(n_j) + \alpha_1 \cdot r_j$;
    end for
    $\bar{r} \leftarrow \frac{1}{|N_i|}\sum_{n_j \in N_i} r_j$;
    for each neighbour of a_i, i.e., $n_j \in N_i$ **do**
      $\pi(n_j) \leftarrow \pi(n_j) + \zeta(Q(n_j) - \bar{r})$;
    end for
    $\boldsymbol{\pi} \leftarrow Normalise(\boldsymbol{\pi})$;
    a_i selects a neighbour based on $\boldsymbol{\pi}$;
  end for
end for

3.3 Relation Adaptation

Our relation adaptation algorithm is based on the past information of the individual agents. Specifically, agents use the information about the past task allocation processes to evaluate their relations with other agents. We develop a multi-agent Q-learning algorithm to tackle the relation adaptation problem. **Algorithm 2** demonstrates our relation adaptation algorithm in pseudocode form.

Algorithm 2: Relation adaptation according to a_i

1 $Candidates_i \leftarrow a_i$ selects agents in the network;
2 **for each** $a_j \in Candidates_i$ **do**
3   $Act_i \leftarrow available\_actions(a_i, a_j)$;
4   $Act_j \leftarrow available\_actions(a_i, a_j)$;
5   **for each** $x \in Act_i$, $y \in Act_j$ **do**
6   Initialise Q_{ix} and Q_{jy} arbitrarily;
7   **for** $k = 0$ to a predefined integer **do**;
8   calculate $\pi_{ix}(k)$ and $\pi_{jy}(k)$;
9       $Q_{ix}(k+1) = Q_{ix}(k) + \pi_{ix}(k)\alpha_2(\sum_y r_i^{x,y}\pi_{jy}(k) - Q_{ix}(k))$;
10       $Q_{jy}(k+1) = Q_{jy}(k) + \pi_{jy}(k)\alpha_2(\sum_x r_j^{x,y}\pi_{ix}(k) - Q_{ix}(k))$;
11   **end for**
12   **end for**
13   $\langle x_{opti}, y_{opti}\rangle \leftarrow argMax_{match(x,y)}(Q_{ix} + Q_{jy})$;
14   a_i, a_j take actions x_{opti} and y_{opti}, respectively;
15   $\mu_{ij} \leftarrow \mu_{ij} + (L_j^i/\rho_1 - 1)$;
16   **if** $\mu_{ij} > 1$ **then** $\mu_{ij} \leftarrow 1$;
17   **if** $\mu_{ij} < 0$ **then** $\mu_{ij} \leftarrow 0$;
18   $\mu_{ji} \leftarrow \mu_{ij}$;
19 **end for**

When finishing learning Q-values, a_i and a_j (Line 13) cooperate to find the optimal actions for both of them.

Algorithm 3 illustrates the reasoning aspect of each agent for selecting a group of agents to initilise the relation adaptation process.

Algorithm 3: Candidates selection of each agent

for each $a_i \in A$ **do**
  $Candidates_i \leftarrow \emptyset$;
  for each $\Delta_k \in tokens_i$ **do**
    statistics of $\Delta_k.owner$;
  end for
  if $\exists$ # of same $\Delta_k.owner > \rho_2$ **and**
    $\Delta_k.owner \notin Neig_i^{\sim} \vee Neig_i^{\succ} \vee Neig_i^{\prec}$ **then**
    $Candidates_i \leftarrow Candidates_i \cup \{\Delta_k.owner\}$;
  end if
  if $\exists$ # of same $\Delta_k.owner < \rho_3$ **and**
    $\Delta_k.owner \in Neig_i^{\sim} \vee Neig_i^{\succ} \vee Neig_i^{\prec}$ **then**
    $Candidates_i \leftarrow Candidates_i \cup \{\Delta_k.owner\}$;
  end if
end for

4. CONCLUSION

This paper introduced a self-organising multi-agent framework which considers the three principles of self-organisation, i.e., agent cloning/spawning, resource exchange and relation adaptation. Through combining the benefits of the three principles, our framework outperforms state of the art approaches which focus on a single principle only. Since our framework is decentralised and continuous over time without external control, it meets the definition of self-organisation given by Serugendo et al. [2].

5. REFERENCES

[1] P. Mathieu, J.-C. Routier, and Y. Secq. Principles for dynamic multi-agent organizations. In *PRIMA'02*, pages 109–122, Tokyo, Japan, Aug. 2002.

[2] G. D. M. Serugendo, M.-P. Gleizes, and A. Karageorgos. Self-organization in multi-agent systems. *The Knowledge Engineering Review*, 20(2):165–189, 2005.

Dynamic change impact analysis for maintaining and evolving agent systems

(Extended Abstract)

Hoa Khanh Dam and Aditya Ghose
School of Computer Science and Software Engineering
University of Wollongong
New South Wales 2522, Australia
{hoa,aditya}@uow.edu.au

ABSTRACT

In contrast to an increasing number of agent-based applications in various domains, there has been very little work on maintenance and evolution of agent systems. This paper addresses this gap with a focus on change impact analysis, i.e. estimating the potential effects of changes before they are made as an agent system evolves. We propose a technique for performing impact analysis in an agent system using dynamic information about agent behaviour. Our approach builds a representation of an agent's behaviour by analyzing its execution traces which consist of goals and plans, and uses this representation to estimate impacts.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Distribution, Maintenance, and Enhancement

General Terms

Design

Keywords

change impact analysis, multi-agent systems

1. INTRODUCTION

Complex agent-based applications will evolve and will need to be maintained throughout their life, which would require substantial costs. The focus of this paper is on *change impact analysis* of agent systems – predicting the potential consequences of a proposed change. Change impact analysis [1] usually starts with the software maintainer examining the change request and determining the entities initially affected by the change (i.e. the *primary changes*). The software maintainer then determines other entities in the system that have potential dependency relationships with the initial ones, and forms a set of impacts. Those impacted components also relate to other entities and thus the impact analysis continues this process until a complete impact set is obtained. Change impact analysis plays a major part in planning and establishing the feasibility of a change in terms of predicting the cost and complexity of the change (before implementing it). This help reduce the risks associated with making changes that have unintended, expensive, or even disastrous effects on an existing system. Furthermore, change impact analysis can be used to predict or identify parts of a system that will need to be retested (i.e. regression testing) as a result of changes.

Although notions and ideas from a large body of work addressing change impact analysis for classical software systems (e.g. [1]) can be adapted, agent systems with their distinct characteristics and architectures introduce new problems in software maintenance. For instance, while object-oriented software deals with classes, methods and fields, a typical agent-based software, e.g. the Belief-Desire-Intention (BDI) [4] agents, consists of agents, plans, events/goals and beliefs.

A recent effort [2] has proposed a change impact analysis technique specifically for agent systems. It is however based on static analysis of agent source code, which can safely estimate the impact of changes, but its conservative principle leads to a large impact set which may contain many unnecessary entities. This is because static analysis considers all possible behaviours of a software system while only a subset of such behaviours may be executed in practice.

Therefore, this paper takes a dynamic approach to change impact analysis for agent systems: we propose an impact analysis technique using dynamic information about agent behaviour. Our dynamic impact analysis technique focuses specifically on agent systems, in particular the well-known and widely-used BDI agents. We identify the essential information needed to perform dynamic impact analysis on a BDI agent system. Such dynamic information is collected from execution data for a specific set of agent executions (e.g. executions based on an operational profile or executions of test suites) which contains two key aspects determining the behaviour of a BDI agent system: the *goals* an agent pursued and the *plans* it deployed to achieve those goals. We further define a technique to analyse that information to determine when a plan or goal is changed, what other plans and goals are potentially impacted by the change.

2. DYNAMIC IMPACT ANALYSIS

The hierarchical structure of BDI plans which determine the run-time behaviour of a BDI agent can be viewed as a goal-plan tree where each goal has children representing the

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

relevant plans for achieving it, and each plan has children representing the subgoals (including primitive actions) that it has. This goal-plan tree can be seen as an "and/or" tree: each goal is achieved by a successful execution of one of its plan ("or"), and the success of each plan relies on all of its sub-goals being resolved ("and"). Figure 1 shows an example of such an goal-plan tree. Goal G can be realised by either plan P1 or P2. Plan P1 has two subgoals G1 and G2 in which G1 can be achieved by one of plans P3, P4 and P5, and G2 can be achieved by plan P6. Plan P2 has only one subgoal G3, which can be realised by either plan P7 or P8.

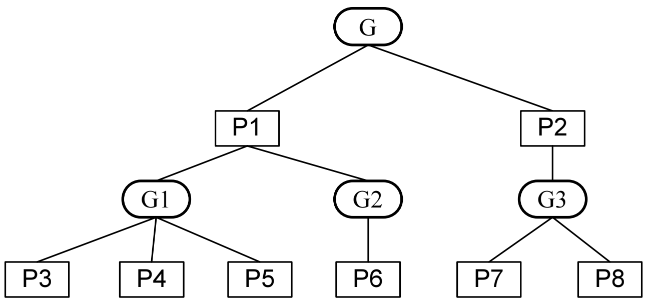


Figure 1: A goal-plan tree for agent A

As an example, suppose we have a single execution trace t, shown by a string of letters in figure 2, for an agent A whose a goal-plan tree appears in figure 1. Note that G_p denotes goal G being posted, whereas G_s indicates goal G being successfully achieved. Similarly, P_e denotes plan P beginning execution and P_s indicates a successful completion of plan P. As can be seen, the execution trace in figure 2 demonstrates that after goal G is posted, plan P1 executes, goal G1 is posted, plan P3 executes and successfully completes, goal G1 is successfully resolved, goal G2 is posted, plan P6 executes and successfully completes, goal G2 is successfully resolved, plan P1 successfully completes, and goal G is successfully achieved.

G_p $P1_e$ $G1_p$ $P3_e$ $P3_s$ $G1_s$ $G2_p$ $P6_e$ $P6_s$ $G2_s$ $P1_s$ G_s

Figure 2: A typical execution trace t **for an agent** A

Assume that we propose to change plan $P6$ in the above example, an impact analysis technique needs to determine the other plans and/or goals that are potentially affected by the change (i.e. the impact set). The static analysis technique proposed in [2] computes the impact set by considering static (direct and indirect) dependencies between $P6$ and other goals or plans in the agent system. It works under the assumption that a change in $P6$ has potential impact on any node reachable from $P6$ in the goal-plan tree for agent A. Therefore, an impact set of plan $P6$ returned by the static technique in [2] contains all entities in the goal-plan tree in figure 1. This would result in highly inaccurate impact set, as evidenced by the experimental result (i.e. precision of approximately 0.3–0.4). We will now show that our dynamic analysis technique which relies on information from execution traces can predict impact sets that are more accurate than those computed by static analysis.

Our dynamic analysis technique relies on execution traces such as the one in figure 2 rather static goal-plan trees. Given a set of changes, we adapt the PathImpact technique [3] to perform forward and backward walks of a trace to identify the impact set of the changes. The forward walk determines all plans executed and all goals posted after the changed goal/plan, whereas the backward walk identifies plans/goals into which the execution can return. More specifically, for each changed entity E (which can be either a plan or goal) and each occurrence of E_e (if E is a plan) or E_p (if E is a goal), we will do the following. Note that we will illustrate our technique using an example of trace t in figure 2 and a change set {P6} (i.e. only plan P6 is modified).

- In the forward walk, we start from the entity immediately following E_e (if E is a plan) or E_p (if E is a goal), add every plan executed or goal posted into the impact set (i.e. every entity F such that the trace contain an entry F_e or F_p after the occurrence of E_e or E_p), and count the number of unmatched successes. In our example, in the forward walk we start at $P6_s$ and add nothing to the impact set since there is no plan executed or goal posted after P6. We however count 3 unmatched successes (i.e. $G2_s$, $P1_s$, and G_s)
- In the backward walk, we begin from the entity immediately preceding E_e (if E is a plan) or E_p (if E is a goal), and add into the impact set as many unmatched plans or goals as the number of unmatched successes counted in the forward walk. In our example, we add G2, P1, and G to the impact set.
- Add E to the impact set if it is not already there. Therefore, the impact set in our example would be {P6, G2, P1, G}.

The above trace is an example of a typical, successful execution. An agent's execution may however contain parallelisation (e.g. achieving two goals concurrently), interruption (e.g. suspending an executing plan to deal with higher priority events), and failures handling (e.g. trying alternative plans in pursuing a goal). We can apply the same technique described earlier to determine impact sets from traces derived from those agent behaviours. In practice, there are usually multiple execution traces of an agent system. In this case, we process each single trace and compute the union of the impact sets returned by each execution traces.

3. REFERENCES

[1] R. Arnold and S. Bohner. *Software Change Impact Analysis*. IEEE Computer Society Press, 1996.

[2] H. K. Dam and A. Ghose. Automated change impact analysis for agent systems. In *Proceedings of the 27th IEEE International Conference on Software Maintenance*, ICSM '11, pages 33–42, Washington, DC, USA, 2011. IEEE.

[3] J. Law and G. Rothermel. Whole program path-based dynamic impact analysis. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, pages 308–318, Washington, DC, USA, 2003. IEEE Computer Society.

[4] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In V. R. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multiagent Systems, June 12-14, 1995, San Francisco, California, USA*, pages 312–319. The MIT Press, 1995.

Supporting User-Centric Business Processes with WADE (Extended Abstract)

Federico Bergenti
Università degli Studi di Parma
43124, Parma, Italy
federico.bergenti@unipr.it

Giovanni Caire, Danilo Gotta
Telecom Italia S.p.A.
10148, Torino, Italy
{giovanni.caire,danilo.gotta}@telecomitalia.it

ABSTRACT
In this paper we present the latest developments of *WADE (Workflows and Agents Development Environment)* that provide concrete support for a better realization of the innovative paradigm of *agent-based BPM (Business Process Management)*. We discuss the new functionality that WADE offers to enable the rapid and effective realization of *user-centric business processes*, i.e., business processes that are tightly integrated with the work of users and that are mainly driven by user interactions. Such processes are met frequently in practice and WADE seamlessly accommodates Web and Android users by means of dedicated views.

Categories and Subject Descriptors
I.2.11 **[Computing Methodologies]**: Distributed Artificial Intelligence – multiagent systems, languages and structures, coherence and coordination

General Terms
Management, Design, Languages

Keywords
Agent-based business process management, user-centric business processes, WADE

1. INTRODUCTION
The extensive use of WADE in mission-critical applications (see the concluding section and [1] for some examples) has witnessed the notable importance of user interactions in the scope of workflows. This is not surprising and we acknowledge that the idea of workflows has its origins in the management of the work of people. Nonetheless, we believe that the common approach of treating user interactions as *yet another type of event* does not adequately capture the importance and the high frequency of them.

So called *user-centric workflows* are introduced in WADE version 3.0 as a means to capture workflows that *(i)* frequently need to interact with users, and *(ii)* are mainly intended to gather information and provide feedback to users. WADE now lifts user interactions to a higher level and it provides specific tools and features to manage them effectively. The design guidelines for such a recent development of WADE are as follows:

- The description of the information to provide to users and the related input to acquire from users must be independent of the device that the user is concretely accessing;
- Any element of such a description must be extensible in order to let developers provide more specific descriptions of both input and output information;
- The software application that the user accesses must be replaceable by any custom application, once the communication with the WADE platform is correctly set up; and
- No device is privileged and developers must be able to describe workflows in full generality, if they really want.

From such very generic guidelines we could easily choose the *Model-View-Controller (MVC)* [2] architectural design pattern as the coarse grained model around which we designed the new *interactivity package* of WADE. The new WADE interactivity package provides the Java classes of the model of interactions (see Section 2.1) and a number of *visualizers* (see Section 2.2) intended to be integrated in the application shipped to users.

2. WADE USER-CENTRIC WORKFLOWS
In order to fully exploit the power of user-centric business processes, the developer of a workflow should first inform WADE that the workflow itself needs to interact with users. This is accomplished by realizing a workflow class that extends the `InteractiveWorkflow` class rather than the common `Workflow` class. Such an `InteractiveWorkflow` class is a specific subclass of `Workflow` that provides the needed machinery to link a workflow instance to a visualizer. WADE ensures a one-to-one correspondence between a user and an instance of an `InteractiveWorkflow`, and therefore an `InteractiveWorkflow` has just one user at a time.

When an `InteractiveWorkflow` is connected to a visualizer, it is requested to provide the visualizer with a description of the information to present to the user and with a related description of the possible user inputs. Such a mechanism is concretely driven by the workflow developer who can freely use the new method `interact()` that `InteractiveWorkflow` provides. Such a method is supplied with an `Interaction` object that contains the following parts: *(i)* an abstract description of the information to be presented to the user with some abstract requirements on the way information is presented, e.g., by indicating how a set of

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4–8 June 2012, Valencia, Spain.

labels should be aligned on the device screen; *(ii)* an abstract description of the information that the user is allowed to return in a response; *(iii)* an abstract description of the constraints that the user response must meet to be considered valid; and *(iv)* a list of possible abstract actions that the user is allowed to choose as valid responses.

Upon executing the `interact()` method, the workflows is put into a *SUSPENDED* state to allow the corresponding visualizer to present the information to the user and to enable the user to provide feedback by means of one of the available response actions. The visualizer is on duty for showing the information in the best possible way and for allowing the user to provide its response. The visualizer is also responsible for the correctness of the provided response because it is in charge of checking the constraints that identify valid responses.

Once the user has validly compiled its response and chosen one of the available response actions, the visualizer returns user response to the workflow instance in terms of a copy of the original `Interaction` object that now contains relevant user input and from which the developer can extract the user response easily. Such an approach allows developers retrieving response information from where they originally decided they should be contained. Moreover, it ensures no redundant information is sent back in responses.

2.1 A Model of Interactions

In the WADE nomenclature, an interaction is both an abstract description of the information to be provided to users and a means to allow users constructing responses. Therefore, WADE provides a set of Java classes that are used to describe interactions with such a dual meaning. Such classes are designed using the standard approach adopted in modern user interfaces and they are structured in a containment tree. They are divided into the following major groups:

- Passive elements, e.g., labels and pictures, that are leafs of the containment tree intended to describe the information to be provided to users;
- Information elements, e.g., text areas and menus of various types, that are leafs of the containment tree and that are meant to provide the user with a means to provide responses;
- Containers, e.g., list and grid panels, that are designed to aggregate a group of children in order to describe their relative position in an abstract manner;
- Actions, that describe the types of responses the user can select; and
- Constraints, that concretely provide check procedures to ensure the correctness of responses.

With the notable exception of constraints, all such Java classes are purely descriptive and they are simple containers for information flowing between an `InteractiveWorkflow` and a visualizer. They are designed to maintain the clear separation of concerns of the MVC design pattern.

All such classes describe the model of an interaction, while the relative controller is implemented by the adopted visualizer, which also generates on the fly the relative view. Such an approach ensures, among other things, that developers are free to add new visualizers and that no visualizer is privileged.

Constraints are peculiar in the scope of the MVC pattern because they are intended to validate input. They represent a pluggable part of the controller because they are responsible for updating the view upon changes in the model, e.g., by marking invalid components with an error notification. WADE provides a set of general purpose constraints that can be used, e.g., to make sure a mandatory menu has at least one item selected or to warrantee that the text in a text field conforms to a given regular expression.

2.2 Available Visualizers

At the time of writing WADE provides two visualizers meant to accommodate two important classes of users: Web users and Android users. Web users are allowed to activate new interactive workflows and to connect to suspended workflows by means of a dedicated visualizer developed using the ZK toolkit [4]. ZK is a very popular toolkit to develop AJAX applications in Java and it is easily interfaced with WADE. The ZK visualizer instantiates one JADE agent on the server side of the Web application for each and every Web session, and it ensures agents are properly connected with the WADE platform. The client side of the ZK application is meant to: *(i)* present information to the user; *(ii)* provide selectable actions in terms of buttons; and *(iii)* ensure constraints are met before passing any response to workflow agents. The chosen approach ensures a lightweight client that is only in charge of realizing the user interface on the fly and of validating constraints. ZK provides a proprietary communication mechanism between the client browser and the server side of the application which is completely hidden in the deep internals of ZK, thus becoming transparent to developers.

The Android visualizer is developed along the lines of the ZK visualizer and we ensured that the internals of the two visualizers are designed using the same architecture and adopting closely related approaches. The major difference with the ZK visualizer is that the Android visualizer is a single Android application that hosts: *(i)* a JADE container in *split mode* (see JADE documentation for details [3]) which is created in the scope of the WADE platform; *(ii)* the agent needed to connect the user with the workflow; and *(iii)* the visual components that are used to dynamically assemble and render the user interface. No proprietary communication mechanism is needed in this case because the agent and the visual components share some memory of the Android device.

3. REFERENCES

[1] Caire, G., Gotta, D., and Banzi, M. 2008. WADE: A Software Platform to Develop Mission Critical Applications Exploiting Agents and Workflows. In *Proc. 7th Int'l Conf. Autonomous Agents and Multiagent Systems*, 29-36.

[2] Fowler, M. 2003. *Patterns of Enterprise Application Architecture*, Addison-Wesley.

[3] JADE – Java Agent Development framework. Available at http://jade.tilab.com

[4] ZK. Available at http://www.zkoss.org

OrgMAP: An Organization-based Approach for Multi-Agent Programming
(Extended Abstract)

Cuiyun Hu
School of Computer, National University of Defense Technology, China 410073
hcy56316@163.com

Xinjun Mao
School of Computer, National University of Defense Technology, China 410073
mao.xinjun@gmail.com

YinChen
School of Computer, National University of Defense Technology, China 410073
yinchen@163.com

Huiping Zhou
School of Computer, National University of Defense Technology, China 410073
icent@qq.com

ABSTRACT
This paper proposes a new organization-based multi-agent programming (OrgMAP) approach to constructing dynamic and flexible software systems. A computational and programming model named Oragent is defined following software engineering principles such as modularity, reusability and etc. Oragent model not only allows programmers to represent the systems with high-level abstractions in terms of organizations, rules, protocols and roles, but also provides a number of mechanisms, such as encapsulation, inheritance, enactment and event, to improve the dynamics and flexibility of MAS.

Categories and Subject Descriptors
D.3 [**Programming Language**]: Miscellaneous

General Terms
Languages, Theory

Keywords
MAS programming, Organization theory, Organization-oriented programming

1. INTRODUCTION
With organizational concepts, agent-oriented software engineering (AOSE) provides a natural representation for complex software systems. Recently a variety of organization-oriented approaches to multi-agent systems engineering have been brought forth including modeling approaches, methodologies, infrastructures and programming languages[1][2][3].While organization metaphor has made significant contributions to analysis and design multi-agent system (MAS), when it comes to implementation, the fact is that current MASs are usually implemented as a set of agents in terms of mental concepts, where information about organization structure and collective behavior is lost [4][5]. As a result, programmers have to manually translate and incorporate the organizational concept from design model to metal concepts, which leads to poor engineering practice and hinders engineers to exploit the full potentials of AOSE.

A recent trend in the AOSE is to employ organization concepts in the implementation of MAS [2][3]. However, until now, few languages have explicitly provided primitives for the organizational concepts. Current researches in this field usually focus on the normative MAS with the aims to deal with the openness and heterogeneity [1], but inadequate in handling dynamics and flexibility. This paper proposes a new organization-based multi-agent programming (OrgMAP) approach with a computational and programming model named Oragent, which allows programmers to constructs the systems with first-class organizational concepts, such as organizations, roles, protocols and rules. In addition, mechanisms supporting dynamics and flexibility are proposed.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2. ORGANIZATION-BASED MULTI-AGENT PROGRAMMING
2.1 Oragent Models

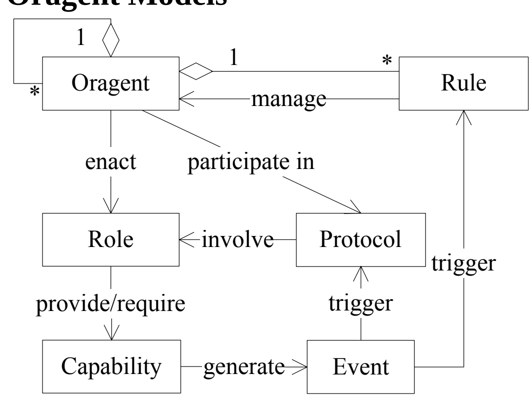


Figure 1. Oragent computational model.

Figure 1 shows the computational model of an oragent. Oragents are self-managed entities that can be either collective agency or individual agents. In the oragent model, an OrgMAP application is modeled with oragents that interact with each other with protocols based on roles. Each oragent can have a set of rules to manage its structure and behavior, enacting roles that enable it to interact with others base on protocols and have some member oragents. The capabilities of an oragent are organized with a delegation mechanism, i.e. an oragent can delegate the implementation of its capabilities to its members. The execution of capabilities will generate a set of events which can trigger the protocols or rules. The protocols enable the involved role players (i.e. oragents) to interact with each other. With the rules, each oragent can manage the transformation of the roles it enacts and can also reorganize its structure.

An Oragent programming model is defined to describe the main programming facilities. Firstly, a set of oragents with the same structure and behavior can be specified as either an organization (for composite agency) or a role (for individual agent). Organizations are composed of four parts: state variable declarations, roles, interaction protocols and management rules. Secondly, each role defines the capabilities that contribute to its organization. It can either declare the capabilities required from its players or define the capabilities provided to its players. The roles with required capabilities are considered as abstract roles, and the organizations that enact them should provide the required capabilities. Thirdly, protocols describe interaction patterns as a sequence of messages based on roles. Each protocol can subscribe the events that are published by its involved roles. Finally, the management rules in an organization define how an oragent reorganize its structure based on the event from its members or context. In additions adaptation rules are defined within roles to describe how the oragent transfers its roles on events.

2.2 Mechanisms

This section introduces several programming mechanisms to support the construction and execution of an Oragent program, including encapsulation, inheritance, enactment and event.

Encapsulation is an important mechanism for information hiding in programming. To support the self-management, oragents encapsulate state, actions, behavior and management, so that they are able to decide how to construct and regulate their structures.

Inheritance is an important mechanism in OO programming for reuse. Similarly, the Oragent model also supports inheritance among roles, so that the sub-roles have the properties and capabilities of their super-roles. Furthermore, in the Oragent model if the roles have a common super-role, they are said to be sibling. The sibling roles are allowed to transfer from one to another according to the rules defined within their common super-role at run-time. Therefore, not only the properties and capabilities can be reused between a role and its sub-roles, but the state of oragents can also be reused among sibling roles.

Enactment is a new mechanism introduced in AOP by Mehdi Dastani in [3] with the aim to capture role dynamics. However, in the Oragent model, enactment mechanism describes the relationship between oragents and roles, contrast to the instantiation mechanism in OO. While an object has to adhere to one class that cannot be changed once it is instantiated, an oragent can possess multiple roles that can be changed dynamically during its lifetime. With the enactment mechanism, the state of the oragents can be reused during the role transformation.

Event mechanism enables the self-management of oragents. In Oragent model, each oragent owns an event management center that allows its content and context roles publish and subscribe events and a logic reasoning engine that takes as input the published events, together with programmer-defined management rules that map each event to the corresponding adaptation or management behavior. The output, then, corresponds to the published event provided with a sequence of actions including enacting, deacting, activating and deactivating of a given role and initiation, termination and regulation of a given protocol.

3. CONCLUSIONS AND FUTURE WORK

This paper proposes a new programming approach—OrgMAP and the core computational and programming model is provided as Oragent model. The Oragent model allows programmers to explicitly represent the structure of the system with high-level organizational abstraction so that the gap between design and implementation is bridged. Moreover, a set of mechanisms such as encapsulation, inheritance, enactment and event, are provided to facility the development of dynamic systems. Finally, from the software engineering perspectives, the Oragent model provides more high-level reusable entities, such as organizations, roles.

In order to implement the Oragent model, we are currently working on the programming language adhering to the Oragent model as well as a platform to execute the Oragent programs. Moreover, a programming methodology will be designed to guide the developers to code and deploy the OrgMAP programs.

4. ACKNOWLEDGMENTS

Acknowledge the financial support from Natural Science Foundation of China under granted number 61070034, 91024030 and 61133001, Program for New Century Excellent Talents in University, National Grand Fundamental Research 973 Program of China under granted number 2011CB302601.

5. REFERENCES

[1] Tinnemeier, N.A.M. 2011. Organizing Agent Organizations: Syntax and Operational Semantics of an Organization-Oriented Programming Language. SIKS Dissertation Series 2011(2), Utrecht University.

[2] Olivier Boissier, Jomi Fred Hübner, and Jaime Simão Sichman. 2007. Organization Oriented Programming: From Closed to Open Organizations G. In Proceeding of ESAW 2006. Springer Press, Heidelberg, 86–105.

[3] Dastani, M.M., van Riemsdijk, M.B., Hulstijn, J., Dignum, F.P.M., Meyer, J.-J.C. 2005. Enacting and deacting roles in agent programming. In Proceedings of AOSE 2004. Springer Press, Heidelberg, 189-204.

[4] Rafael H. Bordini, Mehdi Dastani, and Michael Winikoff. 2007. Current Issues in Multi-Agent Systems Development. In Proceedings of the Seventh Annual International Workshop on Engineering Societies in the Agents World. ESAW '07. 38-61.

[5] Nick Tinnemeier, Mehdi Dastani, John-Jules Meyer. 2009. Roles and Norms for Programming Agent Organization. In Proceedings of AAMAS 2009. 121-128.

MAPLE: Multi-Agent Programming with Letter Exchanges on Sensor Networks

(Extended Abstract)

Tiffany Yi-Ting Tsao
Department of Computer Science and Information Engineering
Nation Taiwan University
yiting.tsao@gmail.com

Wan-rong Jih
Department of Computer Science and Information Engineering
Nation Taiwan University
wanrong.jih@gmail.com

Jane Yung-jen Hsu
Department of Computer Science and Information Engineering
Nation Taiwan University
yjhsu@csie.ntu.edu.tw

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent system*

General Terms

Design

Keywords

multi-agent system, sensor network, service composition

1. INTRODUCTION

Developing reusable and flexible sensor networks is an important task for context-aware applications. Previous researchers have deployed several applications on sensor networks [3, 5]. These sensor networks are consisted of various sensor nodes to monitor and control the environment. Since these sensor networks are consisted of various sensor nodes, developers need to take sensor deployment and application design into consideration at the same time to achieve their goals. It is a great challenge for developers to deal with low-level sensor controls while designing policies to interact with users. The flexibility of sensor networks is inherently limited by its design. Therefore, to make developers focus on application design without concern for any hardware issues is very important.

Some researchers have proposed the middleware solution to improve the flexibility of sensor network [4, 7]. By building a virtual machine on top of each sensor node, it is easy for developers to program the nodes using a predefined instruction set. The provided instruction set is an assembly-like language which brings limited assistance to developers. OASiS is a programming framework for middleware solution [6]. It treats the unit of application functionality as a service so that developers can compose the services to achieve their goals. However, the services in this framework still need to be redesigned when the application changes.

In order to improve the reusability of sensor networks, this paper proposes *MAPLE (Multi-Agent Programming with Letter Exchanges)*, a multi-agent architecture, to manage dynamic sensor networks. Each agent in MAPLE provides specific service, such as sensor network maintenance, environment perception, device control, service discovery, service composition, and context reasoning. Each service corresponds to an instruction. Then, the agents can easily use the services by wrapping the instructions as letters and sending them to others.

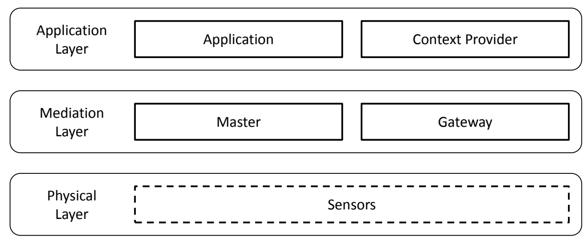


Figure 1: MAPLE: a multi-agent sensor network architecture. The rectangles with solid borders are agents.

2. DESIGN

The MAPLE architecture includes three layers (see Figure 1). Rectangles with solid borders represent the agents. All practical sensors with different abilities are used in the physical layer. The mediation layer is a communication interface to simplify the procedure for fetching sensor data and controlling actuators. In the application layer, the agents interact with end users and provide high-level services.

Gateway

Gateways are the managers that supervises sensors with the following instructions.

INSERT adds a sensor profile to an unused port when a new sensor is attached. For example, a user plugs in a motion sensor to the analog port 3, places it on the desk, and sets its sensitivity level to 10.

INSERT PORT="A.3" TYPE="MOTION" SENSITIVITY="10" LOCATION="DESK"

UPDATE refreshes the profiles when the physical deploy-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

ment changes. For example, a user replaces the motion sensor on the analog port 3 with a sound sensor and all other settings remain the same.

UPDATE PORT="A.3" TYPE="SOUND" SENSITIVITY="10" LOCATION="DESK"

DELETE removes the stale profiles when sensors are broken accidentally. For example, a dog bites the sound sensor on the analog port 3 and then the broken sensor will be removed.

DELETE PORT="A.3" TYPE="SOUND" SENSITIVITY="10" LOCATION="DESK"

READ acquires electrical signals from sensors. For example, a user retrieves the data from the analog port 3 every 5 seconds.

READ PORT="A.3" INTERVAL="5000" RECEIVER="USER"

WRITE controls the interaction with actuators in the real world. For example, a user sends a signal to the digital port 0, and then the corresponding device will respond immediately.

WRITE PORT="D.0" INTERVAL="-1" RECEIVER="Tester"

Master

Master provides two services, SELECT and SET, to search and compose the other services on the same sensor network.

SELECT searches available services that fit the given constraints. For example, retrieving the agents with a motion sensor in the living room.

SELECT NAME="MOTION" LOCATION="LIVING ROOM"

SET encapsulates the steps for achieving a given context or gathering context information. For example, gathering the luminance state on the desk every 5 seconds.

SET GOAL="LUMINANCE" INTERVAL="5000" LOCATION="DESK"

Context Provider

Each context provider infers contexts from the sensor data. For example, the motion level provider infers the status from the given motion data.

INTERPRET MOTION="400"

Application

Each application provides reminder or assistance to users. For instance, the auto lighting application controls the lamp on the ceiling according to the motion level at the desk.

SUBSCRIBE LOCATION="CEILING,DESK"

3. IMPLEMENTATION

To demonstrate the usability of the MAPLE framework, we build an application to lighting a lamp when the motion level in the office is high. Several different sensors are deployed in a personal office. We integrate PL-PLAN [2] planner into the JADE [1] platform to control sensors and provide the instructions/services. In line 1- 2, the application requests the motion level and the lamp state. After the application makes a decision according to its build-in rules, it sends a request to control the lamp in line 7.

Algorithm 1 The behavior of the auto lighting application

```
1: SET GOAL="MOTIONLEVEL" INTERVAL="1000" LOCATION="DESK"
2: SET GOAL="LAMPSTATE" INTERVAL="1000" LOCATION="CEILING"
3: while true do
4:    // Receive the contexts
      ...
5:    // Make a decision
6:    if MOTIONLEVEL = HIGH and LAMPSTATE = OFF then
7:       SET GOAL="LAMP" INTERVAL="-1" LOCATION="CEILING"
8:    end if
9: end while
```

4. CONCLUSION

Sensor network programming is a very challenging task because application developers need to design not only the sensor deployment but also their control strategies. In this paper, we propose the MAPLE framework to simplify the programming effort. It defines different agent roles to provide a variety of services in sensor networks, such as sensor management and context inference. Therefore, application developers can leverage the available services to design applications without worrying about the hardware. To summarize, MAPLE can not only manage dynamic sensor networks but also help application developers reuse the services in sensor networks.

5. ACKNOWLEDGMENTS

This work was also supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC99-2911-I-002-201, NSC 100-2911-I-002-001, 10R70500, and 10R70501.

6. REFERENCES

[1] JADE - Java Agent DEvelopment Framework. http://jade.tilab.com/.

[2] PL-PLAN. http://plplan.philippe-fournier-viger.com/.

[3] D. Christin, P. Mogre, and M. Hollick. Survey on Wireless Sensor Network Technologies for Industrial Automation: The Security and Quality of Service Perspectives. *Future Internet*, 2(2):96–125, 2010.

[4] C. Fok, G. Roman, and C. Lu. Agilla: A Mobile Agent Middleware for Self-Adaptive Wireless Sensor Networks. *ACM Transactions on Autonomous and Adaptive Systems*, 4(3):1–26, 2009.

[5] B. Kim, S. Hong, Y. Jeong, and D. Eom. The Study of Applying Sensor Networks to a Smart Home. In *Proceedings of the International Conference on Networked Computing and Advanced Information Management*, 2008.

[6] M. Kushwaha, I. Amundson, X. Koutsoukos, S. Neema, and J. Sztipanovits. OASiS: A programming framework for service-oriented sensor networks. In *Proceedings of the International Conference on Communication System Software and Middleware*, 2007.

[7] P. Levis and D. Culler. Maté: A Tiny Virtual Machine for Sensor Networks. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2002.

Efficient Context Free Parsing of Multi-agent Activities for Team and Plan Recognition

(Extended Abstract)

Bikramjit Banerjee
School of Computing
The University of Southern Mississippi
Hattiesburg, MS 39406
Bikramjit.Banerjee@usm.edu

Jeremy Lyle
Dept. of Mathematics
The University of Southern Mississippi
Hattiesburg, MS 39406
Samuel.Lyle@usm.edu

Landon Kraemer
School of Computing
The University of Southern Mississippi
Hattiesburg, MS 39406
Landon.Kraemer@eagles.usm.edu

ABSTRACT

We extend a recent formalization of multi-agent plan recognition (MAPR), to accommodate compact multi-agent plan libraries in the form of context free grammars (CFG), incomplete plan executions, and uncertainty in the observation trace. Some existing approaches for single agent plan recognition cast it as a problem of parsing a single agent activity trace. With the help of our multi-agent CFG, we do the same for MAPR. However, known hardness results from multi-agent plan recognition constrain our options for efficient parsing, but we claim that *static* teams are a necessary (though not sufficient) condition for polynomial parsing. The necessity is supported by the fact that MAPR becomes NP-complete when teams can change dynamically. For sufficiency, we impose additional restrictions and claim that if the social structure among the agents is of certain types, then polynomial time parsing is possible.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory, Performance

Keywords

Plan recognition, Modeling other agents

1. INTRODUCTION

Multi-agent plan recognition (MAPR) refers to the problem of explaining the observed behavior of multiple agents by identifying the (dynamic) team-structures and the team plans (based on a given plan library) being executed, as well as predicting their future behavior. Applications of MAPR range from monitoring and surveillance, to automated sports commentary, to assistive technologies. Recently, Banerjee et. al. introduced a formal model for MAPR and used it to investigate the complexity of its simplest setting [1]. Zhuo and Li advanced this model to address missing observations in the activity traces as well as incompleteness of the plan library [5]. However, these models assume that the plan library is presented in an uncompact and non-hierarchical form, in particular as a set of team plans where each plan is a matrix of a fixed number of steps. Moreover, these models do not handle uncertainty in the observation trace in a general manner.

We have refined existing models from [1, 5] to make three important generalizations: allow compact, hierarchical, non-trivial plan libraries that correspond to infinite languages as opposed to the finite language in [1, 5], allow incomplete plan executions, and allow traces to be uncertain. Typically for plan recognition with single agents, a plan library is given in a compact hierarchical form, such as an HTN [2]. We have developed polynomial-time algorithms for a less expressive plan library, viz., context free grammars (CFGs) which incorporates some desirable features of HTN, e.g., recursiveness and hierarchies. This advances previous formalization in [1, 5] which accommodated none of these desirable features. Results from single agent plan recognition have shown that as long as partial ordering of plan steps is not allowed in the grammar, activity strings can be parsed in polynomial time [4]. However, even with the multi-agent CFG (i.e., with no partial ordering), MAPR would still be hard unless additional constraints are imposed. We identify specific types of social structures with static teams as such constraints.

In [5], missing steps in the trace as well as in the plan library were allowed. In constrast, assuming Σ to be the set of all possible observable activities, we model a missing observation as a uniform distribution over Σ to enunciate complete uncertainty. Unlike [5], this also allows for a varying degree of uncertainty on other observations that are not missed. A missing step in a plan is modeled as a don't care ($*$), similar to [5]. We include *noop* $\in \Sigma$ (i.e., "no operation") for cases when an agent is idling, which is often required for coordination among teammates. The CFG plan library is constructed in the same manner as [3, 4], except that each terminal activity represents a *vector* of activitites for the members of a *team* rather than a single agent.

2. ILLUSTRATIVE EXAMPLE

Figure 1 shows an illustrative example of the MAPR problem. The input is a (n =3)-agent trace, $\mathcal{T}$, that shows their

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Input trace

Steps	Agent activities		
	Agent 1	Agent 2	Agent 3
1	guard	collect	threaten
2	guard	collect	threaten
3	guard	collect	threaten
4	ride	drive	ride
5	ride	drive	ride
6	ride	drive	ride

Plan library

Plan: P_1^2 (2-agent Money Pickup)

$P_1^2 \rightarrow \langle collect, guard\rangle Q_1^2$
$Q_1^2 \rightarrow \langle collect, guard\rangle Q_1^2 | \langle drive, ride\rangle Q_2^2$
$Q_2^2 \rightarrow \langle drive, ride\rangle Q_2^2 | \epsilon$

Plan: P_1^3 (3-agent Bank Robbery)

$P_1^3 \rightarrow \langle collect, guard, threaten\rangle Q_1^3$
$Q_1^3 \rightarrow \langle collect, guard, threaten\rangle Q_1^3 | \langle drive, ride, ride\rangle Q_2^3$
$Q_2^3 \rightarrow \langle drive, ride, ride\rangle Q_2^3 | \epsilon$

Figure 1: Illustrative example of MAPR with a CFG plan library.

recognized activities for ($T = 6$) steps, where n is the number of agents and T is the observation horizon. Suppose

$$\Sigma = \{collect, guard, threaten, drive, ride, noop\},$$

and each activity in $\mathcal{T}$ is associated with likelihood $1 - \delta$, with the probability that each could be some other activity in Σ being $\delta/5$, for some small $\delta > 0$. The input also contains the CFG shown in Figure 1(right) as the plan library.

Given the CFG plan library and the trace, the activities of agents 2 and 1 (in that order) could be parsed as fitting plan P_1^2 (2-agent money pickup in an armored car) with a high probability. However, activities of agents 2, 1, and 3 (in that order) might also be parsed as fitting plan P_1^3 (3-agent bank robbery) with a high probability. As in [1, 5], this ambiguity is resolved by noting that if the first hypothesis is accepted then it would be difficult to explain the activity of the remaining agent 3, and any explanation (provided there are other plans in the library that can explain agent 3's actions) could have a very low probability. In other words, the trace does not *partition* well. On the other hand, accepting the second hypothesis explains all agents' activities with a high probability. This also illustrates the power of partitioning the trace. Since money pickup is a more commonly observed plan, it has a high prior likelihood compared to bank robbery. Thus, if partitioning was not required and we were allowed to leave some activities unexplained, then bank robbery would be consistently missed.

3. MAIN CONTRIBUTIONS

Due to space limitation, we highlight the main contributions of our work.

- We have formally extended the definition of an *occurrence* [1, 5] to account for plans whose executions have not been completed by the observation horizon T. Such occurrences are called *partial* occurrences. As a consequence of the partiality of occurrences, we have also adapted the formal definitions of notions of conflict of (complete/partial) occurrences (i.e., hypotheses competing to explain some observations), and the definition of MAPR as a partition of the trace ($\mathcal{T}$) into complete or partial occurrences such that partial occurrences can only end at step T. In order for such a solution concept to apply, we have extended the closed world assumptions of [3] for multi-agent systems enunciating that every team-member of a team plan are observed.
- We have specified an algorithm (Parser) that extends Villain's Earley parser, for the determination of the maximum likelihood parse of a certain set of columns of the trace, $\mathcal{T}$. That is, given a team hypothesis, the parser yields the most likely sequence of high-level plans that the team might be executing. This algorithm has complexity $O(s^{2.5}G^2t^3)$, where s is the size of the hypothesized team, G is the size of the grammar, and t is the number of steps explained.
- We have revisited the notion of social structures that was used in the past to facilitate team hypotheses formation for MAPR. Instead of the hierarchical decomposition of teams into subteams, as done in the past, our notion of social structures captures the actual hierarchical organization of the agents. We allow teams to form only along paths in the social structure graph.
- Using Parser as a subroutine, we have formally proved that MAPR can be solved in polynomial time when the social structure graph is a star, a path, or a tree of bounded depth, and the teams are static.
- We have formally proved that when the team structures can vary with time, then MAPR is NP-complete even when the social stucture graph is as simple as a path. This proof is based on a reduction from the rectangular tiling problem (RTILE). Together with the previous result, this means that the staticity of teams is a necessary, but insufficient condition for the polynomial solvability of MAPR. Furthermore, social structure graphs such as star, path and trees, impose additional structure that turn out to be sufficient for polynomial solvability, in conjunction with static teams.

Acknowledgment: This work was supported in part by a start-up grant from the University of Southern Mississippi.

4. REFERENCES

[1] B. Banerjee, L. Kraemer, and J. Lyle. Multi-agent plan recognition: Formalization and algorithms. In *Proceedings of AAAI-10*, pages 1059–1064, Atlanta, GA, 2010.

[2] K. Erol, J. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In *Proc. AAAI-94*, pages 1123–1128. AAAI Press, 1994.

[3] H. A. Kautz and J. F. Allen. Generalized plan recognition. In *Proc. AAAI*, 1986.

[4] M. Vilain. Getting serious about parsing plans: a grammatical analysis of plan recognition. In *Proc. of AAAI-90*, 1990.

[5] H. H. Zhuo and L. Li. Multi-agent plan recognition with partial team traces and plan libraries. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 484–489, 2011.

Agent Deliberation via Forward and Backward chaining in Linear Logic

(Extended Abstract)

Luke Trodd, James Harland, John Thangarajah
School of CS & IT, RMIT University, Melbourne, Australia
{luke.trodd, james.harland, john.thangarajah}@rmit.edu.au

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*intelligent agents, languages and structures*

General Terms

Languages, Design

Keywords

BDI Deliberation cycle, linear logic, forward chaining, Lygon

Agent solutions to programming problems are often based on the *Belief-Desire-Intention (BDI)* paradigm [12]. *Beliefs* represent what the agent believes to be the current state of the world. *Desires* specify the proactive behaviour of the agent, in that the agent works to make these true. Often desires can be mutually exclusive or contradictory, requiring the agent to select from among them, and so BDI implementations often use *goals*, which can be thought of as desires with some restrictions on them (such as requiring goals to be consistent, feasible and not yet achieved). There can be several types of goals, including *achievement goals*, whcih are dropped once they have been achieved, and *maintenance goals*, which are continually monitored, even when currenlty true. *Intentions* are plans of action that the agent has committed to to achieve its current goals. Often there are many ways to achieve a set of goals that the agent is working on, implying the need for a mechanism to choose between them.

Implementations of BDI systems are usually based around an *observe-think-act* cycle, in which an agent will observe the current environment, which may have changed since the last observation, determine which goals it should be pursuing and what plans should be used to achieve them, and choose a particular action to perform. Note that while the number of actions performed in the *act* phase is not specified, it is intended to be relatively small, so that the agent will be able to detect changes in the environment (which is only done in the *observe* phase) and respond to them within an appropriate amount of time. Hence a fundamental feature of BDI systems is the manner in which they provide both proactive (or goal-directed) and reactive behaviour.

In this paper, we consider how we may adapt existing logical inference techniques to implement a BDI architecture. Using logic as a basis for the architecture will mean that we can develop methods for formal analysis of agent systems via logical inference, as well as being able to exploit existing automated reasoning technologies to develop applications. In particular, we will investigate the use of *linear logic* [4] for such systems. Linear logic has the potential to offer many advantages in the agent context over other logics due to its resource-oriented nature. Linear logic is able to specify actions cleanly and intuitively [9], can effectively express resource oriented problems and has a native notion of concurrency appropriate for agent architectures. Linear logic has also been recently applied to agent negotiation [11], and adaptive narratives [1]. This suggests that there is significant potential for the development of BDI agents based on linear logic, for which there are existing logic programming languages such as *Lygon* [5].

Our BDI agent architecture is based on Lygon technology. This means that we proceed in a bottom-up manner, commencing with what can be readily implemented in Lygon, identifying where extensions are needed, adding these to Lygon and eventually developing a BDI deliberation cycle. This has been implemented and applied to various problems (including the gold mining problem used in the CLIMA agent programming contest[1]). Our focus is hence not so much on the design of (yet another) agent programming language, nor on the formal analysis of such a language, but on the similarities and differences between what is provided in linear logic programming languages such as Lygon and what is required by a BDI agent architecture. Once this is done, we intend to use our implementation experience to develop both appropriate language features and a formal analysis of their properties.

We have developed and implemented a forward-chaining inference mechanism to complement Lygon's existing backward-chaining, in order to provide a natural mechanism for reactive behaviour. This combination turns out to be a simple but effective technique for proactive checking of maintenance goals [3] in a generic manner. We have implemented these techniques in Lygon and have developed and tested a number of applications.

The requirement for an agent to use a combination of both proactive and reactive behaviour corresponds in automated reasoning to a combination of both *backward-chaining* and *forward-chaining* inference [6, 2, 8]. Backward-chaining involves reasoning backwards from a goal towards known truths, whereas forward-chaining involves using what is known to be true to infer new results. Harland and Winikoff [7] have proposed a BDI system based on linear logic, in which the proactive behaviour of the agent is provided by backward-chaining methods and the reactive behaviour of the agent is provided by forward-chaining methods. In terms of the BDI cycle, this means that the *think* phase would be implemented by backward-chaining techniques and the *act* and *observe* phases by forward-chaining ones. Backward-chaining methods have been the basis of logic programming languages based on linear logic, such

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1] http://centria.di.fct.unl.pt/~clima

as Lygon [5]. Forward-chaining methods have also been used[6], and techniques to combine both methods into one system have also been studied [2, 8]. However, there has been comparatively little work on applying such methods in linear logic to agent systems. In addition, the work of Harland and Winikoff was purely a design; no precise execution method was given and no implementation was developed.

Our first step is to extend Lygon with abductive capabilities, so that the result of a computation is not just an answer, but is a set of actions to be performed (possibly empty, corresponding to a "yes") in order to make the goal true. The *abducibles*, i.e. the results of the abduction process, are constrained here to be actions.

We write actions and plans as rules in Lygon, and use backward-chaining together with abduction to determine a set of actions that will achieve the goal. A subtlety here that may not be immediately apparent is that *there is a need to specify sequences of goals*, i.e. goals and actions that must be performed in a particular order. For example, a robot vacuum cleaner that needs to move to a particular room before cleaning it will require the move action to be done before the cleaning one, as a post-condition of the move action is a pre-condition of the next action. Moreover, it is common for plans to require that a particular set of actions be performed in a specific order, sometimes intermixed with subgoals. This means that to implement a BDI-style system, we need to be able to specify a sequential order in which actions, plans and goals are to be executed or achieved. This is nothing more or less than a reflection of the fact that the actions required to achieve a particular goal are usually constrained to work in a particular sequence.

Unfortunately there is no (simple) way to use existing Lygon connectives to do this. One possibility is to use *, which does something related, but as discussed by Winikoff [13], this does not work, as * can only distribute existing resources. Given a goal $G_1 * G_2$ any new information generated in the solution of G_1 (and in particular the postconditions of an executed action) cannot be passed onto G_2. Using $G_1 \# G_2$ does allow this, but does not restrict the computation of G_1 to be performed before G_2 (and in fact allows both goals to be pursued concurrently). Another possibility is to use the "continuation-passing style" mechanism proposed by Winikoff, which adds a continuation argument to each rule, and splits each rule into a number of rules. However, this is unwieldy, and the number of rules can potentially grow very large and hence difficult to maintain, especially due to the recursive nesting of rules that is required.

Hence we introduce a new connective $>>$ (read 'then'), in order to succinctly state what is required. Intuitively, an agent wanting to sequentially achieve goals G_1 and G_2 will first perform actions to achieve G_1, and, having noted the updates to the world that these actions have made, make plans for achieving G_2 from that updated world. Hence a program and goal $P, G_1 >> G_2$ results in the program and goal P_1, G_2 where P_1 is the result of actions A_1 which convert P to P_1 and for which $P_1 \vdash G_1$.

The $>>$ mechanism makes it straightforward to specify agent behaviours. It also seems intuitively simple, although it in some ways combines both forward- and backward-chaining. Consider a program P_0 and the goal $G_1 >> G_2 >> \ldots >> G_n$. This asks the agent system to find, if possible, actions $A_1, A_2 \ldots A_n$ such that $P_{i-1} \xmapsto{A_i} P_i$ (i.e. the actions A_i will convert P_{i-1} to P_i) and $P_i \vdash G_i$. If at any point, such an A_i cannot be found, backtracking occurs to see if some alternatives can be found for earlier goals (meaning that there can be many such A_i for each G_i). In other words, solving for each goal G_i results in a backward-chaining computation to find A_i, and the results of each action are propagated forwards to the next goal.

The mechanisms that have been discussed in this paper have been implemented in an extended version of Lygon. Our extensions to Lygon have added around 1100 lines (of sparsely arranged and duly commented code) to the original Lygon interpreter of 720 lines.

We believe that our experiments show that this approach has been an effective way to develop BDI agents. One of the more pleasing artefacts of the implemented agent extensions was the relatively straightforward means by which proactive constraints could be implemented. Proactive constraints provide an extremely powerful mechanism for arbitrarily restricting agent behaviours in an intuitive way. The constraint mechanism effectively implements many of the ideals proposed by Duff at al. [3] for proactive maintenance goals in an agent context.

For future work, the precise relationship between the $>>$ operator and the increasingly sophisticated proof-theoretic combinations of backward- and forward-chaining [2, 8] requires further investigation. The definition of the $>>$ operator itself is in some sense orthogonal to the issues of backward- and forward-chaining, but the way in which it is used in agent programs seems to imply that further analysis will be rewarding. Given that $G_1 >> G_2$ specifies a particular order in which G_1 and G_2 must be used, non-commutative versions of linear logic may be an appropriate starting point [10]. The key technical issue is finding an appropriate interaction between the non-commutative connective $>>$ and the other commutative connectives, as distinct from having only commutative or non-commutative properties alone.

Another aspect of future work is to incorporate maintenance goals into the planning mechanism. This would mean that the generation of actions would also include the possibility to generate actions designed to restore maintenance goals after a predicated violation. Hence rather than just avoid situations where violations occur, the agent can take actions to recover from violations.

1. REFERENCES

[1] A.-G. Bosser, M. Cavazza, and R. Champagnat. Linear logic for non-linear storytelling. In *ECAI*, Lisbon, August 2010.

[2] K. Chaudhuri, F. Pfenning, and G. Price. A logical characterization of forward and backward chaining in the inverse method. *Journal of Automated Reasoning*, 40:133–177, March 2008.

[3] S. Duff, J. Harland, and J. Thangarajah. On proactivity and maintenance goals. In *AAMAS*, pages 1033–1040, Hakodate, 2006.

[4] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[5] J. Harland, D. Pym, and M. Winikoff. Programming in lygon: An overview. In *AMAST*, pages 391–405, Munich, July 1996.

[6] J. Harland, D. Pym, and M. Winikoff. Forward and backward chaining in linear logic. In *CADE-17 Workshop on Proof-Search in Type-Theoretic Systems*, Pittsburgh, June 2000.

[7] J. Harland and M. Winikoff. Agents via mixed-mode computation in linear logic. *Annals of Mathematics and Artificial Intelligence*, 42(1):167–196, 2004.

[8] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logic. *Theoretical Computer Science*, 410(46):4747–4768, 2009.

[9] M. Masseron, C. Tollu, and J. Vauzeilles. Generating plans in linear logic i: Actions as proofs. *Theoretical Computer Science*, 113(2):349–370, 1993.

[10] J. Polakow. Linear logic programming with an ordered context. In *Principles and Practice of Declarative Programming*, Montreal, September 2000.

[11] D. Porello and U. Endriss. Modelling multilateral negotiation in linear logic. In *ECAI*, page 939, Lisbon, August 2010.

[12] A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In *KR*, pages 439–449, 1992.

[13] M. Winikoff. *Logic Programming With Linear Logic*. PhD Thesis, University of Melbourne, 1997.

On the Failure of Game Theoretic Approach for Distributed Deadlock Resolution

(Extended Abstract)

Nadav Sofy
Department of Computer Science
Bar Ilan University
Ramat Gan 52900, Israel

David Sarne
Department of Computer Science
Bar Ilan University
Ramat Gan 52900, Israel

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Performance, Human Factors

Keywords

Deadlock resolution, Equilibrium, Bounded rationality, Modeling the dynamics of MAS

1. INTRODUCTION

Deadlocks may occur in many multi-agent environments and in various contexts. In particular, deadlock is a common problem in multiprocessing where many processes share a specific type of mutually exclusive resource. As such, the problem has received much attention in Operating Systems and Databases literature, resulting in various mechanisms for avoiding, detecting and recovering from deadlock situations. Recent advances in deadlock research extend the deadlock model to distributed environments. Here, deadlocks are harder to manage since none of the participating agents have a full knowledge of the entire system. Consequently, a number of approaches were pursued for handling deadlocks in distributed systems. Still, all these studies assume that agents are cooperative and follow a dictated deadlock resolution protocol.

Nevertheless, in many deadlock situations occurring in multi-agent systems, agents are self-interested and a cooperative resolution scheme cannot be enforced. This situation is also likely to hold in future virtual environments where agents migrate between different platforms, communicating and negotiating with other agents autonomously, without the mediation of the hosting platform. In such environments, deadlocks can be resolved only if an agent willingly retracts from its deadlock-related requirements. The problem becomes even more complex if the agents are not fully rational or are pre-programmed with various deadlock handling logic. In this case, each individual agent needs to be incentivized to comply with the required behavior. Here, the vast deadlock management solutions designed for cooperative and fully rational agents may become irrelevant.

Game-Theory principles can be applied to construct a simple stable distributed solution to the problem which guarantees immediate deadlock resolution. While game-theoretic approaches are widely used in studying MAS conflict situations, there is extensive evidence in literature for the failure of such approaches where the main players are people or bounded rational agents [1]. On the other hand, there are works that report the successful use of game-theoretic approaches, in particular in repeated interaction domains [2]. Therefore, the success of such approaches in the deadlock domain is a priori inconclusive. In this paper we report the results of an experiment testing the effectiveness of a Game-Theoretic approach to the problem of distributed deadlock resolution of autonomous self-interested partially-rational agents. This is a part of an ongoing research aimed at studying distributed deadlock resolution in such settings and designing a restructuring heuristic that changes the input that each agent receives as a means for affecting the agents' decisions to better align with the desired solution.

2. THE DEADLOCK MODEL

This paper considers the standard Coffman deadlock model, commonly found in Operating Systems literature. The system is in a deadlock state if a circular chain $A = \{A_1, ..., A_N, A_1\}$ of agents exists, where each agent $A_i \in A$ attempts to acquire a resource held by agent A_{i+1} (A_1, in case $i = N$) in order to proceed with its execution.

Each agent A_i is associated with a processing time t_{A_i}, the time it needs to use the resource it requests from A_{i+1} before releasing the resource that A_{i-1} is waiting for. Each agent can also willingly release the resource it holds (opt out) at any time. In such case the agent needs to acquire both the resource released (now held by the previous agent in the chain) and the resource it was waiting for (held by the next agent) in order to proceed with its task. We assume that resources are acquired as soon as they are available.

We assume the existence of a central entity (e.g., operating system) that receives demands for resources and expected processing times and identifies deadlocks as they occur. The central entity supplies system-related information to the agents, though it cannot preempt the agents' hold on resources or enforce any particular behavior. In particular, since agents in a MAS can block their regular operation for various reasons, we assume that the system informs the agents once they are actually in a deadlock and supplies

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

them with the deadlock description. This latter information includes the number of agents in the deadlock and their processing times. The agents are assumed to be homogeneous in the sense that each agent has an equal chance of being the i-th agent in any deadlock and its processing time is drawn from a common distribution of values. An agent's strategy is thus the mapping $S : A \to t$, where t is the time since the deadlock is first reported to the agent until the agent opts out. The agents are assumed to be self-interested and their goal is to minimize the time it takes to complete their task. Since no agent has control over its own processing time, this goal is equivalent to minimizing its overall waiting time. From the system's point of view, the goal is to minimize the average waiting time of the agents.

3. ANALYSIS

In this section we develop the dominant Nash-Strategy for the problem. For exposition purposes, we use $A_{sub}(A_i, A_j)$ to represent the subchain of agents in A positioned between A_i and A_j. Also, WLOG, the deadlock is taken to be formed at time $t = 0$. Once an agent $A_i \in A$ opts out, the deadlock is resolved. In this case, agent A_i will need to wait $\sum_{k \neq i} t_{A_k}$, while any other agent A_j will wait a time equal to the total processing times of all the agents along the subchain $A_{sub}(A_j, A_i)$ (formally given by $\sum_{A_k \in A_{sub}(A_j,A_i)} t_{A_k}$). Notice that once the deadlock is resolved by agent A_i, no agent $A_j \neq A_i$ can reduce the time it needs to wait by opting out as well. This is because opting out will not affect the time that any of the agents in $A_{sub}(A_j, A_i)$ will need to wait until gaining a hold of the resource they requested.

From the system's point of view, regardless of the identity of the agent to opt out first, the optimum is achieved at $t = 0$. This is because all agents necessarily gain from an earlier deadlock resolution, given that all other parameters are fixed. If the deadlock is resolved by agent $A_i \in A$ at time $t = 0$, then the average waiting time is given by:

$$\frac{1}{N}\Big(\sum_{j \neq i} \sum_{A_k \in A_{sub}(A_j,A_i)} t_{A_k} + \sum_{k \neq i} t_{A_k}\Big) \qquad (1)$$

A lower bound for the expected average waiting time is obtained when agent A_i, for which Equation 1 is minimized, opts out at time $t = 0$. This solution can theoretically be achieved when each agent checks whether it is the agent to minimize Equation 1, and if so, opting out at $t = 0$. This solution can also be extended to the dominant Nash-Strategy when each agent waits indefinitely if it is not the agent that minimizes Equation 1. In such a case, none of the agents have an incentive to deviate from its strategy. Since none of the other agents ever opt out, the agent that is supposed to opt out at $t = 0$ will do better if it sticks with this strategy. Opting out at $t > 0$ is dominated by opting out at $t = 0$ for this agent and never opting out will necessarily result with an infinite waiting time. As for the other agents, since the deadlock is supposed to be resolved at time $t = 0$, none of them will have an incentive to deviate to a different strategy. In fact, based on the same argument, any protocol according to which one of the agents is selected to opt out by an external event (e.g., having the shortest or longest processing time) while none of the other agents ever opt out is in equilibrium.

4. EXPERIMENTS AND RESULTS

To enable the simulation of distributed deadlocks, we developed a system that simulates Coffman deadlocks. Agents in the system are put in a deadlock upon instantiation. The sole functionality of each agent is deciding whether to opt out of the deadlock based on the deadlock description and the time elapsed. The experiment was carried out with 28 agents programmed by computer science students in a core Operating Systems course. The goal set for the agents was to minimize their expected waiting time throughout the experiment. The students were given a detailed explanation about the game-theoretic-based solution to the problem. To simplify coding and assure that a deviation from the equilibrium strategy will not be a result of implementation bugs or difficulty, it was decided that the agent with the longest processing time in each deadlock will be the one to opt out. The drawbacks of deviating from this strategy, assuming everyone else is using it, were fully discussed and detailed in the task description. It was suggested that the students use this strategy, though it was made clear to them that there is no centralized mechanism enforcing it. In order to make the evaluation as realistic as possible, the experiment took place over the course of a few weeks, allowing the students to revise their strategies based on the results of thousands of deadlocks in which their agents participated. This process of repeated strategy updates and evaluations of performance was carried out until a week elapsed without any change made to the agents' strategies. The agents stored in the system at the end were considered the steady-state strategies. Deadlocks were generated automatically and randomly assigned with agents. The number of agents participating in each deadlock was uniformly drawn from a range of 2-10. The processing times were drawn from an Erlang distribution, which is the typical distribution of CPU burst times in operating systems, with parameters $\lambda = 0.01$ and $k = 1.5$ (yielding a mean of 150). Once an agent opted out, the system terminated and the waiting times of all the agents in the deadlock were calculated.

The results indicate that none of the subjects initially implemented the Nash-Strategy fully. Only 18 percent of the students have implemented the Nash-Strategy with an empty threat, by opting out after a constant time in deadlocks in which their agent is not the one with the longest processing time. The analysis of the steady-state set of distributions revealed that not only has no one changed her strategy to the game-theoretic one, even the number of partial implementations of the type described above decreased to only 3 percent of the strategies. In addition, the system's average waiting time in the steady-state was worse compared to the average obtained with the initial set of strategies.

The results demonstrate the failure of the Game-Theoretic approach in the distributed setting with self-interested bounded-rational agents. The main implication is that other approaches, such as input restructuring, should be considered for such settings.

Acknowledgments

This work was supported in part by ISF/BSF grants 1401/09 and 2008-404.

5. REFERENCES

[1] E. Manisterski, R. Lin, and S. Kraus. Understanding how people design trading agents over time. In *AAMAS*, volume 3, pages 1593–1596, 2008.

[2] M. Parameswaran, H. Rui, and S. Sayin. A game theoretic model and empirical analysis of spammer strategies. In *Collaboration, Electronic Messaging, AntiAbuse and Spam Conf.*, number 7, 2010.

Bounded Model Checking for Knowledge and Linear Time *

(Extended Abstract)

A. Męski†, W. Penczek‡, M. Szreter
Institute of Computer Science, PAS, Poland
{meski,penczek,mszreter}@ipipan.waw.pl

B. Woźna-Szcześniak, A. Zbrzezny
Jan Długosz University, IMCS, Poland
{b.wozna,a.zbrzezny}@ajd.czest.pl

ABSTRACT

We investigate symbolic approaches to Bounded Model Checking (BMC) for the Linear Temporal Logic extended with epistemic components (LTLK), interpreted over Interleaved Interpreted Systems. We propose two BMC translations for LTLK - one is based on SAT and the other is based on BDD - which we have implemented and tested on several benchmarks. We report on our experimental results that reveal advantages and disadvantages of SAT- versus BDD-based BMC for LTLK.

Categories and Subject Descriptors

F.3.1 [**Specifying and Verifying and Reasoning about Programs**]: Specification techniques; D.2.4 [**Software/Program Verification**]: Model checking; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Modal logic, Temporal logic

General Terms

Verification, Theory, Performance

Keywords

SAT, BDD, Bounded Model Checking, Temporal Epistemic Logic (LTLK), Interleaved Interpreted Systems

1. INTRODUCTION

Several approaches based on model checking [1] have been put forward for verification of multi-agent systems (MAS) [2, 8, 9]. Typically, they employ combinations of epistemic logic with branching or linear temporal logic. Some approaches reduce the verification problem to the one for plain temporal logic, while others treat typical MAS modalities such as (distributed, common) knowledge as first-class citizens and introduce novel algorithms for them.

In an attempt to alleviate the state-space explosion problem (i.e., an exponential growth of the system state space with the number of the agents) two main BMC approaches have been proposed, based on either BDDs [3] or SAT [8]. However, these approaches deal with the properties expressed in CTLK (i.e., CTL extended the with epistemic component) only.

In this short paper we aim at completing the picture of applying mathods based on the BMC symbolic verification to MAS by looking at the existential part of LTLK (i.e., ELTLK) interpreted on *interleaved interpreted systems* (IIS) [6]. IIS are a special class of interpreted systems (IS) in which only one action at a time is performed in a global transition. Our original contribution consists in defining two novel model checking methods for LTLK, namely a SAT- and BDD-based BMC. The methods have been implemented, tested, and compared with each other as well as with the tool MCK [2] on three benchmarks for MAS. Our experimental results reveal advantages and disadvantages of SAT- versus BDD-based BMC for LTLK on MAS, which are consistent with comparisons for temporal logics. Although our methods are described for IIS, they can be applied to IS as well, which we will show in our future paper.

2. BMC FOR ELTLK

Our SAT- and BDD-based BMC methods for ELTLK are, to our best knowledge, the first ones formally presented in the literature; the manual for MCK states that the tool supports SAT-based BMC for CTL*K. Unfortunately, no theory behind this implementation has ever been published.

Let M be a model for a given IIS, φ - an ELTLK formula, and $k \geq 0$ - a bound. The problem of checking whether M is a model for φ can be translated to the problem of checking the satisfiability of the following propositional formula: $[M, \varphi]_k := [M^{\varphi,\iota}]_k \wedge [\varphi]_{M,k}$. The formula $[M^{\varphi,\iota}]_k$ constrains the finite number of symbolic k-paths to be valid k-paths of M, while the formula $[\varphi]_{M,k}$ encodes a number of constraints that must be satisfied on these sets of k-paths for φ to be satisfied. Once this translation is defined, checking satisfiability of an ELTLK formula can be done by means of a SAT-solver. In the case of the BDD-based approach we reduce the ELTLK model checking problem to the problem of the ELTL model checking. When processing the verified LTLK formula, the states of the model are labelled with the subformulae that hold in these states. This approach is similar to the approach proposed for CTL* [1]. To perform BMC using BDDs we interleave the fixed-point computation of the reachable states with executions of the state-labelling procedure for ELTLK. For details details we refer the reader to [7] and [10].

3. EXPERIMENTAL RESULTS

We consider three benchmarks for which we give performance evaluation of our two BMC algorithms and the BMC algorithm of MCK for the verification of several properties expressed in ELTLK.

†Also affiliated with: University of Łódź, FMCS, Poland.
‡Also affiliated with: University of Natural Sciences and Humanities, Institute of Informatics, Poland.

*Partly supported by the National Science Center under the grants No. 2011/01/B/ST6/05317 and 2011/01/B/ST6/01477.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The tests have been performed on a computer with Intel Xeon 2 GHz processor and 4 GB of RAM, running Linux 2.6, with the default limits of 2 GB of memory and 2000 seconds of time. For every benchmark, each specification is given in the universal form, for which we verify the corresponding counterexample formula, i.e., the formula which is negated and interpreted existentially.

The presented approaches have been implemented as prototype modules of the tool VerICS [5]. All the benchmarks can be found at `http://verics.ipipan.waw.pl/bmcLTLK.zip`, together with instructions how to repeat our experiments.

Faulty Generic Pipeline Paradigm (FGPP) consists of Producer, Consumer, and a chain of n intermediate Nodes transmitting data, together with a chain of n Alarms enabled when some error occurs. We consider the following specifications:

$\varphi_1 = \mathrm{G}(ProdSend \rightarrow \mathrm{K}_C \mathrm{K}_P ConsReady)$,
$\varphi_2 = \mathrm{G}(Problem_n \rightarrow (\mathrm{F}Repair_n \vee \mathrm{G}Alarm_n Send))$,
$\varphi_3 = \bigwedge_{i=1}^{n} \mathrm{G}(Problem_i \rightarrow (\mathrm{F}Repair_i \vee \mathrm{G}Alarm_i Send))$,
$\varphi_4 = \bigwedge_{i=1}^{n} \mathrm{G}\mathrm{K}_P(Problem_i \rightarrow (\mathrm{F}Repair_i \vee \mathrm{G}Alarm_i Send))$.

A faulty train controller system (FTC) consists of a controller and n trains (for $n \geq 2$), one of which is dysfunctional. We consider the following specifications:

$\varphi_1 = \mathrm{G}(InTunnel_1 \rightarrow \mathrm{K}_{Train_1}(\bigwedge_{i=2}^{n} \neg InTunnel_i))$,
$\varphi_2 = \mathrm{G}(\mathrm{K}_{Train_1} \bigwedge_{i=1,j=2,i<j}^{n} \neg(InTunnel_i \wedge InTunnel_j))$.

Dining Cryptographers (DC) is a scalable anonymity protocol, which has been formalised and analysed in many works. Here we assume the formalisation of DC in terms of a network of automata [4], and we consider the following specifications:

$\varphi_1 = \mathrm{G}(odd \wedge \neg paid_1 \rightarrow \bigvee_{i=2}^{n} \mathrm{K}_1(paid_i))$,
$\varphi_2 = \mathrm{G}(\neg paid_1 \rightarrow \mathrm{K}_1(\bigvee_{i=2}^{n} paid_i))$,
$\varphi_3 = \mathrm{G}(odd \rightarrow \mathrm{C}_{\{1,\ldots,n\}} \neg(\bigvee_{i=1}^{n} paid_i))$.

Performance evaluation. An important difference in performance between the SAT- and BDD-based BMC reveals itself in the FTC benchmark, where the BDD-based method performs much better in terms of the total time and memory consumption. In the case of FGPP, BDD-BMC is still more efficient, but the difference is not that significant. Our SAT-based BMC significantly outperforms the BDD-based BMC for φ_2 of DC: SAT-BMC has computed the results for 3500 cryptographers, whereas BDD-BMC for 41. The reason is that there are at most two symbolic k-paths, and the length of the counterexamples is constant. This is also the case for φ_3 of FGPP. The efficiency of BDD-BMC improves for the formula φ_4 of FGPP comparing to φ_3, although they are similar. The reason is the presence of the knowledge operator that causes the partitioning of the problem to several smaller ELTL verification problems, which are handled much better by the implementation of the operations on BDDs. A noticeable superiority of SAT-BMC for φ_2 of DC follows from the long encoding times of the BDD for the transition relation. The reordering of the BDD variables does not cause any change of the performance in the case of FGPP and FTC, but for DC it reduces the memory consumption. This means that the fixed interleaving order we used can often be considered optimal, but the loss in the verification time to reorder the variables, in favour of reducing memory consumption, is also not significant and is often worth the tradeoff. In the case of φ_3 for DC, SAT-BMC was remarkably inferior to BDD-BMC, i.e., SAT-BMC managed to compute the results only for 3 cryptographers in the time of 5400 seconds, whereas BDD-BMC managed to compute the results for 17 cryptographers. This follows from the fact that φ_3 contains the common knowledge operator, which requires many symbolic k-paths to be analysed. For φ_1 of DC, our BDD-BMC has computed the results for 14 cryptographers, outperforming SAT-BMC (4 cryptographers). In most cases, BDD-BMC spends a considerable amount of time on encoding the system, whereas SAT-BMC on verifying the formula. Therefore, BDD-BMC may provide additional time gains when verifying multiple specifications of the same system.

We have compared MCK with our methods for the cases where the lengths of counterexamples scale correspondingly, thus minimising the factor played by different semantics. The comparison shows that for FGPP and FTC our methods are superior to MCK for all the tested formulae (sometimes by several orders of magnitude). There could be several reasons for this. While our approach is especially optimised for LTLK, it is likely that MCK treats LTLK formulae as CTL*K formulae, for which the translation is typically much less efficient. MCK consumes all the available memory even when formulae are surprisingly small (approx. 10^6 clauses and 10^5 variables) compared to those successfully tested in our SAT-BMC experiments (more than 10^8 clauses and variables in some cases). However, it should be noted that MCK implements different semantics of MAS, in which agents can perform independent actions simultaneously in a single step of the protocol, what may result in different counterexamples than given by IIS. This is the case of the DC benchmark, where MCK can profit from the strong locality and produces counterexamples of constant length, independently of the number of cryptographers, for all the formulae, being able to verify 15, 32, and 14 cryptographers for φ_1, φ_2, and φ_3, respectively. Using our approaches, we could verify, respectively, 14 cryptographers (BDD-BMC), 3500 (SAT-BMC), and 41 (BDD-BMC). We can conclude from our analysis that the BDD- and SAT-based BMC approches remain complementary and none of them is clearly superior in general, whereas in most cases MCK seems to be inferior to our BMC approaches.

4. REFERENCES

[1] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[2] P. Gammie and R. Meyden. MCK: Model checking the logic of knowledge. In *Proc. of CAV'04*, volume 3114 of *LNCS*, pp. 479–483. Springer-Verlag, 2004.

[3] A. Jones and A. Lomuscio. A BDD-based BMC approach for the verification of multi-agent systems. In *Proc. of CS&P'09*, volume 1, pp. 253–264. Warsaw University, 2009.

[4] M. Kacprzak, A. Lomuscio, A. Niewiadomski, W. Penczek, F. Raimondi, and M. Szreter. Comparing BDD and SAT based techniques for model checking Chaum's dining cryptographers protocol. *Fundam. Inform.*, 72(1-2):215–234, 2006.

[5] M. Kacprzak, W. Nabiałek, A. Niewiadomski, W. Penczek, A. Półrola, M. Szreter, B. Woźna, and A. Zbrzezny. VerICS 2007 - a Model Checker for Knowledge and Real-Time. *Fundam. Inform.* 85(1-4):313-328, 2008.

[6] A. Lomuscio, W. Penczek, and H. Qu. Partial order reduction for model checking interleaved multi-agent systems. In *AAMAS, IFAAMAS Press.*, pp. 659–666, 2010.

[7] A. Męski, W. Penczek, and M. Szreter. Bounded model checking linear time and knowledge using decision diagrams. In *Proc. of CS&P'11*, pp. 363–375, 2011.

[8] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundam. Inform.*, 55(2):167–185, 2003.

[9] F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via OBDDs. Journal of Applied Logic, 5(2):235–251, 2007.

[10] W. Penczek, B. Woźna-Szcześniak, and A. Zbrzezny. Towards SAT-based BMC for LTLK over interleaved interpreted systems. In *Proc. of CS&P'11*, pp 565–576, 2011.

The role of identity in agent design

(Extended Abstract)

Ines Di Loreto
ines.diloreto@fastwebnet.it

Fabien Hervouet
LIRMM/CNRS, University of Montpellier, France
fabien.hervouet@lirmm.fr

Categories and Subject Descriptors

I.2.0 [**Artificial Intelligence**]: Philosophical foundations

General Terms

Theory

Keywords

Agent, Identity, Uniqueness, Embodiment, Enaction

1. INTRODUCTION

This year AAMAS conference introduced a perspective track for "papers that analyze in some way the agent research community". The aim of this track is to understand what the trends are in agent research and foresee possible future directions. Instead of looking at where the agent community is going in an emergent way analyzing numerical trends, with this article we want to suggest where agent research could go but is not yet going. We are of the opinion that in the agent research community most of the current trends originate from the translation of particular concepts – mostly from analytic philosophy – which are only a particular western way to look at philosophy and agents. With this article we want to suggest that other paths originating from philosophy could be taken into account in order to create different directions in agent design.

2. THE ROLE OF IDENTITY: FROM PHILOSOPHY TO AGENT DESIGN

Analyzing current trends in agents design we observed that while trying to model and reproduce humans and societies, agent design mostly does not use a structured construction of the identity concept. In the rest of this paper we will support this position analyzing the identity concept, paralleling agent design and contemporary philosophical assumptions about the concepts of uniqueness, body and mind.

2.1 The concept of uniqueness

There is almost no debate about uniqueness in agent design. More generally uniqueness is in essence an issue for computer science. Any data can be copied and replicated with an absolute guarantee of ending up with two exact similar objects. This interesting property prevented researchers from really tackling such an issue. As a result a certain part of agent design seems to work on a "Universal Agent", deriving from a "Universal Man" theory from the philosophy of Plato and the republican ideals of equality, which in essence does not need a structured uniqueness to be implemented. In Multi-Agent System (MAS) identity is mainly structured from the point of view of the role of the agents. MAS usually put a multitude of agents together in order to accomplish a certain global task or to have a certain global behavior. This means that even if agents do not act exactly the same in a local way, they often originate from the same piece of code that takes into account some predefined interactions with their peers.

However, while computer science seems not to care about uniqueness, in the 1970s Maturana and Varela addressed the complex problem of autonomy, knowledge and identity in biology [9]. They characterized living organisms by coining the concept of *autopoiesis* which is defined as a complex incessant process of self-production of the system by itself, replacing its components to compensate for continuous external disturbances. In short an autopoietic system can be seen as a homeostatic system whose invariant principle is its own organization (seen as the network of relationships that defines it). Therefore in this context uniqueness may be defined as this historical coupling, i.e. the historical adaptive activity of the structure in order to fit the organization. The autopoiesis theory has inspired some scholars in the artificial life and agent design domains such as [4, 3, 13], but it remains generally too few addressed.

In the same way psychology has discussed the concept of uniqueness from its very beginning. In the '60s, Piaget led the constructivist movement promoting the vision that every individual has the ability to hold their own reconstruction of reality. This theory of knowledge supports the fact that identity is perpetually in construction, deriving from our own adaptation process in direct confrontation with the environment. This point of view will be explored a few years further by Varela et al. and their theory of *enaction* [12]. The enaction paradigm postulates the co-emergence of both cognition and perceived world through the performative body in action in the environment. Therefore uniqueness can only appear within a pure bottom-up mechanism.

2.2 The concept of body

Regarding the "body concept" it is very interesting to notice a fundamental difference in the agent design approach

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

between computer science and robotics. Indeed, while computer science focused mainly on disembodied reasoning capabilities, robotics was created with the idea of body, of physical interaction, and it is based on the experimental principles of physics and mechanics in a very grounded manner. Therefore, the advent of robotical agent design is a major step towards the consideration of the notion of embodiment for intelligent agents. Three major kinds of robotic architectures emerged : *deliberative architectures* [10] using symbolism and generally organized into multiple hierarchical layers, *purely reactive architectures* [6, 1] built by stacking finite state machines without reasoning nor symbolization, and *hybrid layered architectures* like [8] combining the advantages of behavioral and deliberative architectures.

Nevertheless, as Ziemke argues in [13], even if it has been recognized for a decade that embodiment is a necessary condition to characterize living organisms [11, 12] and that more and more researchers have attempted to address this absolute need for embodied cognition (in the developmental robotics community for instance, see [2] for a recent survey), robotics "is largely 'stuck' in the old distinction between hardware and software". Cognitivist vision is still largely dominant over a pure varelian enactive vision [13].

2.3 The concept of mind

In the 20th century problems deriving from the division between mind and body also become evident in philosophy. One of the most well-known 20th century philosophical movements is analytical philosophy. Very simply, analytical philosophy is characterized by the application of a logical method to traditional philosophical problems often using modern formal logic and language analysis. Computer science has deep analytical foundations, since the von Neumann's vision of cognition as logical problem solving.

Nowadays although the metaphor of the agent as a symbol interpreter is always present, more complex models of agents have been proposed. For instance, we can cite the Belief-Desire-Intention model [5] which is a widely used more complex model articulated around the notion of knowledge in pure bodyless approach.

Furthermore in the same pure mind-only way, interactions between artificial agents have been historically only communicational. Languages designed were nothing but logical formalized protocol philosophically based on Austin's and Searle's speech acts theories. We think this approach is inherited from the ideas of philosophers like Wittgenstein relayed later by the behaviorist psychology of Skinner. For these authors the only way we can study thought is to look at verbal behavior because, unlike in private thoughts, the behavior can be *scientifically* verified. The legacy of analytic philosophy is the vision of the mental representation.

This logical vision combined with an omniscient point of view in agent design has shown its limits for researchers who wanted to create more subjective and complete agent by-passing the mind-body dualism.

3. PERSPECTIVES

In the first part of this paper we have shown that contemporary psychology, philosophy and even biology have interesting ways of looking at the concepts of identity, uniqueness, performance and environment as interlaced and interacting. Although the problem solving vision is useful in many ways, integrating different concepts can lead to a more global vision about autonomous agent design. Obviously identity is only one of the concepts that could be analyzed and the analysis proposed in this paper makes up only a subset of the concepts that can consist in identity. For a more in depth analysis of the concept of identity in agent design see [7].

Based on our analysis we suggest that agent design can integrate the following concepts.

Uniqueness The concept of uniqueness could be very interesting to integrate in agent design in a mixed environment involving virtual agents as well as human agents in order to bond more easily with each other.

Autopoiesis The concept of autopoiesis is strictly linked with the uniqueness one. In autopoietic systems uniqueness may be considered as a particular trajectory of the coupling between organization and structure.

Enaction The concept of enaction could be integrated in agent design in order to overcome the dualism of mind/body. Going beyond this dualism can help to create agents which are more adaptive to unknown environments thanks to their deep physical grounding.

At the end of this short paper we can then say that the concepts we suggest integrating in the agent design paradigm are nothing more than necessary steps – but not necessarily sufficient – to reach the autonomy stage. However we strongly believe that as long as the design of agents is mainly based on analytic philosophy, we can only have an enlargement of the domain and not a paradigm shift which is at the basis of major advances in science.

4. REFERENCES

[1] R. C. Arkin. Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *International Conference on Robotics and Automation*, volume 4, pages 264–271. IEEE, 1987.

[2] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida. Cognitive developmental robotics: a survey. *IEEE Transactions on Autonomous Mental Development*, 1:12–34, 2009.

[3] R. D. Beer. Autopoiesis and cognition in the game of life. *Artificial Life*, 10:309–326, 2004.

[4] P. Bourgine and J. Stewart. Autopoiesis and cognition. *Artificial Life*, 10:327–345, 2004.

[5] M. Bratman. *Intention, Plans, and Practical Reason.* Harvard University Press, Cambridge, MA, 1987.

[6] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[7] I. Di Loreto and F. Hervouet. Identity: the contribution of contemporary philosophy to agent design. Technical report, LIRMM, Jan 2012.

[8] R. J. Firby. *Adaptive Execution in Complex Dynamic Worlds.* PhD thesis, Yale University, 1989.

[9] H. R. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living.* Reidel, 1980.

[10] N. J. Nilsson. *Principles of Artificial Intelligence.* Tioga Publishing Company, 1980.

[11] R. Pfeifer and C. Scheier. *Understanding intelligence.* MIT Press, 1999.

[12] F. Varela, E. Thompson, and E. Rosch. *The Embodied Mind: Cognitive science and human experience.* MIT Press, 1991.

[13] T. Ziemke. Are robots embodied? In *Lund University Cognitive Studies*, pages 75–83, 2001.

Demonstrations

SAFEPED: Agent-Based Environment for Estimating Accident Risks at the Road Black Spots (Demonstration)

Gennady Waizman
Department of Geography and Human Environment,
Tel Aviv University
Ramat Aviv, Tel Aviv, 69978, Israel
+972-54-2012193
gennadyw@post.tau.ac.il

Itzhak Benenson
Department of Geography and Human Environment,
Tel Aviv University
Ramat Aviv, Tel Aviv, 69978, Israel
+972-3-6409896
bennya@post.tau.ac.il

ABSTRACT
The data on traffic accidents clearly point to the "Black Spots", where the accident rate remains high in months and years. However, road safety research is still far from understanding why this certain place on a road is risky. We tackle the problem by developing SAFEPED, multi agent microscopic 3D simulation of cars' and pedestrians' dynamics at the black spot.

Categories and Subject Descriptors
I.6.5 Computing Methodologies, Simulation and modeling, Model Development

General Terms
Algorithms, Design, Reliability, Experimentation, Human Factors, Standardization

Keywords
Traffic accidents, Black Spot, agent-based modeling, spatially-explicit modeling

1. INTRODUCTION

Police data on the number of traffic accidents clearly point to the "Black Spots", where the accident rate remains high in months and years. However, road safety research is still far from understanding why certain road locations are risky.

Essentially, we lack the knowledge of how pedestrians and drivers interact when facing a potentially dangerous traffic situation and, most important, the integrated framework that relates the data on human behavior to the real-world traffic situations.

So far, road safety is studies with the general purpose traffic simulation models extended towards including conflict statistics. This approach, however, is inherently limited. The dynamic road safety model should incorporate the variables that are critical for road incidents but superfluous for simulating general traffic: the characteristics of mechanical and functional characteristics of vehicles and in-vehicle systems and, especially, the rules of drivers' and pedestrians' behavior, including drivers and pedestrians' awareness and reaction to each other [1].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4–8 June 2012, Valencia, Spain.

We present safety oriented high-resolution spatial micro-simulation model of car and pedestrian traffic that enables direct simulation of the road accidents and associated risks.

2. SAFEPED simulation environment

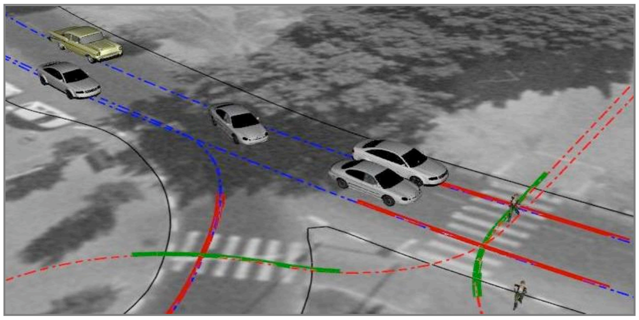
To represent the dynamic reality at the Black Spot and merge it with the experimental data on drivers' and pedestrians' behavior we have developed SAFEPED - Multi-agent environment for spatially explicit microscopic 3D simulation of the Black Spot dynamics.

SAFEPED serves as a testbed for evaluating experimentally estimated drivers' and pedestrians' behavioral rules and estimating accident risks in various traffic situations. It aims at analyzing disadvantageous environmental design at the Black Spot and assessing alternative architectural solutions there.

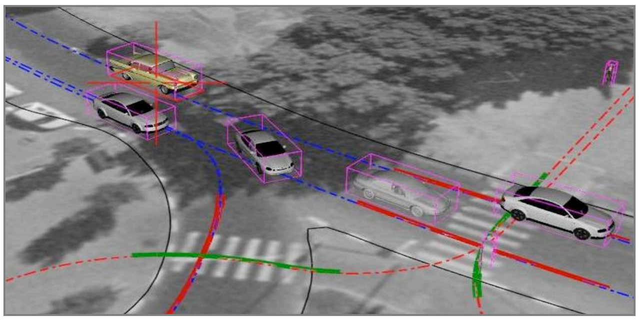
The major features of the SAFEPED are as follows:
- SAFEPED agents are autonomously behaving pedestrians and drivers who see and estimate the 3D-movement of the other agents and react in response once in 0.04 sec;
- SAFEPED agents see each other in 3D and behave based on the 3D visibility
- The user defines the properties and goals of movement of the drivers and pedestrians participating in the traffic episode;
- The rules of agents' behavior are based and validated based on the analysis of video footage captured at the places of drivers-pedestrian interaction.

During the simulation, SAFEPED records the full life-history of every agent, including all crash and near-crash episodes. The user can analyze the crash and near-crash statistics, rewind and replay the simulation starting from any moment of time, observe accidents from various viewpoints, including the viewpoints of the crash participants (Figure 1). The user can also intervene into the model dynamics by taking control over one or more agents. To analyze accident risks SAFEPAD applies indicators describing the conflicts between traffic participants, such as Time-to-Collision and Post Encroachment Time [2].

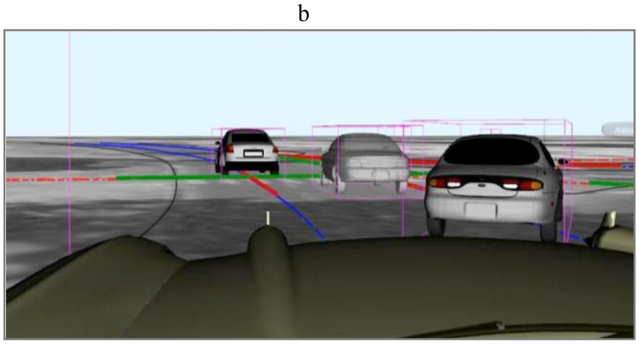
The paper presents the SAFEPED and results of investigation of several accident scenarios. See general view of the SAFEPAD at http://www.youtube.com/watch?v=ia3W8oiTVYw&feature=related , formalization of visibility at http://www.youtube.com/watch?v=6KFcfFREIt8&feature=related, and illustration of traffic accident at http://www.youtube.com/watch?v=axWEGNetpM0



a



b



c

Figure 1. SAFEPAD traffic episode: (a) agents participating in the episode; (b) visibility of the other cars for chosen agent; (c) visibility of the other cars from the viewpoint of the chosen agent

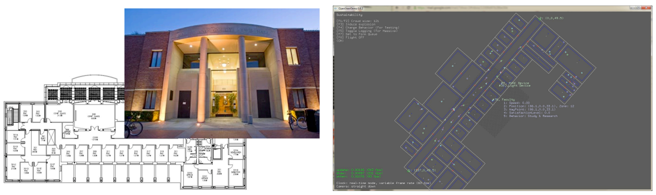
REFERENCES

[1]. Gettman, D. and Head, L., 2003, Surrogate Safety Measures From Traffic Simulation Models, *Report No. FHWA-RD-03-050, Federal Highway Administration*

[2]. Morsink, P., Wismans, L., Dijkstra, A. 2008, Micro-Simulation For Road Safety Impact Assessment Of Advanced Driver Assistance Systems, *Seventh European ITS Congress*, Geneva, paper 2676

Sustainable Multiagent Application to Conserve Energy (Demonstration)

Jun-young Kwak, Pradeep Varakantham*, Rajiv Maheswaran, Milind Tambe, Farrokh Jazizadeh, Geoffrey Kavulya, Laura Klein, Burcin Becerik-Gerber, Timothy Hayes, Wendy Wood
University of Southern California, Los Angeles, CA, 90089
*Singapore Management University, Singapore, 178902



(a) Educational Building (RGL) (b) Simulation Testbed

Figure 1: The actual research testbed at USC and our simulator

1. INTRODUCTION

Limited availability of energy resources has motivated the need for developing efficient measures of conserving energy. Conserving energy in commercial buildings is an important goal since these buildings consume significant amount of energy, e.g., 46.2% of all building energy and 18.4% of total energy consumption in the US [1]. This demonstration focuses on a novel application to be deployed at Ralph & Goldy Lewis Hall (RGL) at the University of Southern California as a practical research testbed to optimize multiple competing objectives: i) energy use in the building; ii) occupants' comfort level; and iii) practical usage considerations.

This demonstration complements our paper in the AAMAS innovative applications track [4], presenting a novel multiagent building application for sustainability called SAVES (Sustainable multi-Agent systems for optimizing Variable objectives including Energy and Satisfaction). This writeup will provide a high-level overview of SAVES and focus more on the proposed demonstration, but readers are referred to [4] for a more technical description. SAVES provides three key contributions: (i) jointly performed with the university facility management team, our research is based on actual building and occupant data as well as real sensors and devices, etc.; (ii) it focuses on non-residential buildings, where human occupants do not have a direct financial incentive in saving energy; and (iii) SAVES uses a novel algorithm for generating optimal *BM-MDP* (Bounded parameter Multi-objective MDP) policies.

We demonstrate SAVES to show how to achieve significant energy savings and comparable average satisfaction level of occupants while emphasizing the interactive aspects of our application.

2. APPLICATION DOMAIN

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

Figure 1(a) shows the real testbed building (RGL) in which SAVES is to be deployed and the floor plan of 3^{rd} floor. This campus building has three floors in total and is composed of classrooms, offices for faculty and staff, and conference rooms for meetings. Each floor has a large number of rooms and zones (a set of rooms that is controlled by specific piece of equipment). The building includes building components such as HVAC (Heating, Ventilating, and Air Conditioning) systems, lighting systems, office electronic devices like computers and AV equipment, and human occupants divided into permanent (faculty, staff, researchers, etc.) and temporary (students or faculty attending classes or meetings, etc.).

As an important first step in deploying SAVES in the actual building, we have constructed a realistic simulation testbed (Figure 1(b)) based on the open-source project OpenSteer (http://opensteer.sourceforge.net/) and validated the simulation testbed using real building energy and occupancy data.

Our simulation considers three building component categories: HVAC devices that control the temperature of the assigned zone, lighting devices that control the lighting level of the room, and appliances. The energy consumption of such building components is calculated based on various parameters designated by the ASHRAE standard and actual energy consumption data in the testbed building. We also built two types of human occupants in our simulation using the agent behavior framework. Permanent occupants follow their regular schedules and temporary occupants stay in the building for classes or meetings and leave once they end. Occupants also have a satisfaction level, modeled as a percentage between 0 (fully dissatisfied) and 100 (fully satisfied).

In this domain, there are two types of energy-related occupant behaviors that SAVES can influence to conserve energy use: individual and group behaviors. Individual behaviors only affect an environment where the individual is located, and group behaviors lead to changes in shared spaces and require negotiation with a group of occupants.

The desired goal in the educational building is to optimize multiple criteria, i.e., achieve maximum energy savings without sacrificing the comfort level of occupants.

3. APPROACH: SAVES

SAVES is composed of two types of agents: room agents and proxy agents (Figure 2). There is a dedicated room agent per office and conference room, in charge of reducing energy consumption in that room. It can access sensors to retrieve room information and energy use and impact the operation of actuators. A proxy agent [5] is on an individual occupant's hand-held device and it has the corresponding occupant's models. Proxy agents communicate on behalf of an occupant to the room agent based on their adjustable

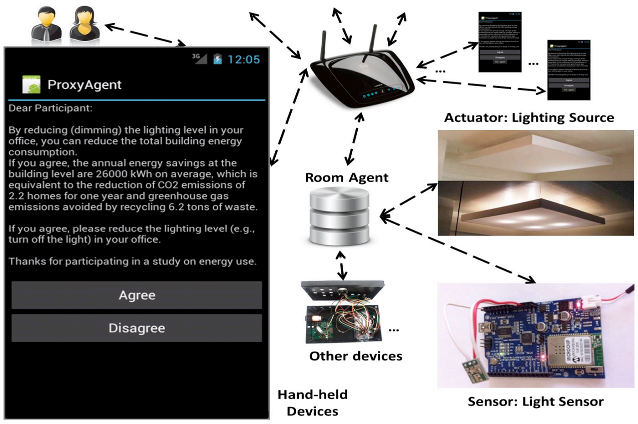


Figure 2: Agents & Communication Equipment in SAVES. An agent in SAVES sends feedback including energy use to occupants.

autonomy – when to interrupt a user and when to act autonomously.

Room agent reasoning is based on a new model called BM-MDPs, which is one of the contributions of this research. BM-MDPs are responsible for planning simple and complex tasks. These tasks include negotiating with groups of individuals to relocate meetings to smaller rooms to save energy, negotiating with multiple occupants of a shared office to reduce energy usage in the form of lights or HVACs, and others. BM-MDPs must reason with multiple objectives, but simultaneously must reason with the uncertainty in the domain, and we ended up building BM-MDPs to address both these challenges and requirements.

BM-MDPs are a hybrid of MO-MDPs [2] and BMDPs [3]. Similar to BMDPs, the transition and reward functions in BM-MDPs have closed real intervals. Whereas BMDPs are limited to optimizing a single objective case, BM-MDPs can i) optimize over multiple objectives (i.e., a vector of reward functions) with ii) different degrees of model uncertainty.

Figure 3 shows the cumulative total energy consumption on the y-axis in *kWh* and time on the x-axis as obtained in our simulator (Figure 1(b)). SAVES (based on the BM-MDP policies) achieved energy savings of 31.27% with an actually measured compliance rate (68.18%) and up to 42.45% with the ideal compliance rate (i.e., SAVES-IDEAL: occupants *always* accept the suggestions provided by the SAVES room agents) when compared to the manual control strategy. The manual strategy represents the current strategy operated by the facility management team in RGL (Figure 1(a)). In addition to energy savings, we compared the average satisfaction level of human occupants under different control strategies in the simulation testbed. Similarly to Figure 3, SAVES reliably showed higher average satisfaction level (70% or higher) than other control strategies as it plans ahead of the schedules using BM-MDP policies.

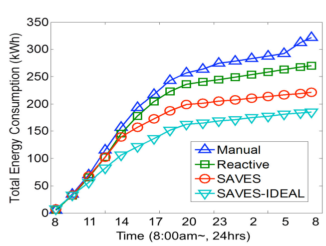


Figure 3: Energy Savings

4. DEMO

We demonstrate SAVES[1] using the 3^{rd} floor of RGL. Our demo consists of two parts. The first part uses our verified simulation testbed (Figure 1(b)) that is capable of communicating and negotiating with simulated occupants in the building and participants in the demo. The simulation environment is shown on the screen during the entire demo so that people are aware of the situation. The demo engages people by asking them to provide energy saving suggestions: we give them the detailed data of RGL 3^{rd} floor including energy rates of different rooms/zones, occupants' information, occupants' comfort levels, etc. Then, we ask them to make suggestions to reduce energy consumption. In particular, we provide three possible energy behaviors to participants in the demo: reduce the temperature by X°F in room/zone Y, dim the lighting level to Z in room/zone A, and relocate a meeting in conference room B to a smaller office C, where X, Y, Z, A, B, and C are user-chosen variables. We implement those suggested energy behaviors in the simulation environment and compare the performance between SAVES and participants' suggestions. Since our demo handles multi-objective optimization problems, we compare a rate of energy savings as well as the resulting comfort level changes.

The second part focuses on demonstration of proxies on the actual hand-held devices based on the following simple meeting relocation scenario considering group behaviors.

Group Meeting Relocation Negotiation Example Consider a meeting that has been scheduled with two attendees (P_1 and P_2) in a large conference room that has more light sources and appliances than smaller offices. Since the meeting has few attendees, the room agent can negotiate with attendees to relocate the meeting to nearby small, sunlit offices, which can lead to significant energy savings. The room agent handles this negotiation based on BM-MDPs. There are three objectives that the room agent needs to consider during this negotiation: i) energy saving, ii) P_1's comfort level change, and iii) P_2's comfort level change. The room agent first checks the available offices. Assuming there are two available offices A and B, the room agent asks each attendee if she or he will agree to relocate the meeting to one of the available offices. In asking an attendee, the room agent must consider the uncertainty of whether an attendee is likely to accept its offer to relocate the meeting. Since asking incurs a cost (e.g., cost caused by interrupting people), the room agent needs to reason about which option is preferable considering P_1 and P_2's likelihood to accept each option and the reward functions for each option to reduce the required cost and maximize benefits. Assuming A is preferable, the optimal policy of the agent is "ask P_1 first about A"–"if P_1 accepts, ask P_2 about A"–"if P_1 does not reply, ask P_1 about A again"–"repeat the process with B"–"if both agree, relocate the meeting"–"if both disagree, find other available options."

Each participant is provided with a mobile phone having a proxy agent on it. A proxy agent has a pre-set adjustable autonomy. Room agents initiate negotiations with simulated occupants or participants in the demo to conserve energy during the simulation, and SAVES specifically provides suggestions for energy savings to participants via mobile phones (as shown in Figure 2).

5. REFERENCES

[1] *Buildings Energy Data Book*. U.S. Dept. of Energy, 2010.

[2] K. Chatterjee, R. Majumdar, and T. A. Henzinger. Markov decision processes with multiple objectives. In *STACS*, 2006.

[3] R. Givan, S. Leach, and T. Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 2000.

[4] J. Kwak, P. Varakantham, R. Maheswaran, M. Tambe, F. Jazizadeh, G. Kavulya, L. Klein, B. Becerik-Gerber, T. Hayes, and W. Wood. SAVES: A sustainable multiagent application to conserve building energy considering occupants. In *AAMAS*, 2012.

[5] P. Scerri, D. V. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *JAIR*, 17:171–228, 2002.

[1]SAVES demo: http://www.youtube.com/watch?v=LtdbroGTFmE

Migrating Artificial Companions (Demonstration)

Iain Wallace, Michael Kriegel, Ruth Aylett
Heriot-Watt University
Edinburgh, UK
{I.Wallace,M.Kriegel,R.S.Aylett}@hw.ac.uk

ABSTRACT
Migration is the ability of an agent to transfer from one embodiment, for example a robot, to another such as a mobile phone. For agents that are to act as personal companions migration is desirable as access to different capabilities can provide more constant companionship to a user. This interactive demonstration of screen to phone migration illustrates one application of the open-source architecture developed on the LIREC project to support migration of an affective agent across many types of embodiment.

Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence – *Intelligent Agents.*

General Terms
Design, Experimentation, Human Factors.

Keywords
Intelligence for human-robot interaction, Middleware for robot systems, Affect and personality, Migration.

1. INTRODUCTION
The LIREC project[1] (Living with Robots and intEractive Characters) investigates long term interaction that combines work on the integration of robots into human social environments with that of virtual agents. It combines the development of an innovative agent framework and user studies in an attempt to carry out experiments to guide the design of social agents that can play an acceptable long-term role. Both social robots and virtual agents are embodied, the former physically and the later virtually. Physical embodiment raises still unsolved engineering problems of power sources, mobility and localisation that typically limit the ability of robots to accompany humans as they move from one social environment to another - for example from home to work. Virtual embodiments are much more transportable but by their nature cannot perform physical tasks such as fetching and carrying.

For this reason, LIREC investigates migration, the ability of a synthetic companion to move from one embodiment to another. This of course raises a new set of research questions, of which the most important is: what exactly migrates? We define this as the companion's identity, by which we mean those features that persist and make it unique and recognisable from the user's perspective. These features, themselves a research topic [6], may include common attributes of the different embodiments, for example similar visual appearance, but also common aspects of interactional behaviour, such as emotional expressiveness, and a memory of events and interactions that have taken place in multiple embodiments.

The question of what migrates also requires a technological answer. It is clear that a degree of architectural commonality across embodiments is required if migration is to be a generic capability of LIREC companions, which consist of many (~20) different robotic and virtual platforms. This demonstration shows instances of the developed architecture on two (virtual) platforms, and allows a user to trigger migration between the two, and see that the agent's identity migrates between them.

2. CHALLENGES
There are several challenges that impact the design of the architecture for the LIREC project, and motivate some of the decisions.

2.1 Identity
The retention of identity is one of the key benefits to having a migrating personal companion, but this raises the question of what exactly makes up identity and therefore needs to be migrated. Or to paraphrase, when does a user perceive that an agent has migrated from A to B rather than that agent A deactivated and B awoke?

Identifying these characteristics, and how to indicate migration to a user, is an open problem and the subject of research and experimentation in the LIREC project (e.g. [4]). As such, the agent architecture used must support migration between diverse platforms and the ability to include things such as behaviour characteristics, appearance and memory.

2.2 Memory
A personal companion should remember and learn from interactions, e.g. an office companion robot could remember messages from callers to the office and pass them on to team members when they return. The focus of LIREC is long-term interaction so memory modelling is a key component and also impacts on migration. Memory affects behaviour, and so must in some way be migrated, but also it grows with time and so it may be impractical to remember and migrate everything. An overview of the memory models employed in LIREC can be found in [3].

2.3 Infrastructure
For a companion to function across many embodiments, each with different capabilities, we must be sure the architecture is designed to handle these differences. This is somewhat at odds with the

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

need to standardise the software infrastructure to allow migration. Any architecture will have to describe the available sensors and effectors of an embodiment on a meta-level so that even an agent unacquainted with a certain embodiment can inhabit it. This requires sufficient abstraction of the low level functions of a platform, and suggests a layered architecture – described in the next section.

3. AN ARCHITECTURE FOR MIGRATING COMPANIONS

Necessarily this is only the briefest overview of the architecture, more details can be found elsewhere, for example [5] provides detail on the middleware principally responsible for supporting migration and [3] has detail on the memory.

The architecture is a standard three layer design to handle the various levels of abstraction required. A summary of these layers follows, with pointers to further reading.

3.1 Top Layer

This is the highest level of abstraction, and contains the agent's "mind" and memory, responsible for action selection and reasoning over goals and emotions. For many of the companion scenarios, it is important the agents reason with reference to emotion, and so the emotional continuous planner FAtiMA [2] is used, which has been enhanced for the LIREC project with a "theory of mind" component and advanced memory mechanisms, e.g. generalisation and activation based forgetting. This level is very embodiment-independent, concerned mainly with goals and high level actions and not the details of how they can be achieved, which may vary between embodiments and is handled by the middle layer.

3.2 Middle Layer

This is the layer responsible for co-ordinating the various sensors and effectors, matching the competencies of a platform to the needs of actions requested by the top layer. The middleware developed and made available as open-source is CMION [5] (Competency Management with ION), built on the agent simulation framework ION [7]. This wraps functionalities of an embodiment in competencies, which are provided with a basic means of intercommunication and data storage. This common interface allows for a competency manager to map actions of the top layer to a predefined competency execution plan consisting of a number of competencies that realise the requested action in the embodiment. It is designed to be modular and portable, written in Java – for example there is a version for Android mobile devices. The modularity allows for dynamic loading of competencies as required. This is the layer that handles migration. All of the top layer is migrated as it is independent, whereas the exact competencies that require migration depend on the embodiments in question, therefore it must be this layer managing the process.

3.3 Lower Layer

This is the layer that handles implementation details of individual platforms, and so varies more significantly. Most of the LIREC robots use SAMGAR [1], a modular robotics framework developed within LIREC, whereas other embodiments may just use platform specific methods, e.g. Android mobile devices.

4. DEMONSTRATION

The demonstration for AAMAS will consist of a virtual agent embodied in a monitor that people may interact with via simple gestures, and an alternative mobile phone or tablet embodiment that the agent migrates to. The user can then continue to interact with the same agent. This combined with a poster detailing the architecture developed will allow us to discuss the various issues encountered on LIREC and explain the operation to those interested in re-use of the framework. The architecture has shown its general applicability in the many different embodiments used by LIREC, and its public provision under open-source licenses is a key legacy of the project and so may be of wide interest.

A video demo of one of our research scenarios, containing several such migrations, can be seen online at:

http://vimeo.com/21156543

5. ACKNOWLEDGMENTS

This work was partially supported by the European Commission (EC) and is currently funded by the EU FP7 ICT215554 project LIREC (Living with Robots and intEractive Companions). The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

6. REFERENCES

[1] K. D. Casse, K. L. Koay, W. C. Ho, and K. Dautenhahn. Reducing the cost of robotics software: SAMGAR, a generic modular robotic software communication architecture. In Int. Conf. on Advanced Robotics 2009, ICAR 2009.

[2] J. Dias and A. Paiva. Feeling and reasoning: A computational model for emotional characters. EPIA 2005, pages 127-140, 2005.

[3] Mei Yii Lim, Ruth Aylett, Wan Ching Ho, Patricia Vargas and Sibylle Enz, A Socially-Aware Memory for Companion Agents. The 9th International Conference on Intelligent Virtual Agents, pp20-26 Amsterdam, Sept 14-16, 2009.

[4] K. L. Koay, D. S. Syrdal, K. Dautenhahn, K. Arent, Ł Małek and B. Kreczmer. Companion Migration – Initial Participants' Feedback from a Video-Based Prototyping Study. In Mixed Reality and Human-Robot Interaction, Intelligent Systems, Control and Automation: Science and Engineering, 2011.

[5] M. Kriegel, R. Aylett, P. Cuba, M. Vala and A. Paiva. Robots meet IVAs: A Mind-Body Interface For Migrating Artificial Intelligent Agents. In proceedings 10th Int. Conf. On Intelligent Virtual Agents (IVA 2011).

[6] Syrdal, D.S., Koay, K.L., Walters, M.L., Dautenhahn, K.: The boy-robot should bark! Children's impressions of agent migration into diverse embodiments. In: Proceedings: New Frontiers of Human-Robot Interaction, a symposium at AISB (2009).

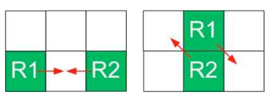
[7] Vala, M., Raimundo, G., Sequeira, P., Cuba, P., Prada, R., Martinho, C., Paiva, A.: ION framework - a simulation environment for worlds with virtual agents. In: Proceedings of the 9th International Conference on Intelligent Virtual Agents. (IVA 2009).

Effective Methods for Generating Collision Free Paths for Multiple Robots based on Collision Type (Demonstration)

Fan Liu, Ajit Narayanan, Quan Bai
School of Computing and Mathematical Sciences
Auckland University of Technology
Auckland, New Zealand
{richard.liu, ajit.narayanan, quan.bai}@aut.ac.nz

1. COLLISION TYPES IN MULTI-ROBOT SYSTEMS

Collision avoidance is an important topic in multi-robot systems. Existing multi-robot pathfinding approaches ignore sideswipe collisions among robots (i.e., only consider the collision which two agents try to occupy the same node during the same time-step) [1, 3, 4], and allow diagonal move between two adjacent nodes (e.g., Figure 1(b)). However, in many real world applications, sideswipe collisions may also block robots' movements or cause deadlocks. For example, as shown in Figure 1, if the size of two robots is as big as the grid size they occupied, collisions will happen not only between robots R1 and R2 in the situation depicted in Figure 1(a), but also that in Figure 1(b), which is typically not considered as a collision in existing multi-robot systems.

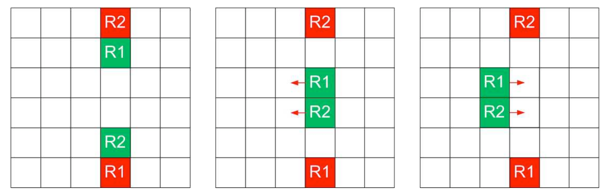


(a) (b)

Figure 1: (a) Occupy the same position. (b) Sideswipe collision.

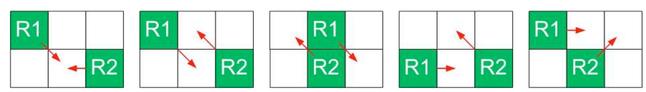
To overcome the limitation depicted in Figure 1(b), we investigate all possible collision scenarios in a multi-robot system (the speed / velocity of robots is taken into consideration when describing these collisions) when robots are moving, and identify one deadloop type and five collision types. Other collision types involving non-movement of robots due to breakdown are not included in our scenarios. We claim that all possible scenarios that may hinder a robot's planned motion in a two-dimensional space can be covered by these collision / deadloop types (with symmetry). The scenario that may cause a deadloop situation in a multi-robot system, on the other hand, is depicted in Figure 2. The five collision types are head-on, front sideswipe, rear sideswipe, front-end swipe and front-end sideswipe, which are illustrated from Figure 3(a) to (e), respectively. Front sideswipe (Figure 3(b)) and rear sideswipe (Figure 3(c)) can occur only on diagonal moves for both robots.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.



(a) (b) (c)

Figure 2: Illustration of deadloop. The green square and the red square are the robot positions and the goal positions for two robots, respectively. R1 and R2 are robot 1 and robot 2. (a) The initial position for two robots. (b) and (c) The dead looping condition is encountered and repeated in-between (b) and (c) infinitely as each robot makes a move that mirrors the other robot's.



(a) (b) (c) (d) (e)

Figure 3: Illustration of 5 collision types. (a) Head-On. (b) Front Sideswipe. (c) Rear Sideswipe. (d) Front-End Swipe. (e) Front-End Sideswipe.

2. ROBUST COLLISION AVOIDANCE STRATEGY

In this work, we also propose a coordinator-based (centralized) strategy to coordinate robots' movements. The strategy repeats a 'plan-evaluate-move' process to plan robots' routes and avoid potential collisions. In details, collision avoidance is achieved through the following steps.

1. Each robot computes its optimal path by using a classical path finding algorithm, i.e., the A* algorithm [2].
2. Each robot reports to the coordinator its current node position, previous node position, intended node position and estimated distance remaining to the goal node.
3. The coordinator detects potential deadloop and collisions based on robots' intentions (i.e., the nodes the robots want to move to). If no collision or deadloop is detected, goto Step (6), otherwise goto the next step.
4. Robots with potential collisions or deadloop will use the Super A* algorithm, which is described in Algorithm 1, to replan their routes in a decoupled manner.

Robots with less remaining distance will get higher priority for path planning purposes.

5. Robots will then report their re-planned intentions to the coordinator. Repeat Step (3)-(5) until no collision or deadloop can be detected by the coordinator.

6. Each robot moves to their intended node. If the goal node is achieved, then the algorithm is stopped for that robot. Otherwise, go back to Step (1) and repeat Step (1)-(6) until all robots achieve their goals.

Algorithm 1 Super A* Algorithm

Input: Input two nodes n_0 (Start Node), n (Goal Node) and L (Robot Label)
Output: Output a set of nodes $N_{close}, n_i \in N_{close}$

$N_{open} \leftarrow n_0$, $N_{close} \leftarrow \emptyset$, flagged $\leftarrow$ **false**
loop
  $pn \leftarrow$ Compute the lowest cost of node, in N_{open}
  if $pn = n$ **then**
    $N_{close} \leftarrow N_{close} \cup \{pn\}$
    Return N_{close}
  else
    for all neighbours n_{new} **do**
      if L accords with a fixed priority scheme **then**
        flagged $\leftarrow$ Check Deadloop and five Collision types
      end if
      if flagged is **true then**
        Continue Loop
      end if
      if $n_{new} \in N_{open}$ and $G_{new} <$ current G **then**
        $G \leftarrow G_{new}$
        Continue Loop
      end if
      if $n_{new} \in N_{close}$ and $G_{new} <$ current G **then**
        $G \leftarrow G_{new}$
        Continue Loop
      end if
      $N_{open} \leftarrow N_{open} \cup n_{new}$
    end for
  end if
  $N_{close} \leftarrow N_{close} \cup \{pn\}$
end loop

3. DEMONSTRATIONS

We have conducted both real robot and simulator-based simulations. Through these simulations, we try to evaluate the following three aspects of the proposed strategy: (1) Practicability: is the strategy relevant and applicable to real robot systems? (2) Solvability: can the strategy find valid collision-free multi-robot paths? (3) Optimality: is the strategy able to generate the best paths despite collision-avoidance behaviour? Experiments were carried out using different configuration environments. The proposed method is applied to a two-robot system with the deadloop and collision conditions described in Section 1. The video link is http://youtu.be/gEHRxpbD_LY.

3.1 Demo 1: Real Robot Application

The goal of the first demo is to demonstrate the applicability of our approach to real robot systems. This experiment has been carried out using the Rovio robots of WowWee Technologies. The task of the robots is to find the optimal / shortest path and move from their initial positions to their goal positions without collision. In the demo, the robots are deployed on two sides of the 5x5 grid and have to move to their goal positions on the other side using the proposed strategy avoiding deadloop (Figure 2) and collisions. The demo shows that one robot moves away from its optimal path given the initial situation. The path of one robot is changed to avoid collisions between each other. For collision avoidance, one robot has to take a detour around the other and then return to its planned path as quickly as possible. At the end, the two robots achieve their goal positions. This is the resolution to this possible collision without introducing a sideways move or a collision involving front-end swipe or sideswipe, given that the robots are as big as the nodes they occupy.

3.2 Demo 2: Simulation

In demo 2, three scenarios are simulated in a 10 by 10 grid. The first scenario demonstrates deadloop and all five collision types, with dynamic path replanning demonstrated according to steps 1-6 above. It can be seen that R1 lets R2 take the optimal path, and R1 selects an avoidance strategy that allows it to return to the optimal path after collision is avoided. In addition, the proposed strategy can also cater for a combination of possible collisions. For instance, if a dynamic change to one or more robots' goal states leads to a deadloop condition, the proposed strategy 1-6 can resolve the problem effectively. The second scenario shows the tunnel-like environment, where two robots need to pass through a tunnel to reach their goals. R1 gets to the tunnel first, so R1 gets the priority to go through the tunnel. This is an example of allocating priority based on time. That is, the proposed strategy not only considers the optimal path cost but also takes optimal time cost into account. Finally, the last scenario shows that, with randomly changing goals in real-time, the proposed strategy is capable of avoiding collisions and returning to the planned optimal path for the new goal nodes. A 50x50 grid for 20 and 50 robots, respectively, with 10% obstacle density randomly generated environment.

4. CONCLUSION AND FUTURE WORK

In real-world multi-robot systems, deadloop and collision types must be clearly identified and managed to ensure that robots reach their destinations as optimally as possible. The proposed strategy, according to our real-world and simulated experiments, is robust and able to handle the deadloop and collisions effectively. We have also shown that the strategy is capable of dealing with both static and dynamic obstacles, and allows robots to resume their planned paths after collision avoidance. In future work, we will design a decentralized approach which can allow robots to achieve peer-to-peer communication for collision avoidance as well as investigate optimality preservation in more detail.

5. REFERENCES

[1] M. Jansen and N. Sturtevant. Direction maps for cooperative pathfinding. In *AIIDE*, pages 185–190, October 2008.

[2] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Publisher, Berlin, New York, 1982.

[3] D. Silver. Cooperative pathfinding. In *AIIDE*, pages 117–122, June 2005.

[4] T. Standley. Finding optimal solutions to cooperative pathfinding problems. In *AAAI*, pages 173–178, July 2010.

Decentralised stable coalition formation among energy consumers in the smart grid

(Demonstration)

Filippo Bistaffa,
Alessandro Farinelli,
Meritxell Vinyals
Department of Computer Science
University of Verona, Italy
{meritxell.vinalssalgado,alessandro.farinelli}@univr.it

Alex Rogers
Agents, Interaction and Complexity Group
School of Electronics and Computer Science
University of Southampton,UK
acr@ecs.soton.ac.uk

ABSTRACT

The vision of the Smart Grid includes demand-side peak shaving strategies, such as real-time pricing or profile's based tariffs, to encourage consumption such that the peaks on demand are flattened. Up to date, most works along this line focused on optimising via scheduling of home appliances or micro-storage the individual user consumption. Alternatively, in this demonstration we propose to exploit the consumers social side by allowing them to self-organise into coalitions of energy users with complementary needs. To this ends, we present an agent-based Java simulation of a social network of energy consumers (based on the domestic electricity market and usage patterns of homes in the UK) that uses to converge to stable energy coalitions.

Categories and Subject Descriptors

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

General Terms

Algorithms, Economics, Experimentation

Keywords

Decentralised Coalition formation, Stability, Smart grid, Energy

Online Material

`http://www.youtube.com/watch?v=FT25oETMkfw`

1. INTRODUCTION AND GOALS

Since energy cannot be stored efficiently on a large scale, the electricity grid must perfectly balance the supply to all customers at any instant with demand. In all current electricity grids this balance is achieved by varying the supply-side to continuously match demand. The amount of demand required on a continuous basis is usually carried by the baseload stations owing to low cost generation, efficiency and safety. However, these stations are slow to fire up and cool down, so they are not able to match the peakload periods that exceed this baseload. This requires the use of expensive, carbon-intensive, peaking plants generators. Although only running when there is high demand, these peaking plants generators are responsible for a significant fraction of the consumers total electricity bill.

To address this, the vision of the Smart Grid includes demand-side peak-shaving strategies such as real-time pricing or profile based tariffs to encourage consumption such that the peaks on demand are *flattened* [1]. Flatter demand results in a more efficient grid with lower carbon emissions and also with lower prices for consumers. Hence, recent works has focused on techniques that flatten individual consumer demand by automatically controlling home domestic or micro-storage devices [3, 4]. However, since each consumer independently optimizes its own consumption, the effectiveness of this approach has a clear limit on the consumer's restrictions and comfort (e.g. it is impossible completely avoid a consumption peak in the non-working hours).

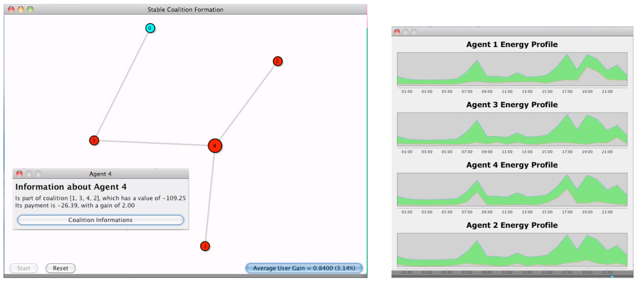
Against this background, in this paper we show how grid efficiency can be further improved from a social perspective. In particular, we explore the idea of allowing consumers to join into coalitions with other consumers with complementary energy needs. Then, a coalition of consumers can act in the market as a single virtual consumer with flattened demand, for which it gets much better prices. As part of the smart grid community, electricity consumers have already access to smart meters that allow them to monitor their (load) energy profile [1] on an hour-day basis. Moreover, given the huge recent success of social networks (e.g. at the time of writing Facebook has more than 500 millions users), consumers can potentially use them as free interaction tools to self-organise into energy coalitions.

2. THE SOLUTION APPROACH

We model the decentralised energy coalition formation problem as a coalitional game [2]. Let $C = \{c_1, \ldots, c_n\}$ be a set of energy consumers and F the set of feasible coalitions among these consumers. Any feasible coalition $S \in F$ is defined as a subset of consumers $S \subseteq C$. Then, a game is completely defined by its characteristic function v which assigns a real value to every feasible coalition. In a game

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

[1]The load energy profile is a graph of the variation in the electrical load versus time.



(a) (b)

Figure 1: Snapshots of a) the simulator main interface and b) the coalition energy profile inspector.

we aim to identify the coalition structure[2] that maximizes the efficiency of the system - i.e. the coalition structure with maximal value, $CS^* = \max_{\{CS\}} v(CS)$. Moreover, we needed to specify the following activities that take place in a coalitional game for this particular energy domain:

Coalitional Value Calculation. The value of a coalition S, $v(S)$, is the total payment that the set of consumers need to carry out to cover the demand of their joint energy profile[3]. Analogously to the operation of the current grid, we consider that consumers buy their electricity directly in two different markets: the forward market and the day-ahead market. In the forward market, consumers in a coalition S buy in advance the fixed baseload of energy of their joint energy profile, $base(S)$, for a better price. The amount of energy that exceeds this baseload, $peak(S)$, is bought in the day-ahead market. In particular, the value of a coalition S is given by:

$$v(S) = -base(S) \cdot p_F - peak(S) \cdot p_{DA} \quad (1)$$

where p_F and p_{DA} are the unit energy price in the forward and the day-ahead market respectively.

Since $p_F < p_{DA}$, the flatter the energy profile, the most a coalition of consumers can buy in the forward market and the lower the payment of the coalition.

Network-based coalitions. Social networks not only provide a way of interaction among energy consumers but also restrict coalition membership by reflecting realistic barriers to the formation of certain coalitions. In particular, consumers may not want to join coalitions with unknown consumers for whom they do not have any source of trust regarding their reported profiles or their capacity to meet their payment obligations. In contrast, if each consumer looks for potential partners for its coalitions through its contacts in a social network coalition membership is restricted to coalitions composed of *friends of friends*, such that there is always somebody responsible in the coalition for the introduction of a new member.

Payoff Distribution. Consumers in a coalition are permitted to freely distribute the coalitional value among themselves. Thus, in addition to the set of optimal coalitions, CS^*, the outcome of the game also needs to specify a payoff vector $\rho = \{\rho_1, \ldots, \rho_n\}$ that divides the value of optimal coalitions among consumers ($\sum_{c_i \in C} \rho_i = v(CS^*)$). However, since consumers are selfish, the value of a coalition should be distributed among its members in such a way that coalition members have no incentive to break away from the identified efficient coalition. When this happens, we say that payments are stable[4]. To be stable, these set of payoffs needs to make sure that there is no other outcome that can make a set of consumers better-off ($\forall_{S \in F} : \sum_{c_i \in S} \rho_i \geq v(S)$).

[2]A *coalition structure* is an exhaustive disjoint partition of the space of consumers into feasible coalitions.

[3]The joint energy profile is computed as the aggregation of individual energy profiles.

3. THE PLATFORM

As a response to these challenges we have developed a platform that allows energy consumers to organise into stable energy profile coalitions. The interface is shown in Figure 1. The demonstration starts by asking the user the number of energy consumers for the simulation. Moreover, the user can choose between creating the social network randomly, or, alternatively, create a user defined social network from scratch. In both cases, the platform generates a set of nodes (see Figure 1(a)), one per energy consumer, and allows the user to modify the network by adding/removing links in an easy way. Each node has an energy profile loaded from real data characterizing the domestic electricity market and usage patterns of homes in the UK.

Once the coalition formation scenario is set, the simulation starts a message-passing algorithm that organises energy consumers into stable optimal coalitions. Upon convergence, energy consumers in the same coalition are coloured with the same colour. For example, observe that in Figure 1(a), consumers 1, 2, 3, and 4 form an energy coalition whereas consumer 0 is on its own. On the right lower corner, the application also shows the average consumer gain - that is the gain that represent the consumer assigned payment with respect to the value of its singleton coalition. By clicking on a node, the GUI displays statistical data related to the specific energy consumer such as its coalition, the coalition's value and its (stable) individual payment. The platform also allows to visualize the energetic profiles of coalition members (see Figure 1(b)). Finally, the GUI allows the user to testing how the existence/nonexistence of a particular link affects the emerging coalitions and consumers gain by reconfiguring the network and restart the simulation.

As a simulator, this platform provides users with a proof of concept of what we can do already today as energy consumers in order to get cheaper and greener energy. Furthermore, it presents the decentralised coalition formation problem among energy users as an exciting real-world domain for the applicability of multi-agent technology.

4. REFERENCES

[1] Grid 2030. a national vision for electricity's second 100 years. Washington DOE, 2003. http://www.oe.energy.gov/DocumentsandMedia/Electric_Vision_Document.pdf.

[2] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge MA, USA, 1994.

[3] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Agent-based control for decentralised demand side management in the smart grid. In *AAMAS*, pages 5–12, 2011.

[4] S. L. Tompros, N. P. Mouratidis, M. Draaijer, A. Foglar, and H. Hrasnica. Enabling applicability of energy saving applications on the appliances of the home environment. *IEEE Network*, 23(6):8–16, 2009.

[4]We focus on the core as main stability solution concept.

Learning to be Scientists via a Virtual Field Trip (Demonstration)

Deborah Richards[1], Michael J Jacobson[2], Meredith Taylor[1], Anne Newstead[2], Charlotte Taylor[2], John Porte[1], Iwan Kelaiah[1], Nader Hanna[1]

[1]Department of Computing
Macquarie University
North Ryde, NSW, 2109, Australia
+61 2 9850 9567
deborah.richards@mq.edu.au

Centre for Computer Supported Learning and Cognition, Faculty of Education and Social Work, The University of Sydney, NSW, 2106, Australia
+61 2 9036 7671
michael.jacobson@sydney.edu.au

ABSTRACT
We have developed the virtual world of Omosa in which school students can learn what scientists do by doing it themselves. In Omosa students are able to observe, collect data and interact with a number of intelligent virtual human and animal agents.

Categories and Subject Descriptors
I.2 ARTIFICIAL INTELLIGENCE; I.6 SIMULATION AND MODELING, I.6.3 [**Applications**] I.6.7 [**Simulation Support Systems**] *Environments*

General Terms
Design, Experimentation.

Keywords
Agents, artificial life, boids, educational virtual worlds, biology education, science inquiry.

1. INTRODUCTION
Genuine scientific inquiry is rare in the classroom. Reasons for this include the reluctance of teachers to engage in genuinely open-ended inquiry arising out of pressures to create efficient learning trajectories and cover all topics in a mandated curriculum. This difficulty is exacerbated by a science curriculum that has become theory and textbook heavy due to resource limitations and occupational health and safety (OH&S) issues. In particular, Zappala [6] notes that teaching behavioural ecology and ethology (the scientific study of animal behavior) is limited by physical, practical and ethical constraints such as: confounds and control of extraneous variables; observer bias leading to data tainting; difficulty of capturing rare events and behaviours; and infeasibility of large scale or long term study. Furthermore, while laboratory conditions can provide consistency and repeatability, for many species this approach may be undesirable or inappropriate"[i]. The ability to conduct a virtual field trip can address many of these issues and provide an opportunity for students to gain knowledge and skills needed for scientific inquiry such as hypothesis formation and testing, designing experiments, conducting investigations, using secondary resources and data, using equipment and ICT, managing risk, collecting data, performing analysis and drawing and communicating conclusions.

In addition to providing a hands-on and experiential approach to honing students' scientific inquiry skills, we are also interested in teaching students about complex systems such as ecosystems. Multiagent systems are particularly suitable for this purpose because while each individual agent may follow a small set of rules complex behaviours at the group level tend to emerge. To this end, we have developed the virtual world of Omosa in which school students are able to observe and interact with intelligent virtual human and animal agents.

Appears in: *Proc of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2. PROBLEM SCENARIO
In Omosa World the students, as junior scientists, are invited by the Chief Scientist at the IEIA (Interplanetary Environmental Investigation Agency) to assist in discovering why planet Omosa has been showing signs of ecosystem change. The indigenous people who live there have reported that the populations of certain species of animals, including those that are an important food source in their society, are declining. The Omosans have agreed to allow scientists to come and study the situation.

Students utilise workbooks to explore different issues and record their findings. Some activities occur in the world (e.g. speaking to the climatologist, hunter or ecologist agents and observing the animal agents); some in the classroom (e.g. proposing a hypothesis and describing the experiment to be conducted to the whole class). There is a progression in concept development as students move from one problem to another.

3. THE VIRTUAL WORLD TECHNOLOGY
Omosa has been developed using the multi-platform game development environment called Unity3D (unity3d.com/) that has inbuilt graphics and physics engines and features such as lightmapping and occlusion culling. To create the Omosan landscape itself and its base texture we used L3DT, a terrain generating tool (www.bundysoft.com/L3DT/). We then imported the heightmap into Unity3D and used its terrain editing tool to add the grass and trees. This tool makes it easy to place details onto a terrain and remove them if the size of the game becomes too large. The island of Omosa contains four main locations (village, hunting ground, weather laboratory, research station) where students can collect information and complete activities. We used Blender (www.blender.org/) to model structures and Mixamo (www.mixamo.com/) to create low polygon human models. We purchased our animals from TurboSquid (turbosquid.com/), where 3D artists can sell models. The models on TurboSquid usually have a high polygon count, which is especially undesirable when creating large herds of animals. We purchased four animals from the same artist, 3Drivers (www.turbosquid.com/Search/Artists/3DRivers). Three of the models are of extinct animals (Andrewsarchus, Bluebuck, and Indricotherium) and one is of an Iberian Lynx. Each had more than 6000 polygons. Using Blender we reduced the number of polygons to no more than 1800 each.

4. OUR AGENTS

The human and animal agents in Omosa are not directed or led by any other agent. However, the rules which drive them, particularly those relating to their predator and prey roles, result in numerous emergent group behaviours. Part of the behavior we wished to simulate in our animals was their tendency to live and move in flocks or herds and the emergent behavior that this demonstrates. To achieve this we started with an implementation of the Boids flocking algorithm [4]. Fundamentally the original Boids algorithm involves the summation of multiple vectors (separation, alignment and cohesion) to achieve a single output vector which determines the direction for each boid. These vectors represent the intentions of the boid. We rigged and animated our animals in Blender and imported them into Unity3D where we applied our modified Boid algorithm and Unity3D's physics engine. This presented another challenge, namely balancing the processing requirements of the modified Boid algorithm with the processing requirements of the graphics. We have solved this primarily by increasing the number of frames between each animal in the herd updating its behavior. This slightly decreases the realism of the animals' behavior but significantly improves game performance.

Other virtual worlds for scientific exploration exist, such as Quest Atlantis [1] and Virtual Singapura (VS) [2]. The intelligent predator-prey and flocking behaviours of our animals allows students to conduct observation as they might in the field (see Figure 1). This distinguishes our work significantly from other projects; changing the data gathering process from reading agent/avatar dialogues to experiential learning.

Figure 1. Tooru (predator) feeding on a Yernt (prey)

Figure 2. Hunter agent beckons user to come over

5. INTERACTING WITH OUR AGENTS

Students are able to interact with our agents in a number of ways. Firstly, as the user approaches, the agent will wave (see Figure 2). If the user walks up the agent, they are able to click on the speech bubble (as shown on the waving hunter) to initiate a dialogue with agent. Dialogue is basic and allows the user to select questions they wish to ask the agent. Questions are ordered/included based on the prior activities and demonstrated knowledge of the student. Additionally, users can select/collect data and items and observe the Omosans and animals in their natural settings. Students are able to perform laboratory-based prey-predator simulations using NetLogo [5]. However, to assist with development and balancing of our ecosystems involving the 3D animal agents, numerous parameters can be adjusted (see Figure 3). Interactions between animal agents are described in our full paper in this proceedings.

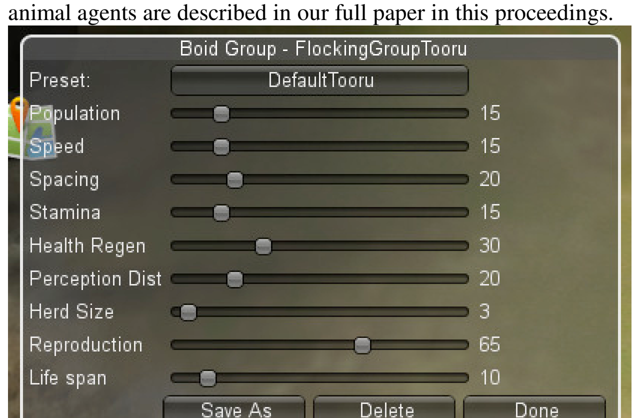


Figure 3. Predator-Prey Agent Parameters .

6. EVALUATION & FUTURE WORK

To ensure the accuracy of the behaviours of our animals in Omosa we 1) based our animal agents on Wilensky's Wolf Sheep Predation model [5] 2) included biologists on our team and 3) had an independent animal communication and conservation expert evaluate and tweak our animal behaviours. To design and evaluate Omosa as a learning environment we consulted biologists and secondary science teachers, performed an initial pilot with an afterschool science special interest group and used our world and workbooks over a two week period in late 2011 with around 50 Year 9 children in two classes (one comprehensive and one selective) involving four 50 minute lessons for each class.

In 2012 we will be adding new scenarios and schools. From an educational research standpoint we are conducting classroom based experiments on the merits of *productive failure* [3]. From an agent standpoint, we will focus on collaborative learning involving agent-human and human-human collaboration. A demonstration of Omosa can be found at http://www.comp.mq.edu.au/~richards/aamas12Omosa/

7. REFERENCES

[1] Dede, C., Clarke,J., Ketelhut, D., Nelson, B. and Bowman, C. 2005. Students motivation and learning of science in a multi-user virtual environment. AERA, Montreal, Canada.

[2] Jacobson, M. J., Lim, S. H., Lee, J., & Low, S.-H. 2007. Virtual Singapura: Design considerations for an intelligent agent augmented multi-user virtual environment for learning science inquiry. 15th Int.l Conf. on Computers in Education.

[3] Kapur, M. 2008. Productive failure. Cognition and Instruction, 26(3), 379 - 424.

[4] Reynolds, C. 1987. Flocks, herds and schools: A distributed behavioral model. SIGGRAPH '87: Proc. 14th Conf. on Computer graphics & interactive techniques (ACM): 25–34.

[5] Wilensky, U. 1997. NetLogo Wolf Sheep Predation model. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation. Northwestern University, Evanston, IL.

[6] Zappala, J. 2008. Multi-agent simulation of group decision making in animals, Masters Thesis, University of Nottingham.

[ii] http://ima.ac.uk/slides/jxz-03-11-2009.pdf

Virtual Characters in Agent-Augmented Co-Space (Demonstration)

Yi-Lin Kang, Budhitama Subagdja, Ah-Hwee Tan, Yew-Soon Ong, Chunyan Miao
School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
{kang0028,budhitama,asahtan,asysong,ascymiao}@ntu.edu.sg

ABSTRACT

Co-Space refers to interactive virtual environment modelled after the real world in terms of look-and-feel, functionalities and services. We have developed a 3D virtual world named Nanyang Technological University (NTU) Co-Space populated with virtual human characters. In order to create realistic virtual humans, we have designed a brain-inspired agent architecture with the properties of goal-directed autonomy, natural interactivity and human-like personification. The demo will show how the virtual characters may enhance the interactivity and playability of the virtual worlds.

Categories and Subject Descriptors

I.2.0 [**General**]: Cognitive simulation; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Design, Human Factors, Languages,Theory

Keywords

Virtual characters, autonomy, interactivity, personality

1. INTRODUCTION

Virtual worlds has become a popular platform used in a variety of contexts, including education, business, and e-commerce. We are particularly interested in a special class of virtual world, called Co-Space, referring to interactive virtual environment modelled after a real physical space in terms of look-and-feel, functionalities and services.

Besides providing faster and easier access to information and services, the development of Co-Space has offered great opportunities for innovative applications. In particular, intelligent agents can be deployed in Co-Space enhancing its interactivity and playability. We have developed a 3D virtual world called NTU Co-Space, modelled after the Nanyang Technological University (NTU) campus and populated with virtual human characters. This demo shall show how virtual humans, designed based on a brain-inspired agent model, may enhance the interactivity and playability of the virtual world in a natural manner.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2. NTU CO-SPACE

The NTU Co-Space is implemented using the Unity3D, a 3D game engine that can be deployed into different platforms including Microsoft Windows™, Mac OS™, popular game consoles, and mobile devices. The Co-Space can also be embedded in a web page to be easily accessed using typical web browsers (e.g. IE, Chrome, Safari, Firefox).

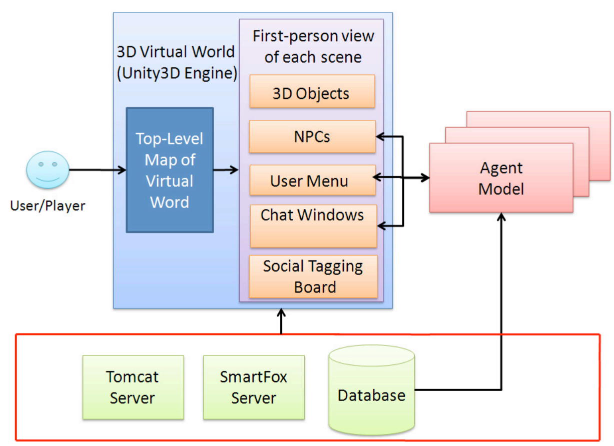


Figure 1: System architecture of NTU Co-Space.

The overall architecture of the NTU Co-Space is depicted in Figure 1. At the back-end, two application servers, Apache Tomcat and SmartFox, support the multi-user environment. A dedicated database is used for the storage and retrieval of environmental and content data.

3. THE AGENT TECHNOLOGY

As shown in Figure 2, the integrated agent architecture consists of a *Perception Module* receiving situational signals from the environment through a set of sensory APIs and an *Action Module* for performing actions through the various actuator APIs. If the sensory signals involve a text input, the *Chat Understanding Module* interprets the text for the player's intention. The outputs of *Situational Assessment* and *Chat Understanding Modules* then serve as part of the working memory content . The *Inference Engine* then identifies the most appropriate action, by tapping a diverse pool of knowledge, in accordance to the desire, intention and personality of the virtual agent. The knowledge learned and used by the Inference Engine include declarative knowledge of self, players, and environment, as well as procedural knowledge of goal-oriented rules, which guide an agent in fulfilling goals, and social rules, for generating socially appropriate behavior. The decision of the *Inference Engine* again forms part of the *Working Memory*, which throughout maintains the context of the interaction. For actions involving a verbal response, the *Natural Language Generation Module* translates the chosen response into natural text for presentation.

Consistent with the view in the state of the art [1], we outline three key characteristics of realistic characters in virtual worlds, namely autonomy, interactivity, and personification, described as follows.

Autonomy Based on a family of self-organizing neural models known as fusion Adaptive Resonance Theory (ART) [3], the *Inference Engine* of the proposed agent architecture performs a myriad of cognitive functions, including recognition, prediction and learning, in response to a continual stream of input signals received from multiple pattern channels. As a result, an agent makes decisions not only based on the situational factors perceived from the environment but also her mental states characterized by desire, intention and personality. By modelling the internal states of individual agents explicitly, the virtual humans can live a more complete and realistic life in the virtual world.

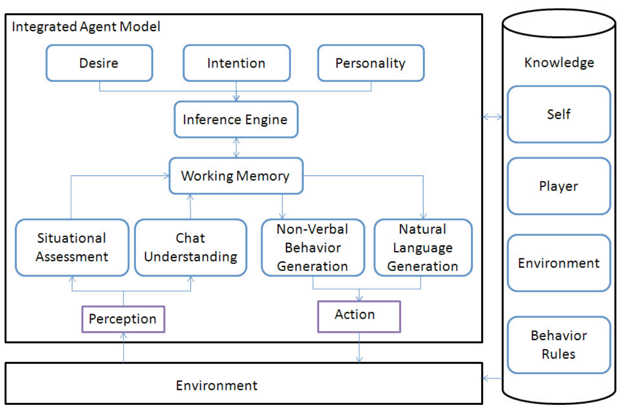


Figure 2: A schematic of the integrated agent model.

Interactivity For interaction between the agents and the players, an intuitive user interface is provided, through which a player may ask typical questions and provide quick responses by button clicks. The player may also enter free-text sentences via the chat box. The dual communication mode provides the players both ease of use and flexibility. While interacting with player, the agent builds an internal model of the player, with his/her profile, interests and preferences. The player model in turns allows the agent to make intelligent conversation on topics relevant to the player.

Personification For improving the believability of virtual humans, our agents adopt the Five Factor Model (FFM) [2], which characterizes personality in five trait dimensions . By giving a weighage to each dimension, a unique personality can be formed by a combination of the traits. Comparing with traditional pattern-matching-based conversational agent, our agents with strong *openness* and *extroversion* personality are much more warm and friendly as they do not stay idle and wait for input queries. Acting pro-actively, they approach the players, offer help, and make conversation.

4. DEMONSTRATION DESCRIPTION

A video clip of the NTU Co-Space can be viewed on You Tube (http://www.youtube.com/watch?v=bYIthOYjrxw). During the live demo, multiple players will be able to log in and experience the NTU Co-Space. As illustrated in Figure 3, a player may choose to roam around the campus on his/her own for self-discovery or play a mini-game, called *Amazing Quest*, which will bring the player to experience the key places of the NTU campus through a fun and interactive journey. Specifically, the player will visit the five key landmarks on campus, with the accompaniment of a virtual character.

Figure 3: A player touring the NTU Co-Space.

5. CONCLUSIONS

With the virtual characters befriending and providing personalized context-aware services, we hope players will find virtual worlds more fun and appealing. To the best of our knowledge, this is perhaps one of the few in-depth works on building complete realistic agents in virtual worlds with autonomous behavior, natural interactivity and personification.

Finally, it is also our objective that the NTU Co-Space may serve as a open platform for agent researchers to deploy and field test their technologies. A set of APIs with documentation have been made available to ease the integration effort. For a wider accessibility of the Co-Space content and services, implementation of mobile clients running on iPhone and iPad are already under way.

6. ACKNOWLEDGEMENT

This research is supported by the Singapore National Research Foundation under its Interactive & Digital Media (IDM) Public Sector R&D Funding Initiative and administered by the IDM Programme Office.

7. REFERENCES

[1] Z. Kasap and N. M. Thalmann. Intelligent virtual humans with autonomy and personality: State-of-the-art. *Intelligent Decision Technologies*, 1:3–15, 2007.

[2] R. Mccrae and P. Costa. An introduction to the five-factor model and its applications. *Journal of Personality*, 60:172–215, 1992.

[3] A.-H. Tan, G. A. Carpenter, and S. Grossberg. Intelligence through interaction: Towards a unified theory for learning. *Lecture Notes in Computer Science*, 4491(I):1098–1107, 2007.

ARGUS: A Coordination System to Provide First Responders with Live Aerial Imagery of the Scene of a Disaster (Demonstration)

F. M. Delle Fave, A. Rogers and N. R. Jennings
University Of Southampton
{fmdf08r, acr, nrj }@ecs.soton.ac.uk

ABSTRACT

We present ARGUS, a coordination system for unmanned aerial vehicles (UAVs) deployed to support situational awareness for disaster management settings. ARGUS is based on the max-sum algorithm, a well known decentralised coordination algorithm for multi-agent systems. In this demonstration, we present an interactive simulation environment, where a user acting as a first responder submits imagery collection tasks to a team of UAVs, which then use max-sum to assign themselves to the tasks. We then present a set of real flight tests, in which two Hexacopter UAVs again use ARGUS to coordinate over tasks. Our tests indicate that the system responds positively to the dynamism and the heterogeneity of the real world.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems, coherence and coordination*

General Terms

Algorithms, Experimentation

Keywords

Simulation, Coordination, Unmanned Aerial Vehicles

1. INTRODUCTION

Current research in artificial intelligence is dedicating great effort in developing technologies for disaster management (see, for instance, the ALADDIN project[1]). In such settings, first responders need to quickly assess the severity of a disaster in order to prioritise intervention. To this end, the deployment of autonomous vehicles, such as unmanned aerial or ground vehicles (UAVs and UGVs) is highly recommended, since these can provide information inaccessible to humans, either because they are able to fly or because they can reach dangerous areas. Such vehicles then should be capable of gathering such information in an efficient and timely fashion, without relying on valuable and scarce human resources to control them [2]. Thus, having them coordinate their decision making autonomously is a key factor to achieve an effective situational awareness. To this end, a variety of coordination algorithms has been studied in the literature, among which decentralised ones are typically preferred due to their scalability and their robustness to component failure [3]. In particular, the max-sum algorithm has been shown to perform well in a variety of simulated problems while requiring very little communication and computation [5].

However, despite its demonstrated potential, thus far, max-sum has not been deployed in real application domains. It has only been tested in simulation, which lacks the dynamism and the heterogeneity of the real world. Hence, to ascertain its effective performance, this paper introduces ARGUS a coordination system where the max-sum algorithm is deployed to coordinate a team of UAVs to provide live aerial imagery to the first responders operating in the area of a disaster.

The remainder of this paper is organised as follows. Section 2 describes the problem; Section 3 introduces ARGUS and Section 4 concludes.

2. PROBLEM DESCRIPTION

At the scene of a disaster, first responders require up to date imagery to assess the situation. ARGUS provides such imagery by using a team of UAVs deployed over the area, each equipped with a miniature video camera that can stream live video over a short range wireless link. The first responders interact with the UAVs using a personal digital assistant (PDA), to request imagery collecting tasks. Each task T_i represents a location (in geographic coordinates) for which imagery is required. To submit a task, each first responder sets three properties: (i) a priority $p_i = \{normal,\ high,\ very\ high\}$, representing the importance of the task (i.e. collecting imagery of an occupied building is more important than doing so for an empty one); (ii) an urgency $u_i = \{normal,\ high,\ very\ high\}$ used to prevent tasks with low priority from remaining unattended (i.e. collecting imagery from an evacuated building is less important than doing so for a burning building but needs to be done) and (iii) a duration d_i, which defines the interval of time for which imagery needs to be collected. Note that a first responder does not know this duration with precision since it depends on the specific reason for which imagery is required (e.g. to search for a casualty or to check access to an area). Thus, three estimates are considered ($d_i = \{5\ min,\ 10\ min,\ 20\ min\}$). In order to complete a task a UAV is required

[1] *http://www.aladdinproject.org/*

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

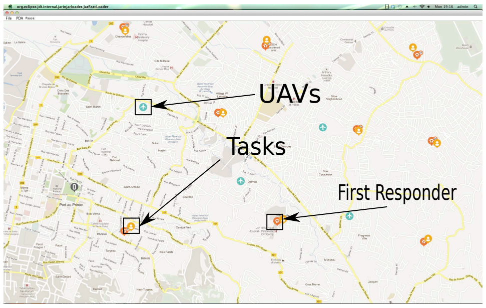
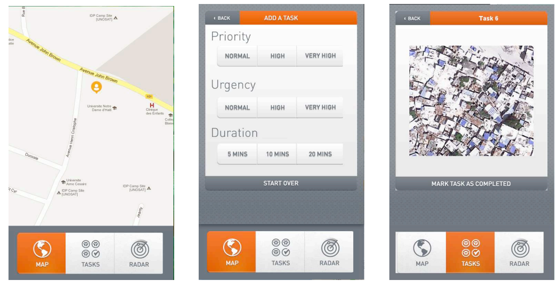


Figure 1: The area of a disaster where the first responders (FPs in the figure) and the UAVs operate, as generated by the software simulator.



(a) The Map Interface (b) The task input interface (c) The imagery collection interface

Figure 2: The simulation software representing the PDA's interface

to fly to the specified location, station itself above it and stream live video to the PDA until the first responder indicates that the task is complete. The aim of the UAVs is then to *jointly* decide which task each vehicle should complete.

3. THE ARGUS SYSTEM

A video describing the system is provided at *http://vimeo.com/user9939345/videos*. In what follows, we describe the two ways in which we evaluate our system.

3.1 The Simulation Environment

Each user acts as a first responder and is given a PDA running a touch screen system to submit tasks to the the UAVs operating in the area of a disaster (Figure 1). The system is composed of three interfaces that allow the user to (i) submit tasks by selecting any location within his local area (Figure 2(a)), (ii) to specify the properties of the imagery collection task (Figure 2(b)) and (iii) to view the live streaming videos of the tasks that are being attended by a UAV (Figure 2(c)). These tasks can appear at anytime. Thus, the UAVs need to constantly revise their decisions over which ones to attend, and thus, they need to continuously coordinate. This happens in a decentralised fashion by using the max-sum algorithm[2]. The main aim is to maximise the number of completed high importance tasks (i.e. those with a very high importance) given the limited battery capacity of the UAVs which have to periodically leave the scene to recharge. For a more thorough description of the algorithm used in this dynamic optimisation setting see [1].

3.2 The Flight Tests

The ARGUS system was deployed on two Mikrokopter Hexacopter UAVs over three different settings (see [1] for more details) to ascertain its performance in the real world. The flight tests were run at a test facility outside of Sydney, in conjunction with the Australian Centre for Field Robotics (ACFR). A video summarising the flight tests can be found at *http://vimeo.com/user9939345/videos*. In the video (Figure 3), windows *A* and *B* show the hexacopters, window *C* shows the computation over the factor graph over which max-sum is running and window *D* shows the path of the UAVs.

4. CONCLUSIONS AND FUTURE WORK

In this paper we described ARGUS a coordination system for UAVs deployed to support situational awareness for disaster management settings. ARGUS is based on the max-sum algorithm, which, thus far, has been deployed only on simple simulated environments. The system was evaluated in two ways. First an interactive simulation environment was developed where first responders can submit imagery collection requests to a team of UAVs. Second, a set of real flight tests were performed to evaluate its performance in the real world. These tests indicated that the system responds positively to the dynamism and the heterogeneity of the real world. Thus they show that max-sum is a powerful technique to use to coordinate teams of UAVs for disaster management.

Our future work will be focused on scaling-up the system to consider a large number of UAVs and tasks.

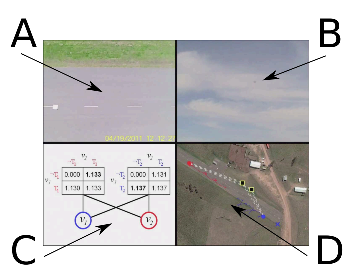


Figure 3: A snapshot of the video summarising the three flight tests.

5. REFERENCES

[1] F. M. Delle Fave, A. Rogers, Z. Xu, S. Sukkarieh, and N. R. Jennings. Dynamic decentralised task assignment for teams of unmanned aerial vehicles. In *Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012. (in press).

[2] M. R. Endsley. *Situation Awareness, Analysis and Measurement*. Lawrence Erlbaum Associates, 2000.

[3] J. How, C. Fraser, K. Kulling, L. Bertuccelli, O. Toupet, L. Brunet, A. Bachrach, and N. Roy. Increasing autonomy of uavs. *IEEE Robotics & Automation Magazine*, 16(2):43–51, 2009.

[4] K. Macarthur, R. Stranders, S. D. Ramchurn, and N. R. Jennings. A distributed anytime algorithm for dynamic task allocation in multi-agent systems. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI)*, pages 356–362, 2011.

[5] A. Rogers, A. Farinelli, R. Stranders, and N. R. Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

[2] We adopt here a modified version of the algorithm to reduce the computation and communication in task assignment domains. See [4] for more details.

Pogamut Toolkit (Demonstration)

Jakub Gemrot
Charles University, Faculty of Mathematics and Physics
Ke Karlovu 3
Prague 2, 121 16, Czech Republic
jakub.gemrot@gmail.com

Michal Bída
Charles University, Faculty of Mathematics and Physics
Ke Karlovu 3
Prague 2, 121 16, Czech Republic
michal.bida@gmail.com

Cyril Brom
Charles University, Faculty of Mathematics and Physics
Ke Karlovu 3
Prague 2, 121 16, Czech Republic
brom@ksvi.mff.cuni.cz

ABSTRACT

All experiments using intelligent virtual agents, sooner or later, ask for a specific virtual environment that would fit their setup. Seeking such environment is a daunting task accompanied with the need for an appropriate agent adapter that provides infrastructure for mediation of virtual body senses and actions thereby enabling remote high-level agent control. This demo presents Pogamut toolkit, which provides out-of-box programmer tools for creating virtual agents for Unreal Tournament 2004, Unreal Development Kit and Defcon virtual environment. Pogamut's virtual world abstraction is compatible with many agent oriented languages and architectures including Jadex, GOAL, POSH, Soar or ACT-R, which makes it highly suitable for research on intelligent virtual agents.

Categories and Subject Descriptors

D.2.13 [**Reusable Software**]: Reusable libraries

General Terms

Design, Experimentation.

Keywords

IVA toolkit, Virtual environments, Action-selection

1. INTRODUCTION

The development of intelligent virtual agents (IVA) is still far from being easy as every IVA application calls for complex chain of tools and libraries that must work together to enable quick and efficient IVA production. IVA production typically comprises several cycles, during which researchers:

(a) design,
(b) implement, run, observe & debug,
(c) test & validate their IVAs.

Technically, IVA applications can be conceived as consisting of three parts (see Picture 1):

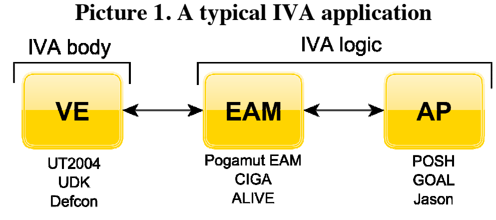
(1) a virtual environment (VE),
(2) an environment-agent middleware (EAM),
(3) an agent platform (AP).

Furthermore, as every research have to implement & debug (Point (b)) and test & validate the application (Point (c)), a researcher needs:

(4) implementation tools,
(5) debugging tools,
(6) testing & validation tools.

As there is no mature standard yet that would cover the whole IVA development process or provide research methodology guidelines and technology interface standards (contrary to "classical" agents, cf. e.g. FIPA), every IVA application setup requires a proprietary solution that combines Parts (1) – (6). Here, we present Pogamut toolkit, a result of 5 years of work, which aims at providing complete solutions for building IVAs for various virtual environments. Pogamut toolkit currently supports development of IVAs for (i) Unreal Tournament 2004 (UT2004), (ii) Unreal Development Kit (UDK) and (iii) Defcon. Unreal Tournament 3 (UT3) is a work-in-progress. The toolkit complements similar attempts, such as [1] capitalizing on BML.

Picture 1. A typical IVA application



2. FEATURES OF IVA TOOLKITS

Instead of listing Pogamut features, it is better to review IVA production cycle (Points (a) – (c)) with respect to IVA application Parts (1) – (3). That will provide the list of features that every IVA toolkit should possess.

2.1 Designing IVAs (Point (a))

Process of designing an IVA is typically sensitive to the selection of Parts (1) – (3). A researcher has to understand capabilities, limitations and options of every part involved. She must understand a VE (1) to be able to create its particular instance suitable for the application; she has to work with an EAM (2) encoding agents' reflexes and complex sensory and motor primitives; finally, she will work with an AP itself (3), which will accommodate agents' plans and strategic decision making.

The support from the IVA toolkit here is to have getting-started tutorials, be well documented and provide a lot of executable example agents that exemplify various features provided by Parts (1) – (3).

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2.2 Implementing & Debugging (Point (b))

Once an IVA application is designed and the tool chain is understood, the implementation can take place. This phase itself will contain a lot of iterations of Point (b) (see Table 1).

Unfortunately, all steps of Point (b) will happen in each of VE, EAM and AP so the toolkit must provide (ideally integrated) (4) & (5) to help the researcher along the way. The list of desired features is presented in Table 1.

2.3 Testing & Validating IVAs (Point (c))

Once an IVA is implemented, it needs to be run through series of tests that provide data for answering experiment hypotheses, e.g., for comparison to other existing IVAs fulfilling the same goal. Usually, it means to run the IVA multiple times (e.g., 20x or 100x) to gain statistical validity of the obtained data.

The IVA toolkit has two roles in this process (Part (6)). First, it should provide means for gathering such data, e.g., stubs for agent observers that can collect data of agent actions, reasoning, decision making and a VE itself. Second, it should provide tools (GUIs, libraries, scripts) for automatic testing, so that the researcher does not need to run every test manually or create such tools.

Table 1. The list of IVA platform features that ease implementing & debugging of IVAs

	(1) VE	(2) EAM	(3) AP
Implement	VE editor	IDE for coding reflexes and complex sensory and motor primitives	IDE for creating agent plans
Run	Means for quick (re)starting of the whole tool chain (startup scripts or GUI).		
Observe and debug	VE visualizer	Interactive coding, sync. breakpoints with VE, logs.	Interactive coding, sync. breakpoints with EAM, logs.

2.4 Technical dependencies

Unfortunately, there are technical dependencies between a VE, an EAM and an AP. Thus every complete tool chain will contain a lot of "glue" code that adapts VE-EAM and EAM-AP. As there are no mature standards how VEs, EAMs and APs should look like, no one can expect (for instance) that existing tools for AP will provide much insight into interoperability between EAM and AP or even VE and AP. For example, an automated IVA testing tool that operates over UT2004-Pogamut-SPOSH (as part of (6)) will not work for Defcon-Pogamut-Jason setup as it will contain much of UT2004-Pogamut-SPOSH specific code.

This is not surprising but leads to another observation that every IVA toolkit should state which tools it provides with respect to concrete VE-EAM-AP chosen.

3. FEATURES OF POGAMUT

Tables 2 and 3 provide an overview of existing and implemented tool chains for creating IVAs for UT2004, UDK and Defcon environment by the Pogamut toolkit.

Table 2. Bindings that Pogamut as EAM provides.

VE / EA	Java	POSH	Jason	ACT-R
UT2004	Yes	Yes	No	Yes
UDK	Yes	Yes	No	No
Defcon	Yes	No	Yes	No

4. USAGE

In this paper we have presented a list of general features that are (has to be) common to every IVA toolkit aiming to support development of IVA applications. The crucial point is that Pogamut supports these features with respect to three different virtual environments. Furthermore, Pogamut already proved its applicability by being used for international IVA competition, research and education.

5. ACKNOWLEDGMENTS

This research was partially supported by project P103/10/1287 (GACR), by student grants GA UK No. 0449/2010/A-INF/MFF 655012/2012/A-INF/MFF, and by SVV project number 263 314.

6. REFERENCES

[1] Thiebaux, M., Marshall, A., Marsella, S., Kallmann, M. SmartBody: Behavior Realization for Embodied Conversational Agents. In: Proc. of Autonomous Agents and Multi-Agent Systems (2008).

Table 3. Existing tutorials and features of Pogamut toolkit for respective VE/AP combinations.

	Designing				Implementing & Debugging	Testing & Validating
VE / AP	Installer	Getting started doc.	Tutorials	Commented examples	IDEs / Tools	IDEs / Tools
UT2004+Java	Yes	Yes	Yes	Yes	NetBeans IDE, Debug GUI	Experiment runner lib.
UT2004+POSH	Yes	Yes	Yes	Yes	NetBeans IDE with POSH Editor and POSH Debugger	Experiment runner lib.
UT2004+ACT-R	Yes	No	Yes	Yes	NetBeans IDE	X
UDK+Java	Yes	Yes	Partially	Yes	NetBeans IDE	X
UDK+POSH	No	No	Partially	No	NetBeans IDE with POSH Editor and POSH Debugger	X
Defcon+Java	No	No	Yes	Yes	NetBeans IDE for coding, Auto deploy & run Ant scripts	X
Defcon+Jason	No	No	Yes	Yes	Auto deploy & run	X

An Intelligent Agent for Home Heating Management (Demonstration)

Alex Rogers, Sasan Maleki, Siddhartha Ghosh and Nicholas R. Jennings
Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
{acr,sm9g09,sg2,nrj}@ecs.soton.ac.uk

ABSTRACT

Intelligent software agents are increasingly being applied within the smart grid; a future vision of an electricity distribution network where information flows in both ways between between consumers and suppliers, and where electricity prices change in real-time in response to the current balance of supply and demand across the grid. In this demonstration, we show a home heating management agent that can learn the thermal characteristics of a home and predict local weather conditions, in order to provide home owners with real-time information about their daily heating costs. Furthermore, we demonstrate how the agent can then optimise heating use to minimise cost and carbon emissions whilst satisfying the home owners preferences for comfort.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence

General Terms

Design, Algorithms, Experimentation, Theory

Keywords

Agent, smart grid, electricity, heating optimisation

1. INTRODUCTION

The creation of a smart electricity grid has been posed as one of the greatest engineering challenges of this century, as countries face dwindling non-renewable energy sources and work to minimise the adverse effects of climate change due to carbon emissions [1]. To this end, the UK government has committed to reducing carbon emissions by 80% by 2050, and central to achieving this aim is the mandated roll-out of smart meters to all 26M UK homes by 2020, and support for the electrification of heating through the installation of air and ground source heat pumps [2, 3]. However, this vision of a smart grid, in which electricity prices change in real-time to reflect the current balance of supply and demand across the grid, presupposes that the grid's users are also capable of responding in real-time to reduce loads [4]. In the case of domestic users, and particularly for electric heating loads which involve a significant time lag between cause and effect, this is currently not the case. Thus, the Intelligent Decentralised Energy-Aware Systems (iDEaS) (`www.ideasproject.info`) and the Intelligent Agents for Home Energy Management (`www.homeenergyagents.info`) projects at the University of Southampton are developing and demonstrating intelligent agents, to be deployed within these homes, to manage energy use within them. These agents can learn the thermal characteristics of a home, and predict local weather conditions, in order to provide home owners with real-time information about their daily heating costs. Furthermore, these agents can then optimise heating use, taking full account of real-time pricing signals, to minimise cost and carbon emissions whilst satisfying the home owner's preferences for comfort [5, 6].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. HOME HEATING MANAGEMENT

Our home heating management agent learns the thermal characteristics of the home in which it is installed, and the environment in which it operates. In more detail:

- Using internal and external temperature sensors, and by monitoring the activity of the home's heating system, the agent is able to learn the thermal characteristics of the home. More specifically, it models the thermal characteristics of the home through a coupled set of differential equations that describe the flow of heat from the heater, into the internal air, and then out to the structure of the home, and the external environment. A regression process then fits the parameters of this model to the temperature data observed, in order to define (amongst other things) the heat output of the heating system and the thermal leakage rate of the home.

- Using a computationally efficient implementation of multi-output Gaussian processes, the agent then predicts the local external temperature over the next 24 hours by combining local measurements from an external sensor with predictions from an online weather forecast. In doing so, it creates a *site-specific* forecast for the next 24 hours, by explicitly considering both the period nature of its own 24 hour sensor data, and the likely correlation with the online forecast data.

Using these factors the agent is able to predict the consequences, in terms of cost and carbon, of any thermostat setting and provide this information to the home owner through the agent's graphical user interface, informing them of the predicted daily cost and carbon consequences of their current thermostat and time settings. Going further, the agent is then able to fully optimise the use of heating (using either an optimal CPLEX implementation or a computationally efficient greedy heuristic). In doing so, it provides the same level of comfort as a standard thermostat operating at the same set-point temperature (evaluated using a comfort model based on the

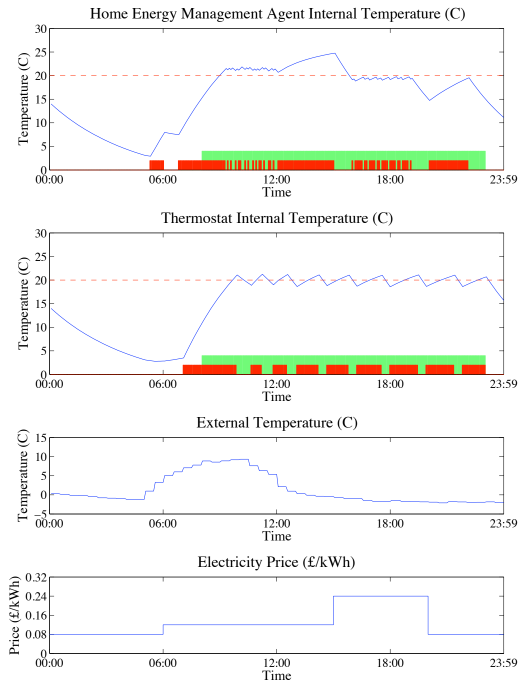


Figure 1: Example setting showing the home heating agent optimising heating use to maintain comfort whilst avoiding a critical pricing period.

ASHRAE thermal comfort standard — ANSI/ASHRAE Standard 55-2010) whilst also minimising either cost or carbon.

Figure 1 shows simulation results for an example setting where the home heating management agent optimises heating to avoid a critical pricing period (i.e. the time period 15:00 to 20:00 where the price of electricity is £0.24 per kWh). In this case, we compare the internal temperature of the home when the heating is controlled by both a standard thermostat and the home heating management agent. In both cases, the green shaded area represents the time interval over which heating is required, and the red shaded area represents when the heating system is actually producing heat. Note that the agent applies heat before the critical pricing period, allowing the temperature to increase, and then allows this heat to leak away over this period (the effect is exaggerated here for clarity). In contrast, the standard thermostat applies heat uniformly across this period. Similarly, note that the agent also exploits the low price of electricity before 06:00 and supplies heat even though it is not immediately required. In both cases, the agent is effectively storing cheap electricity in the form of hot air, so that this stored energy can be used when electricity is more expensive. This provides an alternative to the use of more costly electrical storage batteries, and in this setting, reduces heating costs over the day by 20%.

3. SOFTWARE IMPLEMENTATION

The home heating management agent described here has been implemented within a Java software simulation (see Figures 2 and 3). The simulation represents the physical environment of the home, and is driven by real sensor and weather data for January 2010. A touch sensitive display provides a graphical user interface for the agent that displays the cost and carbon emissions corresponding to any thermostat setting, and allows the both the thermostat setting and mode of operation to be adjusted. A video of the simulator and agent in operation is available online (see `http://www.ideasproject.info/research.php/`).

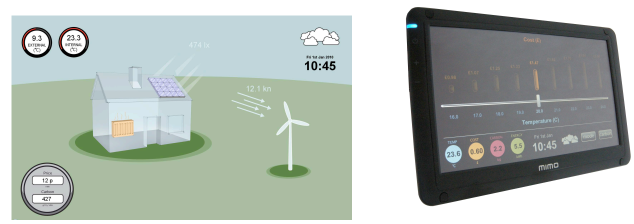


Figure 2: Software implementation of the simulated environment and the graphical user interface to the home heating management agent.

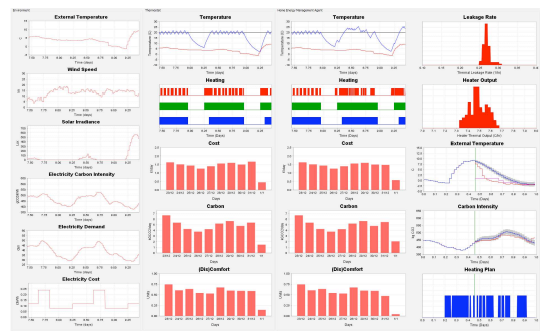


Figure 3: Graphical view of the software agent in operation learning environmental parameters and optimising heating use.

4. CONCLUSIONS

The work demonstrated here shows that a home heating management agent deployed within a home can yield significant cost and carbon savings whilst also facilitating the type of demand response envisaged within the smart grid (i.e. the ability to reduce electricity demand at peak times through real-time pricing). Our future work is focused on developing the currently simulated system into a real-world deployment in conjunction with industrial partners, and to this end, we are currently working to close the control loop and trial the complete controller on a number of instrumented homes owned by the University.

5. REFERENCES

[1] United States of America Department of Energy, *Grid 2030: A National Vision For Electricity's Second 100 Years*, 2003.

[2] HM Government Department of Energy and Climate Change, *Smarter grids: The opportunity*, 2009.

[3] D. MacKay. *Sustainable energy: Without the hot air*. UIT, Cambridge, 2009.

[4] F. Schweppe, B. Daryanian, and R. Tabors. Algorithms for a spot price responding residential load controller. *IEEE Power Engineering Review*, 9(5):49-50, 1989.

[5] A. Rogers, S. Maleki, S. Ghosh and N. R. Jennings. Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming. *Proc. of the 2nd Int. Workshop on Agent Technology for Energy Systems*, Taipei, Taiwan. pp. 71-78, 2011.

[6] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Decentralised control for decentralised demand side management in the smart grid. *Proc. of the 10th Int. Conf. on Autonomous Agents and Multiagent Systems*, Taipei, Taiwan. pages 5-12, 2011

Tactical Operations of Multi-Robot Teams in Urban Warfare
(Demonstration)*

Peter Novák Antonín Komenda Viliam Lisý Branislav Bošanský
Michal Čáp Michal Pěchouček

Agent Technology Center, Dept. of Computer Science and Engineering, FEE
Czech Technical University in Prague, Czech Republic

ABSTRACT
With scaling of multi-robot teams deployed in military operations, there is a need to boost autonomy of individual, as well as team behaviors. We developed a feature-rich simulation testbed for experimental evaluation of multi-agent coordination mechanisms applicable in tactical military operations in urban warfare. In particular, we investigated and implemented four approaches including multi-agent mission planning and plan repair, reactive planning for teamwork, patrolling of mobile targets, and tracking of smart targets. Besides the live-system demonstrator, we aim to showcase a scenario engaging a human in a pursuit-evasion game against the algorithms we implemented.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multi-agent systems*

Keywords
tactical operations, game theory, multi-agent coordination and teamwork, planing and plan repair, simulation toolkits

1. ROBOTS IN TACTICAL OPERATIONS
Maturing of hardware robotic technologies enabled a widespread utilization of robots as a supportive technology in military operations. Their use in urban warfare has proven its merits, be it in bomb disarmament, or information-collection tasks, such as area surveillance or tracking of mobile targets. While the state-of-the-art techniques for robot control based on teleoperation suffice for handling individual robots, they do not scale well to larger-scale operations. Missions like performing several information-collection tasks concurrently in a geographically large urban area are, however, well in reach of the modern hardware technology. In turn, there is a growing need for development of mechanisms for autonomous operation of multi-robot teams in such scenarios.

Here we report on results of the project Tactical AgentScout[1] carried out by our group in the years 2009–2011, in which we studied *coordination mechanisms for multi-agent systems in the context of tactical military operations in urban warfare*. A typical scenario involves a mixed human-robot team acting in an urban area so as to find and arrive to a safe-house where a person of interest is held, secure the person and finally escort it to a given destination where the mission ends. To support an experimental evaluation of four particular multi-agent coordination techniques we studied, we implemented an agent-based simulator. It allows us to test the resulting agent behaviors in more realistic settings and provides a visually attractive and interactive live-system demonstrator.

*Demo video: `http://agents.cz/download/tas2/`
[1]`http://agents.cz/projects/agentscout2/`

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

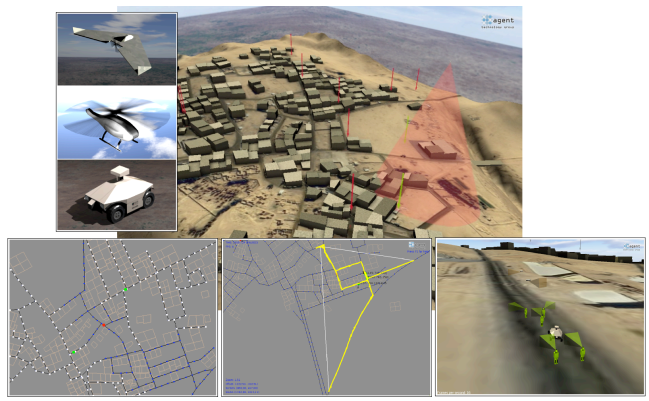


Figure 1: Tactical AgentScout simulator facilitates rich models of physical reality and supports physical mobility dynamics of robots, such as ground and aerial vehicles, be it fixed-wing aircraft, helicopters or various rotorcraft.

2. MULTI-AGENT COORDINATION
We studied four coordination techniques in the context of a rescue scenario in a hostile urban environment. Each approach comes with a test-case demonstrator and an accompanying video material, which are a part of the live-system demonstrator.

Multi-agent planning and plan repair
To carry out the rescue scenario, the team of cooperative robots with heterogeneous capabilities needs to plan the mission including the coordination among the team members, none of which is able to complete it without a help of the others. The hostile urban area represents a highly dynamic environment in which action and in turn also high-level task execution often fails.

We investigated techniques of fully decentralized mission planning and subsequent plan repair in the case of a plan failure [2]. At the heart of our approach lies a multi-agent extension of classical planning based on distributed constraint satisfaction. For the case of a plan failure, we proposed several decentralized plan repair algorithms preserving as much of the original plan as possible. The planning, execution, monitoring and plan repair components are all fully distributed and based on a state-of-the-art multi-agent planning approach MA-STRIPS. Our experiments demonstrate that in a case of an unexpected plan failure, decentralized plan repair leads to a lower communication overhead than a naive replanning.

Reactive planning for teamwork

Predictability of peer behavior is of paramount importance in mixed human-robot teams in military scenarios, such as traversal of cluttered urban environment by an alert squad. Execution of pre-scripted behaviors is more suitable in such contexts in contrast to fully automated mechanisms for activity planning.

We studied extensions of agent-oriented programming techniques to multi-agent teamwork programming and coordination. We employed and adapted the framework of *Distributed Commitment Machines* (DCM) and formulated mission specifications as interconnected networks of individually executable agent commitments. A DCM takes care of both activation of new commitments succeeding the completed ones, as well as correct handling of plan interruptions. Our initial experiments indicate that the DCM framework is a promising tool for concise specification of team-level missions and that reactive agent programs implementing a DCM-based mission specification are capable of generating correct and elaboration tolerant behavior patterns.

Patrolling of mobile targets

Convoys traversing a hostile urban area need to be protected. Firstly, an aerial patrol needs to re-visit the convoys sufficiently often to minimize the window of opportunity for the adversary to attack. Secondly, the patrol's route must be randomized so as to minimize its own predictability assuming the adversaries observe the patrol.

To model the scenario, we used a computational game-theoretic model termed *patrolling games* [1], which we extended with the concept of mobile targets that correspond to the traversing convoys. We use *Strong Stackelberg Equilibrium* (SSE) as the solution concept in the patrolling games. In turn, we formalized computing SSE as finding a solution to a set of non-linear mathematical programs. We performed an experimental evaluation of the model on a scenario with a large number of adversaries aiming to attack the convoys and compared the game-theoretic approach with the deterministic patrolling strategy. Our technique achieved better results even though a number of simplifying assumptions of the mathematical model were not met in the experimental environment.

Tracking of smart targets

Smart targets are those aware that they are being pursued and try to actively avoid the tracker. Tracking of such targets is particularly challenging due to the need to be able to predict the adversary's moves and execute a rational behavior against its best strategies.

Our solution was based on a game-theoretic model of *pursuit-evasion game with imperfect-information* [3]. We model the visibility-based pursuit-evasion game as an imperfect-information extensive form game. The game is too large to compute its exact Nash equilibrium, hence we use the information-set search approach with paranoid opponent model that computes a guaranteed strategy of the players. To meet the real-time computational constraints, we used Monte Carlo tree search to explore the information-set trees. Our experimental results indicate that this combination of techniques allows creating a successful autonomous team of pursuers in practically large domains with real-time computational constraints.

3. DEMONSTRATOR

Our aim is to demonstrate the results of the project embodied in a multi-agent simulator testbed, as well as the underlying technological platform. To this end, we will demonstrate the live Tactical AgentScout system running a series of example scenarios showcasing the multi-agent coordination mechanisms discussed above in an example of a tactical rescue mission in a simulated complex urban environment. The example multi-robot systems are embodied in a simulated physical environment developed in an in-house agent toolkit Alite[2]. Besides facilitating *en masse* experimental data collection, the platform enables rapid modeling of various types of robotic assets (see Figure 1), including features such as physical dynamics of vehicles, physical occlusions, 3D landscape, both discrete, as well as event-based simulations, etc.

Furthermore, the live-system demonstration will be accompanied by an interactive interface allowing a human subject to engage the implemented game-theoretic algorithms in a pursuit-evasion game. The human will be able to control the evader in two basic modes: the full information scenario presenting all the information about the state of the simulation, and an incomplete-information mode providing the evader with only limited sensory data about the world (e.g., unknown positions of the pursuers). The first mode supports debugging and analysis of the algorithms, while the later facilitates evaluation of the methods against a human adversary.

4. DISCUSSION AND FINAL REMARKS

Development and evaluation of multi-agent coordination algorithms targeting their deployment on robotic assets is a challenging task. On one hand, evaluation on target physical platforms is expensive and possibly even risky, or legally problematic (e.g., flight permissions). On the other, testing in simulations is prone to neglect important, but often not obvious constraints of the physical reality. Our approach is innovative in that we experimentally evaluate the techniques in high-fidelity simulations allowing us to model relevant physical features in detail, be it visibility occlusions by buildings, interactions of terrain with physical dynamics of ground vehicles, etc. The natural next step, a subject of an on-going intensive effort in our group, is to adapt and deploy the coordination techniques to real-world robotic hardware, in our case unmanned aircrafts. The underlying multi-agent toolkit Alite provides an open and flexible platform for rapid implementation of the supporting infrastructure in terms of series of experimental simulations and mixed-reality simulations.

Acknowledgements

Supported by the U.S. Army CERDEC grant W911NF-10-1-0112. Authors thank Dušan Pavlíček for production of demonstration videos.

5. REFERENCES

[1] Branislav Bošanský, Viliam Lisý, Michal Jakob, and Michal Pěchouček. Computing time-dependent policies for patrolling games with mobile targets. In *Proceedings of AAMAS 2011*, May 2011.

[2] Antonín Komenda, Peter Novák, and Michal Pěchouček. Decentralized multi-agent plan repair in dynamic environments. In *Proceedings of AAMAS 2012*, 2012.

[3] Viliam Lisý, Branislav Bošanský, and Michal Pěchouček. Anytime algorithms for multi-agent visibility-based pursuit-evasion games. In *Proceedings of AAMAS 2012*.

[2] http://agents.cz/projects#alite

MITRO: an augmented mobile telepresence robot with assisted control (Demonstration)

Sjriek Alers, Daan Bloembergen, Max Bügler, Daniel Hennes and Karl Tuyls

Maastricht University, P.O. Box 616, 6200MD Maastricht, The Netherlands
{sjriek.alers, daan.bloembergen, m.bugler, daniel.hennes, k.tuyls}@maastrichtuniversity.nl

ABSTRACT

We present MITRO: Maastricht Intelligent Telepresence RObot, a custom-built robot system specifically designed for augmented telepresence with assisted control. Telepresence robots can be deployed in a wide range of application domains, and augmented presence with assisted control can greatly improve the experience for the user.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Autonomous vehicles, Operator interfaces

General Terms

Experimentation

Keywords

Telepresence, autonomous navigation, tele-operation

1. INTRODUCTION

Although the idea of a teleoperated robot for remote presence is not new [4], only recently have so called *telepresence robots* become available to the broader public [2, 5, 6]. The idea of a mobile telepresence robot stems from the inherent limitations imposed by traditional videoconferencing systems, in which interaction is restricted to the meeting room only. Such systems do not allow the user to join the - often important - informal part of meetings generally taking place in hallways and coffee corners. A teleoperated robot can provide means for a mobile teleconferencing system, allowing the user to interact more naturally in the remote office environment.

Various authors have already investigated the use of mobile robots for telepresence. In [6] the authors compare two recently launched commercial products, Anybots' QB and VGo Communications' VGo, with respect to user experience in two scenarios: the scheduled meeting, and the informal hallway meeting. One of their findings is that adding some level of autonomy would enhance the user experience, as it would allow to focus more attention to the conversation and interaction, and less to driving. One possible solution, assisted navigation, is investigated in [5]; the authors conclude that assisted navigation decreases the number of collisions with objects in the environment.

To provide assisted control the robot has to be outfitted with a range of sensors that allow to observe the surroundings and steer clear of obstacles. These sensors can also be used to provide additional information to the remote user or allow for additional functionality. *Augmented telepresence with assisted control* goes beyond the idea of a teleoperated robot simply equipped with a screen and camera. Mapping and localization functionality are used to provide the remote user with a floor map indicating the current location; the map can be annotated (e.g. room numbers) and relevant information is overlaid on the live video feed. Furthermore, the system can autonomously return to its charging location after a meeting or wait ready-to-use at a preset location before the meeting commences. People detection and tracking can be used to automatically follow a person to her office; while face tracking allows to follow a conversation without constant steering corrections to keep the conversational partner centered on the screen.

2. IMPACT

Telepresence robots can be deployed in a wide range of application domains, e.g. in workplaces, the public sector or for home use. Telepresence robots are already being used in hospitals to allow doctors and specialists to give consultations from afar [6]. Assisted living facilities outfitted with telepresence systems can provide 24/7 supervision and assistance through remote caregivers. Family members or friends can use the system to pay a virtual visit when time does not allow to be present in person. Telepresence robots can also be used to give people with restricted mobility a new way to outreach and interact beyond their usual living quarters. In all these domains, augmented presence with assisted control can greatly improve the experience for the user.

3. SYSTEM

We present a custom-built robot system (see Figure 1) specifically designed for augmented telepresence with assisted control [1]. MITRO - Maastricht Intelligent Telepresence RObot - is an ongoing research project at the *Swarmlab*[1], the robotics laboratory at the Department of Knowledge Engineering (DKE), Maastricht University.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

[1]For more information visit: `http://maastrichtuniversity.nl/swarmlab`

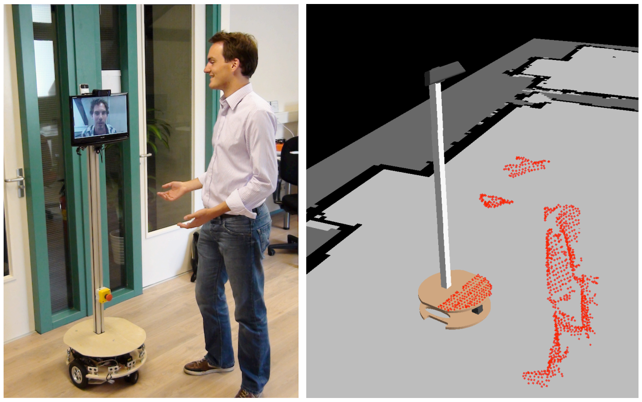


Figure 1: Interacting with MITRO.

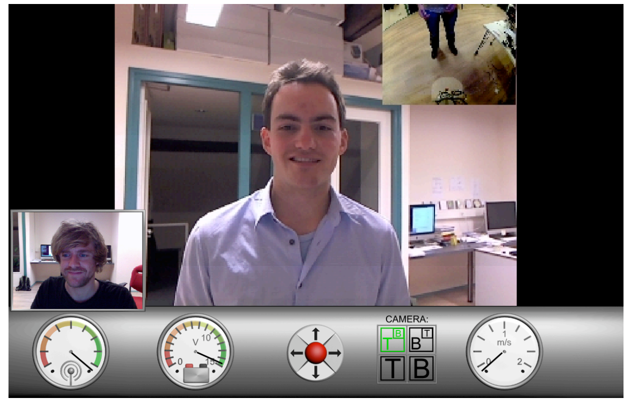


Figure 2: MITRO user interface

3.1 System specifications

The MITRO platform is based on the Parallax Mobile Robot Base kit, which includes the base plate (⌀ 46 cm), powerful motors and 6 inch wheels with pneumatic tires. Additionally, a Hokuyo URG-04LX-UG01 laser range finder is mounted at the front of the base to provide a detailed representation of the environment (240° range; 0.36° resolution; 10 Hz rate, 4m range), used for mapping and localization. A pole is fitted on the base plate and serves as the elevated attachment point for the 14" LCD screen, speakers, two cameras (one pointing forward for conversations, one fish-eye camera pointing downwards for driving), and a Microsoft Kinect sensor. The latter is used for people tracking and obstacle avoidance, and can be used to extract additional features from the environment. The robot has an overall height of 160 cm, the size of a small person, allowing for natural conversation while standing or being seated.

3.2 Software

The MITRO project makes use of and contributes to ROS[2], an open source robot operating system [3]. The modularity and easy extendability of this system makes it an ideal choice for the development of a wide range of robotics applications. ROS fully supports Ubuntu, which makes this the main operating system of choice. In addition to ROS, MITRO makes use of cross-platform video conferencing software and an interface, which enables the user to control the robot, and receive status updates (see Figure 2).

3.3 Capabilities

In order to provide assisted control and augmented telepresence, the robot is able to perform SLAM (simultaneous localization and mapping) to build a map of its environment. This map is used subsequently for localization and autonomous navigation, and can be annotated by the user for convenience. Obstacle avoidance is implemented using a range of sensors, which provides assistance during manual operation or full autonomous navigation if desired. People and face tracking can be used for more natural interactions.

4. DEMONSTRATION

We invite people to engage in a hands-on experience with our MITRO telepresence platform. A laptop computer running the client-side control interface and video-conferencing application allows the user to steer the robot around, take part in conversations and test the assisted control. Additional information (such as an annotated map) is also available. Furthermore, we will demonstrate autonomous drive to a chosen location (e.g. charging station) and people tracking. For more information, visit:
`http://swarmlab.unimaas.nl/papers/aamas-2012-demo/`.

5. REFERENCES

[1] Sjriek Alers, Daan Bloembergen, Daniel Hennes, and Karl Tuyls. Augmented mobile telepresence with assisted control. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC 2011)*, 2011.

[2] Daniel A. Lazewatsky and William D. Smart. An inexpensive robot platform for teleoperation and experimentation. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, 2011.

[3] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Y. Ng. ROS: An open-source Robot Operating System. In *Proceedings of the Open-Source Software workshop at the International Conference on Robotics and Automation (ICRA)*, 2009.

[4] T.B. Sheridan. Telerobotics. *Automatica*, 25(4):487 – 507, 1989.

[5] Leila Takayama, Eitan Marder-Eppstein, Helen Harris, and Jenay M. Beer. Assisted driving of a mobile remote presence system: System design and controlled user evaluation. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, 2011.

[6] Katherine M. Tsui, Munjal Desai, Holly A. Yanco, and Chris Uhlik. Exploring use cases for telepresence robots. In *Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction*, 2011.

[2] For more information on ROS visit: `http://www.ros.org`

Toolkit for Teaching Steering Behaviors for 3D Human-like Virtual Agents (Demonstration)

Markéta Popelová, Cyril Brom, Jakub Tomek and Michal Bída
Charles University in Prague, Faculty of Mathematics and Physics,
Malostranské nám. 2/25, Prague, Czech Republic

ABSTRACT
Steering techniques present an important approach to navigation of 3D human-like characters; however, tools for teaching these techniques to students of courses on computer games, computer graphics or software agents are lacking. Here, we present three freely available tools that can be used for this purpose. The first one is a Java library of steering behaviors for Pogamut toolkit for developing control mechanisms of virtual agents. The second one is SteeringTool, an "off-the-shelf" simulation enabling students to investigate how steering rules work in various situations and under a variety of conditions, and the third one is a serious game SteeringGame for motivating students to study this topic.

Categories and Subject Descriptors
I.6.5 [**Simulation and modeling**]: Model Development

General Terms
Algorithms

Keywords
Steering behaviors, Human-like virtual agents, Interactive agent-based software systems, Virtual agents and interactive virtual environments, Agent-based games, Open-source software tools for agent-based system development, Education.

1. INTRODUCTION
Steering behaviors of C. W. Reynolds [1] are a well known mechanism of navigating virtual agents in a virtual environment. They are simple, predictable and computationally inexpensive. There are several tools for development and demonstration of steering behaviors [2,3] and their comparison and debugging [4,5,6]. However, these tools mostly concentrate on the navigation of boids (e.g., bird flocks or fish schools). Many applications, such as computer games or urban simulations, feature 3D human-like agents, which brings specific requirements on steering behaviors compared to non-human-like agents. For instance, a human observer has usually specific expectations concerning smoothness of the human-like agents' movement, their ability to anticipate and plan actions ahead, but also express social relations to other agents. To our knowledge, a freely available tool supporting education of steering behaviors in the context of 3D human-like agents has been lacking. Such a tool should not only allow a teacher to demonstrate various steering rules to students, but also enable students to gain "hands-on" experience with steering behaviors in a large 3D simulations featuring agents with customizable steering behaviors.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

We have created a freely available Java library of several steering behaviors, based on these of C.W. Reynolds, but tailored to 3D human-like agents. Also, we have created a new steering Walk Along [7], used to steer pairs of people. This steering behavior shows that steering behaviors may not only control low-level navigation, but they may also be used to express social relations between agents. The library is connected to a tool Pogamut [8] for development of control mechanisms of virtual agents, but also to about 1 km^2 large 3D virtual town with four 3D agent avatars we developed; all freely available for educational purposes. The 3D world runs in UnrealEngine2Runtime. On the top of that, we have developed a 3D simulation SteeringTool, in which students can investigate the consequences of the steering behaviors and various settings of their parameters. Finally, we have also developed a serious game SteeringGame that challenges students with several logical tasks on practicing steering techniques. The intended audience is university students and interested high-school students, and their teachers. The toolkit is suitable for steering up to about a dozen of agents, not for crowd simulations.

In this paper, we present the Java library and the two applications. All the mentioned software and the video showing its use is available at http://diana.ms.mff.cuni.cz/pogamut-games/.

2. UT2004STEERING LIBRARY
The library has been written in Java, detailed description of its architecture can be found in [9], a shorter overview is in [7].

In our library, we have implemented the following seven steering behaviors: Target Approaching, Obstacle Avoidance, Path Following, Leader Following, Wall Following, People Avoidance, and Walk Along. The first five are based on [1]. People Avoidance uses a similar approach as in [10]. Leader Following allows for setting the agent's relative position to the leader, which is our innovation to Reynolds' version of this steering behavior. Walk Along is our new steering behavior and is detailed in [7]. All steering behaviors are detailed in [9].

Examples of our improvements to believability of human-like agents are: The agents steered by People Avoidance predict movement of nearby agents, applying further slowing, acceleration or rotation vectors to themselves, thus preventing collisions in more human-like way. Path Following has been made more fluent by adding a steering vector parallel with the current direction of the path. Followers in Leader Following may be steered to a specific location relative to the leader (e.g., 200 cm to the right from the leader), thus any formations may be made. The newly created Walk Along steering has been designed for human-like agents from the beginning.

In general, the exact impact of implemented steering behaviors on believability largely depends on the values of their parameters, which may be set by the user. This makes the library versatile and more interesting from the educational point of view.

3. STEERING TOOL SIMULATION

The SteeringTool simulation has been created with the purpose of easy observation, testing and thereby understanding of steering behaviors. The application contains two main windows: the window displaying our 3D virtual city and the window used to assign steering behaviors to virtual agents. The user chooses the number of agents in the scene, where they start, how they are turned and several other attributes (starting velocity, texture of the agent, etc.). The user also assigns various steering behaviors to these agents (different agents may have different behaviors) and sets the parameters of these steering behaviors (e.g., target location, who is the leader, how powerfully is the agent repelled by other agents).

The application contains a bird-eye-view map of the city, showing, for all agents, their target locations and the path chosen by Path Following steering (if used). The locations may be moved around the map by dragging.

When all parameters of the scene are set, the scene may be played and watched in the 3D virtual environment. The scene may be paused. During a pause, steering behaviors may be reassigned and/or get different parameters. It is also possible to save the scene; it may be loaded and replayed again later.

An important component of the application is Trajectories. The user may load data of previously saved scenes and display the trajectories of agents in the scene. It is displayed how the trajectories change in time, along with forces that affected the agents. This is crucial for understanding how various steering behaviors work and, most importantly, why. Trajectories of several scenes may be displayed simultaneously, thus allowing the user to compare several scenarios at once.

For educational purposes, the application contains predesigned scenes demonstrating specific features of implemented steering behaviors and their combinations. Some of these scenes concentrate on the innovations of steering behaviors that lead to higher believability in human-like agents. In the tool, it is easy to compare how the agents behave with and without the innovations.

4. STEERING GAME

Part of the toolkit is a serious logical minigame SteeringGame. The player solves various missions by assigning proper steering behaviors to agents in the mission, so that they go through a set of predesigned checkpoints. The user has to find a proper combination of steering behaviors to solve the situation and she/he has to set the steering behaviors' parameters properly. The missions are of four difficulty levels, the first being a tutorial. With the growing difficulty of missions, the player understands increasingly more delicate mechanisms of steering behaviors without reading complicated manuals. An editor of new missions is included for teachers.

5. CONCLUSION

We have presented a toolkit facilitating education in the field of 3D human-like virtual agents. The toolkit features three tools: UT2004SteeringLibrary, SteeringTool and SteeringGame. The toolkit should mainly serve to make teaching steering behaviors easy and more fun. The application has been tested by local students and evaluated as easy-to-understand. The application is finished, including a help and a tutorial in English, and it is ready to be used.

We believe that university students may use this toolkit to further their knowledge of virtual agents' navigation and that more high-school students will become attracted to computer science and software agents in particular, via playing SteeringGame.

6. ACKNOWLEDGMENTS

This research was partially supported by project P103/10/1287 (GACR), by student grants GA UK No. 0449/2010/A-INF/MFF and No. 655012/2012/A-INF/MFF and by SVV project number 263 314. We also thank to Jakub Gemrot and Jakub Jirka for their contributions.

7. REFERENCES

[1] Reynolds, C. 1999. Steering behaviors for autonomous characters. In: Game Developers Conference, 763–782.

[2] Reynolds, C. Open Steer: http://opensteer.sourceforge.net/. (8.1.2012)

[3] Schnellhammer, Ch., Feilkas, T. The Steering Behaviors: http://www.steeringbehaviors.de/. (8.1.2012)

[4] Singh, S, et al. 2009. SteerBench: a benchmark suite for evaluating steering behaviors. Computer Animation and Virtual Worlds, vol. 20, issue 5-6, DOI=10.1002/cav.277.

[5] Singh, S, et al. 2009. An Open Framework for Developing, Evaluating, and Sharing Steering Algorithms. In Proc. of the 2nd International Workshop on Motion in Games, LNCS, vol. 5884, Springer, Heidelberg, 158-169.

[6] Kapadia, M., et al. 2009. SteerBug: an interactive framework for specifying and detecting steering behaviors. In Proc. of the 2009 ACM SIGGRAPH/ESCA, ACM Press, New York, NY. 209-216.

[7] Popelova, M., Bida M., Brom, C., Gemrot, J., Tomek, J. 2011. When a Couple Goes Together: Walk Along Steering. In Proc. Motion in Games, 2011, LNCS 7060, Springer, Heidelberg, 278-289.

[8] Gemrot, J., et al. 2009. Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents. In: Dignum, F., Bradshaw, J., Silverman, B., van Doesburg, W. (eds.) Agents for Games and Simulations. LNCS, vol. 5920. Springer, Heidelberg, 1–15.

[9] Popelova, M. 2011. Steering techniques library for virtual agents. Bachelor thesis. Charles University in Prague. (in Czech), http://amis.mff.cuni.cz/emohawk/ (8.1.2012)

[10] Karamouzas, I., et al. 2009. A Predictive Collision Avoidance Model for Pedestrian Simulation. In: Proc. Motion in Games, 2009. LNCS, vol. 5884, Springer, Heidelberg, 41-52.

A Development Environment for Engineering Intelligent Avatars for Semantically-enhanced Simulated Realities (Demonstration)

Stefan Warwas, Matthias Klusch, Klaus Fischer, Philipp Slusallek
German Research Center for Artificial Intelligence
Campus D3_2, 66123 Saarbrücken, Germany
{firstname.surename}@dfki.de

ABSTRACT

As of today, the behavior of avatars in virtual worlds is usually realized by script sequences which provide the illusion of intelligent behavior to the user. In the research project ISReal, our research group developed the first platform for deploying virtual worlds based on Semantic Web technology, which enables agents to reason about and plan with semantically annotated 3D objects. Powerful tool support is required to design agents which exploit the functionality of the ISReal platform. We decided to reuse existing facilities provided by the model-driven Bochica framework for AOSE and extended it with a platform model for agents situated in semantically-enhanced simulated realities.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; D.2.6 [**Programming Environment**]: Graphical environments

General Terms

Design, Languages

Keywords

Agent Oriented Software Engineering, Development Environment, Semantic Virtual Worlds

1. INTRODUCTION

Modeling multiagent systems (MAS) is a complex endeavour. An ideal domain specific agent modeling language would be tailored to a certain application domain (e.g. virtual worlds) as well as to the target execution environment (e.g. a legacy virtual reality platform). At the same time it is desirable to reuse application domain independent model artifacts that already proved their use. In [3], the Bochica framework for engineering MAS was introduced. It is is based on the platform independent core modeling language Dsml4Mas and can be tailored through several extension interfaces to the user's needs.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The underlying idea of the ISReal project was to use Semantic Web technology to enhance purely geometric objects with ontological information (OWL-based) and specify their functionality by semantic service descriptions (OWL-S-based), called *object services* [1]. Object services are grounded in animation and simulation modules. Intelligent avatars are equipped with a sensor component to perceive this information. Developing ISReal avatars involves, beside the core concepts of MAS (e.g. goals, behaviors, and interaction protocols) also ISReal specific aspects such as Semantic Web technology and 3D-related concepts. The remainder of this paper provides an overview of the Bochica framework (Section 2) and the ISReal specific extensions for the development environment (Section 3).

2. THE BOCHICA FRAMEWORK

The Bochica framework evolved from the Pim4Agents approach and is based on Eclipse technology. Here follows an overview of some of the new features:

Expressiveness. Expressive modeling languages are required for closing the gap between models and code. For this purpose, we further developed the underlying core modeling language so that large portions of the source code can be generated.

Conceptual Extensions. The Bochica framework offers various interface concepts that can be extended through external plug-ins. For example, existing concepts can be specialized for certain application domains or execution environments. Moreover, new ways for modeling existing aspects can be contributed (e.g. behaviors or interactions).

Language Extensions. There exists a large number of software languages that are relevant for developing agent-based systems such as knowledge representation languages, query languages, or programming languages. Bochica provides abstract language interfaces such as `BooleanExpression` or `ContextCondition` which can be extended by external language plug-ins. The interfaces check syntactical correctness and the binding of variable symbols in the surrounding scope.

Transformations. The Bochica framework uses modular *base transformations* for generating code for certain target agent execution platforms. As Bochica gets extended, an *extension transformation* extends a base transformation for the new concepts. Currently, we have a base transformation for Jadex which is implemented in QVT.

Reusability. It is desirable to reuse model artifacts that proved their practical use and were validated (e.g. inter-

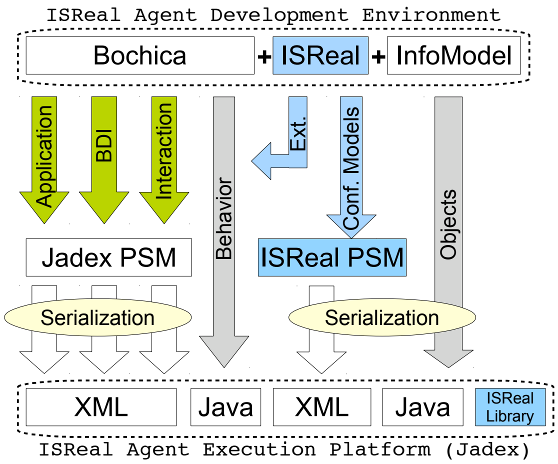


Figure 1: The development environment for ISReal agents consists of a Jadex base transformation (green), behavior and information model transformations (gray), and ISReal specific extensions (blue).

action protocols or goal hierarchies). For this purpose, we established a reverse engineering approach for extracting the underlying structure of Jadex BDI agents [2]. The approach is used to build up model repositories and ease the migration of existing projects to BOCHICA.

3. PLATFORM EXTENSION FOR ISREAL

For the development of intelligent ISReal avatars we decided to reuse the facilities of the BOCHICA framework by providing a ISReal platform extension (see Figure 1). The main features are:

ISReal Concepts. The ISReal platform model contributes ISReal specific concepts such as ISReal sensor configurations and the configuration of the agents' local knowledge bases (e.g. known object services, A- and T-Box).

Service Orchestration. ISReal agents use their sensor component for perception-based service discovery and orchestrate object services of the virtual environment using plans. We extended the modeling environment such that ISReal object services can be orchestrated by plan templates.

Semantic Web. In order to enable intelligent ISReal avatars for SPARQL-based reasoning, we provide a SPARQL language extension for BOCHICA. This extension allows for example to define the target condition of goals and the context condition of plans with SPARQL. We re-used the SPARQL domain specific language provided by EMFText[1].

ISReal Transformation. Based on the existing base transformation from BOCHICA to Jadex we created an extension transformation which provides additional mapping rules for ISReal specific artifacts. For example, it is responsible for generating configurations of the agents' knowledge bases, the SPARQL extension, and the integration into the overall ISReal platform.

ISReal View. Finally, the ISReal plug-in provides a custom ISReal view which allows the creation and configuration of ISReal specific model artifacts such as the agent sensor and the knowledge base configuration.

[1]http://www.emftext.org/

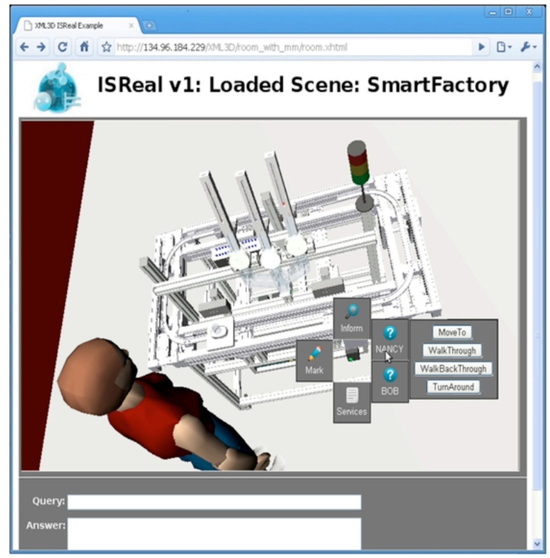


Figure 2: This figure depicts an intelligent avatar operating a virtual machine using object services. The user interface is based on XML3D.

4. CONCLUSION

In this paper we provided an overview of the development environment for intelligent ISReal avatars. The developed cross-disciplinary system integrates agent, Semantic Web, and AI technology, computer graphics, as well as model-driven AOSE. The demonstrator will cover all aspects starting from the modeling phase, throughout code generation, and the execution in the ISReal platform (see Figure 2). The reuse of the infrastructure provided by the BOCHICA framework reduces development and maintenance costs of the tool chain. A set of slides and a video can be found at[2].

5. ACKNOWLEDGEMENTS

We thank Patrick Kapahnke, Cristián Madrigal-Mora, and Stefan Nesbigall who contributed to ISReal and BOCHICA. The work presented in this paper has been partially funded by the German Ministry for Education and Research (BMB+F) under project grant 01IWO8005 (ISReal) and the Saarbrücken Graduate School of Computer Science.

6. REFERENCES

[1] P. Kapahnke et al. ISReal: an open platform for semantic-based 3D simulations in the 3D internet. In *Proc. of the 9th Int. Semantic Web Conference on the Semantic Web*, page 161–176. Springer, 2010.

[2] S. Warwas et al. Making multiagent system designs reusable: A model-driven approach. In *Proc. of the Int. Conf. on Intelligent Agent Technology (IAT'11)*, pages 101–108. IEEE, 2011.

[3] S. Warwas et al. Bochica: A model-driven framework for engineering multiagent systems. In *Proc. of the 4th Int. Conf. on Agents and Artificial Intelligence (ICAART'12)*, pages 109–118. SciTePress, 2012.

[2]http://www.dfki.de/isreal/aamas12demo/slides.pdf

Running Experiments on DipGame Testbed (Demonstration)

Angela Fabregues, Santiago Biec and Carles Sierra
Artificial Intelligence Research Institute (IIIA-CSIC)
Campus Universitat Autònoma de Barcelona, 08193 Bellaterra, Catalonia, Spain
{fabregues, sierra}@iiia.csic.es, santiago@dipgame.org

ABSTRACT

DipGame is a testbed for MAS negotiation involving humans. It is very appropriate to run experiments that mix humans and agents. In this demonstration we introduce an application to facilitate the execution of experiments on several machines and with a friendly graphical user interface.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Experimentation

Keywords

application, negotiation, testbed, diplomacy game.

1. INTRODUCTION

There are recognised difficulties of running experiments on negotiation involving both human agents and software agents [3]. These difficulties are delaying the production of automated negotiation agents. First, most research work on negotiation techniques does not consider humans as counterparts of automated negotiation agents. Second, enticing humans to participate in negotiation experiments is difficult because the negotiation environment is artificial and not attractive, and because the language to use in interactions is unnaturally constrained. Some of the barriers of the latter type are solved by the DipGame testbed [2].

DipGame provides an environment where agents incarnate one of the seven Great European Powers as defined by the Diplomacy Game (`http://en.wikibooks.org/wiki/Diplomacy/Rules`). This game temporally splits in turns where all players move the units that they control over a map of Europe. The goal of the game is to conquer Europe and this is achieved performing cooperative moves with other players that where agreed in the negotiation round that takes place before each turn is executed. Those agreements, usually alliances, can be dishonoured. All the game is about understanding the relationships among agents, knowing to what extend we can ask for help and guessing whether someone is trustful or not [4].

In DipGame we cope with the language problem by providing a formal language and a library that translates human messages about Diplomacy from (a restricted set of) English into the formal language, [2]. This library and a friendly user interface to write the messages are integrated as a web application that everybody can use to play online against other humans and software agents. This interface is available at `http://www.dipgame.org` and helps attracting Diplomacy players to take part in our experiments.

The analysis of the data produced by negotiation experiments, consisting of several game executions, is made with the help of DipTools [1]. This tool allows the experimenter to group the results of sets of game executions in order to compare and analyse them in a intuitive and graphical way.

Several research labs have shown their interest on using the DipGame testbed and have started to design negotiating agents. Building a DipGame negotiation agent became an assignment for some undergraduate and master students who reported the difficulty of testing their agents offline without a simple graphical interface that would allow to set the experimental variables and to collect the resulting experimental data. Offline agent testing is crucial to ensure that software agents perform well and do not crash while playing a game with humans, as this would demotivate them to continue in the experiment. This is the main goal of the experiment manager that we will present in this demo: to facilitate the offline testing of DipGame agents.

Section 2 describes the software that has been developed and Section 3 provides an example of how to use the experiment manager.

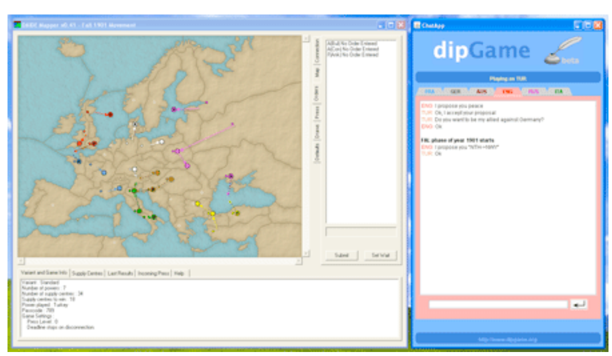


Figure 1: Screenshot of the human graphical interface, ChatApp. The chat is on the right and the map on the left.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2. SOFTWARE DESCRIPTION

The experiment manager, released as the DipGame GameManager, provides a graphical user interface where games can be set up. The application is multi-platform and runs on several devices. It allows to select the players that will take part in an experiment. Each player can be either a provided software agent (currently four such agents are included in the software release), a software agent programmed by the experimenter, or a human interacting through a graphical interface.

The graphical interface for humans is called ChatApp and can be seen in Figure 1. This human interface is included in GameManager and it is also released as an standalone application. ChatApp provides a chat functionality similar to most instant messaging software available in the market. In addition, this chat translates the natural language messages into the formal language that automated agents understand and vice-versa. This translation is done in such a way that players do not know whether the opponent is a human or an agent. Finally, ChatApp renders a map used to select the movements to perform.

The manager allows to set a player to *empty*. This means that the game can be launched even though there are players missing. Once launched, the game will wait for those missing players to connect using the IP address and port indicated by the manager. Missing players can be standalone applications. ChatApp is an example of such standalone application. Software agents can also be executed this way. In section 3 we present an example of experiment concurrently executed over several machines.

3. EXAMPLE

The typical users of the experiment manager are researchers that are developing their own software agents. To run an experiment you have to first download the tool and incorporate your agent into the manager. The software and the instructions for incorporating agents are available at the GameManager section of the DipGame site. Next, you can run the manager and set the experiment selecting the players you like to take part in it. Among the available players you will find those software agents that you incorporated. Finally, run the experiment and save the results into a file.

In [1] we described how to analyse the results of an experiment involving several game executions with negotiating and non-negotiating agents. To run an experiment involving, for instance, four copies of your software agent (each one with possibly different parameter values) and three human agents, you will need at least three machines to interface with the three humans. Three machines would be enough as one might run the experiment manager and the other two the ChatApp standalone application. Thus the experiment manager would have four players set to the software agent, one player set to human and the last two players set to *empty*. When running the experiment, the manager launches the game with two players missing and launches also a ChatApp integrated with the manager. This human interface can be used by one of the humans. For the game to be able to start, the other two humans should connect to the manager by introducing vis their graphical interface the IP of the manager that is shown in the manager main window.

The human interface integrated with the experiment manager should be used only for testing purposes as the human using it would have access to private messages sent between the other players. When the game ends, or when the game is cancelled,[1] the results can be stored in a file. Then, we can take one or several experiment result files, upload them into DipTools and visualise the results as shown in Figure 2 for a particular experiment.

This paper is accompanied with a video demonstration available at `http://www.dipgame.org/media/AAMAS2012demo`.

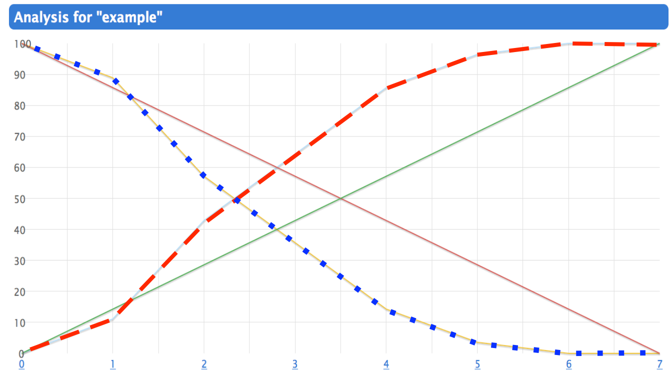


Figure 2: Example of chart extracted from [1]. Percentage of games won per number of negotiating agents. The dashed line represents the percentage of victories of negotiating agents and the doted line the percentage of victories of non negotiating agents. The continuous lines (increasing and decreasing) represent the expected percentage of negotiating and non-negotiating agents in case they all were equal. This particular graphic shows that negotiating agents perform better in the experiment.

4. ACKNOWLEDGMENTS

Research supported by the Agreement Technologies CONSOLIDER project under contract CSD2007-0022 and INGENIO 2010, by the Agreement Technologies COST Action, IC0801, and by the Generalitat de Catalunya under the grant 2009-SGR-1434. This work is supported under project CBIT, which is funded by Spain's Ministry of Science and Innovation under grant number TIN2010-16306.

5. REFERENCES

[1] A. Fabregues, D. López-Paz, and C. Sierra. Diptools: Experimental data visualization tool for the dipgame testbed (demonstration). pages 1315–1316, Taipei, Taiwan, 02/05/2011 2011.

[2] A. Fabregues and C. Sierra. Dipgame: a challenging negotiation testbed. *Journal of Engineering Applications of Artificial Intelligence*, 24:1137–1146, 10/2011 2011.

[3] R. Lin and S. Kraus. From research to practice: Automated negotiations with people. http://u.cs.biu.ac.il/ linraz/Papers/linetal-practice.pdf.

[4] R. Sharp. *The Game of Diplomacy*. 1978. http://www.diplom.org/ diparch/god.htm.

[1]The experiment execution can be canceled at any time from the experiment manager window showing the real time results.

v-mWater: a 3D Virtual Market for Water Rights[1] (Demonstration)

P. Almajano, T. Trescak, M. Esteva
IIIA, CSIC
Barcelona, Spain
{palmajano,ttrescak,marc}@iiia.csic.es

I. Rodriguez, M. Lopez-Sanchez
University of Barcelona
Barcelona, Spain
{inma,maite}@maia.ub.es

ABSTRACT

Most of MAS methodologies and infrastructures do not consider direct human participation even though humans can be seen as autonomous entities (i.e. human agents). Virtual Worlds (VW) provide all the necessary means for direct human inclusion into software systems. Virtual Institutions (VI) take advantage of this and combine Electronic Institutions (EI) and VWs to engineer MAS applications where humans participate together with software agents. In this demo, we introduce virtual mWater (*v-mWater*), a VI for water rights negotiation.

1. INTRODUCTION

Virtual institutions (VI) [1] offer interesting possibilities to both 3D virtual environments and MAS. First, thanks to the regulation imposed by an organization centred MAS –in our case an Electronic Institution (EI) [2]–, the 3D environment becomes a normative Virtual World (VW) where norms are enforced at runtime. Second, this 3D real-time system representation allows a human be aware of its system state and to directly participate in MAS by controlling her/his avatar in an immersive environment.

This demo illustrates a *v*irtual *m*arket based on trading *Water* (*v-mWater*) modelled as a VI with the aim of coordinating participants' interactions and supporting direct human participation in MAS. VIs provide a seamless interaction between both human and software agent participants. We present i) the specification of the system, ii) the VW generation from this specification, iii) the deployment using the Virtual Institution Execution Environment (VIXEE) and iv) how participants interact in the virtual environment.

v-mWater has been deployed using VIXEE, a robust Virtual Institution eXEcution Environment that provides interesting features, such as multi-verse communication and dynamic manipulation of the VW content. VIXEE is a generic and domain-independent solution. Its performance has been evaluated in high load scenarios (more than 500 agents).

[1]This work is partially funded by EVE (TIN2009-14702-C02-01 / TIN2009-14702-C02-02), AT (CONSOLIDER CSD2007-0022) and TIN2011-24220 Spanish research projects, EU-FEDER funds and the Catalan Association for Artificial Intelligence (ACIA).

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

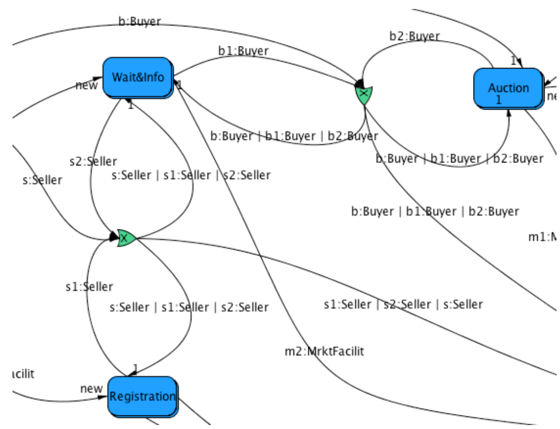


Figure 1: Extract of v-mWater model specification

VIXEE does not introduce any limitations on the scalability of the system as it maintains fast response times even in these scenarios [6]. Although *v-mWater* is an *e-Government* application, VI can also be used in other domains which may benefit from structured interactions and norm enforcement such as *e-Learning* and *e-Commerce*.

2. v-mWATER MODEL

The *v*irtual *m*arket based on trading *Water* (*v-mWater*) is a VI which models an electronic market of water rights. The market is a simplification of *mWater* [3] which is an Electronic Institution (EI) focusing on a water market and including conflict resolution features.

In *v-mWater* scenario, agents may adopt a number of roles. Irrigator agents can participate as either buyer or seller subroles while market facilitator and basin authority correspond to staff agents. Figure 1 shows an extract of the performative structure [2] (i. e. the work-flow among several agent activities called scenes in EI) of *v-mWater*. Besides the obligated initial and final activities to enter and exit the institution, it has three activities which enact the market: *Registration*, where the market facilitator is in charge of registering sellers' rights; *Waiting and Information*, where irrigators can ask for auctions' information to the market facilitator; and *Auction*, where the negotiation of water rights takes place. The auction protocol is multi-unit Japanese. We selected this protocol because it is suitable for perishable and divisible goods (in our case, water). Water rights are auctioned in consecutive rounds. There are three roles involved in this activity: buyers bid for water rights, the

Figure 2: Initial aerial view of v-mWater

market facilitator conducts the auction and the basin authority announces the resulting valid agreements.

3. v-mWATER DEPLOYMENT

In order to engineer *v-mWater*, we follow three steps. First, we specify the normative control layer of the VI – that is an EI– using ISLANDER tool [2]. The output is the EI specification introduced in section 2. Second, we automatically generate the 3D representation from this specification. Third, we define the mapping between VW actions and EI messages and vice versa. We have deployed *v-mWater* model using the *Virtual Institution eXEcution Environment* (VIXEE). Its architecture is composed of three layers: i) normative, ii) visual interaction and iii) causal connection.

The **normative** layer is composed by AMELI, the EI infrastructure that mediates agents' interactions while enforcing institutional rules [2]. AMELI can be regarded as domain-independent because it can interpret any institution specification generated by ISLANDER tool [2]. In our case, it interprets the specification defined in section 2. Software agents are directly connected to this layer. The **visual interaction** layer comprises several virtual worlds (VWs). Each VW can be implemented in a different programming language using a different graphics technology. VW clients provide the interface to human participants whereas servers communicate with the causal connection layer. The **causal connection** layer causally connects the visual interaction and the normative layers, i.e. whenever one of them changes, the other one changes in order to maintain a consistent state [4]. This layer implements a *multi-verse communication* mechanism that allows users from different VWs to participate in the same VI. The mapping between VW actions and AMELI protocol messages –and vice versa– is defined by a *movie script* mechanism. E. g., the welcome event to the institution has been mapped to a *"greeting"* gesture made by the *Institution Manager* avatar (see Figure 3). Moreover, VIXEE uses the Virtual World Grammar (VWG) concept and its implementation in the Virtual World Builder Toolkit (VWBT) to dynamically manipulate the 3D representation of all connected virtual worlds [5].

As a result of this engineering process, Figure 2 depicts the consequent generation[2] in *Open Simulator*, a multi-platform multi-user 3D VW server. In particular, it shows an aerial view of three rooms located at an open space that correspond to the three main activities in *v-mWater*. Software agents have been characterized as bots with the aim of enhancing their artificial nature: they are bold and have differentiated

[2]See http://youtu.be/hJzw40lQvUY for a complete visualisation

Figure 3: Human avatar login: interaction with a software agent by means of a chat window

Figure 4: Bot bidding in a running auction

artificial skin colours that represent their roles (see Figures 3 and 4). Fig. 3 shows the login to the institution: the human participant sends a private message to the *Institution Manager* bot with the password and role (either seller or buyer). Fig. 4 illustrates how human participation in the auction has been improved by providing a comprehensive 3D environment. There, the market facilitator bot appears sited at a desktop and buyer participants at the chairs in front of it. This room includes dynamic information panels. Moreover, bots' bid actions can be also easily identified by human participants since they are displayed as raising hands.

4. CONCLUSIONS

This paper presents *v-mWater*, a Virtual Institution for the negotiation of water rights. We have proposed an immersive environment where human avatars interact with the environment and other participants in the system. As result, our system has favoured direct human participation in MAS. As ongoing work we are extending *v-mWater* with assistance services to participants in order to improve their participation in the system. Moreover, we plan to evaluate the usability of the prototype by measuring interface effectiveness, efficiency and user experience.

5. REFERENCES

[1] A. Bogdanovych. *Virtual Institutions*. PhD thesis, University of Technology, Sydney, Australia, 2007.

[2] M. Esteva. *Electronic institutions. from specification to development*. PhD thesis, UPC, 2003.

[3] A. Giret, A. Garrido, J. A. Gimeno, V. Botti, and P. Noriega. A mas decision support tool for water-right markets. In *AAMAS '11*, pages 1305–1306, 2011.

[4] P. Maes and D. Nardi, editors. *Meta-Level Architectures and Reflection*. Elsevier Science Inc., NY, USA, 1988.

[5] T. Trescak, M. Esteva, and I. Rodriguez. A virtual world grammar for automatic generation of virtual worlds. *The Visual Computer*, 26:521–531, 2010.

[6] T. Trescak, M. Esteva, I. Rodriguez, M. Lopez-Sanchez, and P. Almajano. Execution infrastructure for normative virtual environments. *Engineering Applications of Artificial Intelligence*, submitted.

Context-Aware MAS to Support Elderly People (Demonstration)

Boštjan Kaluža, Mitja Luštrek, Erik Dovgan and Matjaž Gams
Department of Intelligent Systems, Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
{bostjan.kaluza, mitja.lustrek, erik.dovgan, matjaz.gams}@ijs.si

ABSTRACT

This paper presents a context-aware, multiagent system for care of the elderly. The system combines state-of-the-art sensor technologies to detect falls and other health problems, and calls for help in the case of an emergency or issues a warning in cases not needing urgent attention. When deployed at the home of an elderly person it provides them with 24-hour monitoring. Consequently, the elderly may live alone at home, even at an advanced age. The health problems are detected with six groups of agents processing the sensor data and augmenting the data with higher-level information, such as the posture of the person, his/her activity and the context of the situation's environment. The system has been tested in several live demonstrations, where it achieved an excellent performance in complex situations. The system is based on the set of agents observing the elderly person from various points of view, and combining the location and inertial sensors to provide context awareness.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems - medicine and science

General Terms

Algorithms, Measurement, Design, Verification

Keywords

Elderly health care, fall detection, general disability detection, ambient intelligence, ambient assisted living

1. INTRODUCTION

The number of elderly people is increasing rapidly in developed societies. Many of them require assistance with everyday activities. Institutional healthcare already enables monitoring of the elderly and provides help when needed in special facilities or at home. However, the resources for healthcare are insufficient and the presence of care personnel is therefore quite limited. Ambient-assisted-living systems that monitor elderly people at home may be able to effectively cope with this problem.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

Our main objective is the development and integration of innovative technologies to build a care system for the detection of health problems on three time scales: (i) the short-term detection of critical events, such as falls; (ii) the mid-term detection of unexpected behaviors that may be related to health problems, e.g., limping; and (iii) the long-term detection of deviations in behavior that may indicate a disease or a deterioration in the person's health.

There are many commercial and research solutions for fall detection, mainly based on inertial sensors. They typically report nearly 100% performance in laboratory settings. However, when deployed in real-life situations, they often face high false-alarm rates and generally decreased performance. This paper presents a multiagent system[1] to help the elderly. Its novelty is in exploiting the context in problem detection, and in combining inertial and location sensors. The agents are arranged hierarchically, providing increasingly more abstract situational awareness. The system is also able to adapt to each specific user as well as to learn false alarms. The results of the fall-detection experiments show that context-dependent reasoning can detect complex scenarios [3] that might be misinterpreted by inertial-based systems. The exact use of the context is the subject of an ongoing patent application. This paper and demo desribe the functional performance of the Confidence system in complex scenarios.

2. THE CONFIDENCE SYSTEM

The system is designed as a classical hierarchical multiagent system where agents are implemented as task-dedicated heterogeneous procedures with agent properties, i.e., they trigger at a specific pattern and provide one of many opinions or actions. The agents are organized into groups at a specific level of abstraction and coordinated by another, hierarchically higher-level agent. Each agent can be easily modified or replaced and new or redundant agents can be incorporated. The MAS architecture [2] is illustrated in Figure 1. This figure shows the main groups of agents and their interactions indicated by arrows. The agents share the data through direct acquisition using three types of messages. The first type is a measurement message that is created when a new measurement is obtained by the sensor agents. At the initial stage the message contains only raw sensor data. The message is later augmented by other agents with additional interpretation data, e.g., filtered/derived data, posture information, etc. The second message type is an action mes-

[1]The system is the result of the Confidence project, `http://www.confidence-eu.org`

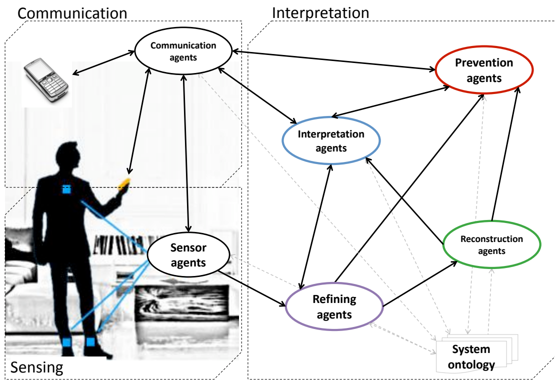


Figure 1: MAS architecture.

sage that is generated when an agent or an agent group requires a service from another agent or group. A typical scenario is when a group of agents detect a fall and require an alarm action from the communication agent group. The last message type is a status message that is used to pool or post the agent state.

At the lowest layer, an arbitrary inertial and location hardware system can be employed. In our case it was the combination of Ubisense[2] and XSens[3]. Sensing agents read the raw sensor data (every 1/10th of a second), and serve them in the form of measurement messages. The sensor agents can also report other information, for example, a status message "low battery" is sent to the communication agent group, which forwards the message to a user-friendly interface. The refining agents filter the noise, compute the derived attributes and map the raw data to a human-body model.

The reconstruction agents aim to determine the posture of the person in the environment. The group consists of classification agents based on machine learning (Random Forest) and expert rules [4]. When all the classification agents provide their label for the posture, a meta-classification agent (Hidden Markov Models) merges the labels to the final posture classification.

The interpretation group of agents detects whether a person is in a dangerous situation. Consider the following situation as an example: an agent detects that the user is *not moving*, the reconstruction agents indicate that the posture of the user is lying, and the refining agents give the location as the *kitchen*. The interpretation agents, implemented with data-driven (Support Vector Machine) and knowledge-driven approaches (expert rules), classify this situation as risky, since it is very unusual and most likely related to a health problem, e.g., the person might have lost consciousness, and inform the communication-agent group.

The prevention-agent group monitors how the person moves on various time scales. It consists of several agents that observe a variety of statistics, e.g., gait characteristics, activity characteristics, daily dynamics, etc. [1, 5]. Each agent pulls the relevant measurement messages from other agents and builds its own behavior model (implemented with outlier detection). When an agent detects a deviation, it notifies the group coordination agent, which decides whether to notify the communication agents.

The last group consists of communication agents that are dedicated to user interaction, for example, the agents that alert the user with a reply demand, make a phone call to relatives or help center, graphically display the state of the system, etc.

The performance was evaluated on a scenario recored by 10 helathy voluteers (five times by each), which included nine different complex fall-detection situations (e.g., fainting, tripping followed by standing up quiclky, quickly lying down on the bed, etc.). The average fall-detection accuracy is 94.7% when using four sensor boxes (neck, belt, both ankels), and 90.1% with one sensor box (neck) only, while the best inertial-based fall-detector was able to achieve the accuracy of 81.8% [3]. To the best of our knowledge, the proposed solution is the only one that: (i) integrates behavior monitoring on several time scales; (ii) incorporates various types of context; and (iii) achieves significantly better performance than inertial-based solutions for fall detection in complex real-life scenarios.

3. DEMO

This demo[4] shows the usability of the system in the following scenarios: (i) complex fall-detection scenarios in which falls can be correctly recognized using the context and (ii) scenarios demonstrating the detection of unusual behavior on two time scales. The fall-detection scenarios include three cases. In the first, a person is lying on the floor and moving. The sequence before the lying posture is crucial to understand the context of the situation and decide whether the situation resulted from a fall or some other activity. The second case shows an example in which the person misses the bed while lying down, which triggers an alarm since the person is not lying where he/she should be. The third case shows the person sitting on a chair and leaning to one side, e.g., due to a heart attack. In this case, the sitting in an unusual posture on the chair raises an alarm, while this posture would not raise an alarm in the bed. Unusual behavior detection is presented with scenarios showing (i) limping detection as a change in the person's gait and (ii) unusual daily dynamics such as frequent toilet visits (or other long-term statistics).

4. REFERENCES

[1] E. Dovgan, M. Luštrek, B. Pogorelc, A. Gradišek, H. Bruger, and M. Gams. Intelligent elderly-care prototype for fall and disease detection. *Slovenian Medical Journal*, 80(11), 2011.

[2] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams. An agent-based approach to care in independent living. In *AmI'10*, pages 177–186, 2010.

[3] M. Luštrek, H. Gjoreski, S. Kozina, B. Cvetković, V. Mirčevska, and M. Gams. Detecting falls with location sensors and accelerometers. In *AAAI 2011*, pages 1662–1667, Aug. 2011.

[4] M. Luštrek and B. Kaluža. Fall detection and activity recognition with machine learning. *Informatica*, 33(2):197–204, 2009.

[5] M. Luštrek, B. Kaluža, E. Dovgan, B. Pogorelc, and M. Gams. Behavior analysis based on coordinates of body tags. In *AmI'09*, pages 14–23, 2009.

[2]Ubisense location system, `http://www.ubisense.net`

[3]XSens inertial motion tracker, `http://www.xsens.com`

[4]Demo video: `http://dis.ijs.si/confidence/aamas2012.html`

Agent Based Monitoring of Gestational Diabetes Mellitus (Demonstration)

René Schumann, Stefano Bromuri, Johannes Krampf and Michael I. Schumacher
Applied Intelligent Systems Laboratory HES-SO
Rue de Technopole 3, 3960 Sierre, Switzerland
rene.schumann|stefano.bromuri|johannes.krampf|michael.schumacher@hevs.ch

ABSTRACT

Gestational diabetes is a type of diabetes affecting temporarily some otherwise healthy pregnant women. Current medical practices do not allow the doctors to monitor such patients as closely as needed. Pervasive Health is a discipline requiring distributed ICT infrastructures to help bridging the gap between the patients and the doctors. In this demo paper we present a complete information system for patient monitoring, including mobile devices for acquiring data from patients and a Web interface for doctors to check the status of their patients. At the core of this information system a multi-agent system monitors the patient health state and triggers alerts to the doctor to raise attention on the specific conditions of a patient. This allows the doctor to react faster to changes of condition of the woman, benefiting the baby's and the mother's health.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Science; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Management, Measurement,Experimentation

Keywords

Intelligent Agents, Pervasive Health, Gestational Diabetes

1. INTRODUCTION

Gestational diabetes mellitus (GDM) [5] affects 2%-5% of all pregnancies and manifests itself with high blood sugar levels. A GDM patient has a higher risk to develop preeclampsia, which can lead to eclampsia, a condition causing the woman to have epileptic seizures and coma, or her baby may develop macrosomia, a condition for which the baby grows too much due to the extra glucose absorbed. Current treatment consists in diet adjustment and introduction of anti-diabetic drugs such as insulin and metformin. The treatment starts by requesting the patient to self-monitor and note down their blood glucose 4 times per day and their blood pressure twice per day. The notebook with the physiological values is then handed to the medical doctors and nurses once or twice per week. The caretakers then propose treatment adjustments according to the patient's status. The problem is that in the two or three days in-between routine checks the patient may develop preeclampsia, eclampsia, or her baby may develop macrosomia [1]. Consequently, a closer monitoring would allow the caretaker to react quickly, preventing damage to the baby's and the mother's health.

For this purpose, Pervasive Health [6] is an approach that aims at bringing healthcare to everyone everywhere, by breaking the boundaries of hospital care, allowing the patients to be monitored during their day-to-day activities. Current Pervasive Health Systems (PHS) have the problem to deal with large amount of data produced by the patients, consequently medical doctors are often loaded with a lot of information on which it is difficult to react promptly, limiting the PHS effectiveness. We have developed the Gestational Diabetes Mellitus Management System (GDMMS), a PHS to handle patient monitoring and react fast on the development of dangerous conditions. We have integrated a multi-agent system (MAS) where we engineered the medical knowledge to reason about the patients' conditions. Each patient has a monitoring agent. If an agent detects a possible harmful condition it creates an alert to notify the doctor in charge. In [2] we described parts of the MAS to handle the monitoring of GDM affected patients by using Event Calculus [4] based agents with abductive logic capabilities. Currently, we are preparing a field test of the entire system at the Lausanne university hospital where patients enter their values using a mobile application, and doctors have a Web interface to interact with the system. We present the usage of the entire system, including smart phones and server infrastructure to monitor GDM patients, by showing how the interaction between its components happens. In particular we will show how our agents produce alerts for medical doctors and nurses, given an anomalous temporal pattern in the patients' physiological data.

2. THE GESTATIONAL DIABETES MELLITUS MANAGEMENT SYSTEM

The GDMMS is at a first glance a conventional information system. Data is entered by users, here most typically via a specific application that runs on Android smart phones, a server back end to store the data, and a web interface to display this information in different ways to the responsible caretakers. The overall architecture is shown in Figure 1.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

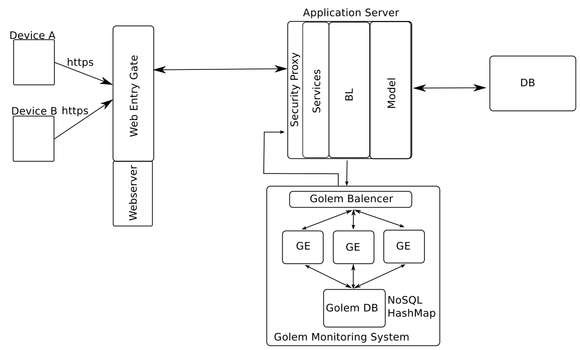


Figure 1: Overall architecture of the GDMMS

The Mobile Infrastructure.

Even though measuring devices for blood-pressure or blood-sugar exist that can transmit data by themselves, we developed an independent Android application. We did so because these newer types of sensors are not accredited for usage in the hospital we collaborate with. Therefore the woman enters her physiological values into a specific applications we have developed. Note that our server-side system offers a REST interface, and therefore is not limited to the usage with this specific application.

The Web Interface.

Our server side system is based on REST interfaces and *Java Persistence API* to handle the data produced within the interaction with the patients and the doctors. Our Web interface is built around *Google Web Toolkit* technology. By means of such interfaces, the doctors have access to the patients' health status, the patients' alerts, and the data associated to the patients. Patients' alerts are also associated to her physiological values and, by clicking on the alert, the doctor can get a summary of the patient's physiological values as associated to the alert of interest. Finally, the Web interface shows to the caretakers the patient's historical values (ethnicity, previous illnesses, age, allergies) and the medicines taken and the contact data of the patient, in the case an alert requires the prompt intervention of the caretaker, such as in the case of preeclampsia.

The Persistent Personal Agent System.

As mentioned before we have added a MAS to the server side that can analyze the patients data and monitor their condition. This MAS is based on the GOLEM platform [3]. The agents are executed within distributed containers that deal with the agents' life-cycles. Our MAS is regulated by a load balancer which splits the traffic generated by the patients in distributed GOLEM containers. The agents are treated as persistent resources associated to a patient, every time a patient logs into the system the agent state is resumed from an agent database. Similarly, when the patient logs off or it's session expires, the GOLEM container serializes and saves the agents in the agent database for future use. This allows us to handle a large number of patients, since the number of active agents is reduced. In particular our agent mind is based on deductive rules using the Event Calculus, to produce treatment adjustment alerts, and on abductive logic rules, to produce macrosomia and preeclampsia alerts.

3. THE DEMONSTRATION

We present the complete system in live operation. We show the components, their interaction, and a live demo. Therefore we present three concrete scenarios, with synthetic patient data. In the first scenario we consider a patient that is experiencing a poor glycemic control, requiring a set of actions to be implemented by the medical doctors to adjust her treatment by introducing further glucose checks or more insulin during the day. In the second scenario we demonstrate the abductive logic capabilities of our agents by showing how preeclampsia alerts are produced when the patient presents a set of symptoms related to preeclampsia or when the patient presents a high blood pressure related to a protenuria confirmed by the medical doctors. In the third scenario, we consider a patient that is towards the end of the pregnancy, has poor glycemic control and is gaining too much weight. Under this condition, our agents produce an alert of macrosomia as the weight gain may be an indicator of the fact that the baby is growing too much.

4. CONCLUSIONS

The GDMMS system will go live in a field test that is performed with the university hospital Lausanne. In a first phase we will collect feedback form patients and doctors to adopt the systems to their needs. Afterwards, we will enter a second field-test with a larger number of patients to measure the effects of the usage of PHSs from the medical perspective. So we see that the GDMMS system will have a strong foundation and will hopefully improve the care for pregnant women and their babies. Furthermore ideas and components, among them the MAS, will be applied to other illnesses in a EU project that has just started.

Acknowledgment

We want to thank Dr. Juan Ruiz from CHUV for his valuable medical advices. This work has been partially funded by the Hasler Foundation, the Nano-Tera Consortium and the COMMODITY12 project (FP7 #287841).

5. REFERENCES

[1] M. E. Boyd, R. H. Usher, and F. H. McLean. Fetal macrosomia: prediction, risks, proposed management. *Obstet Gynecol*, 61:715–722, Jun 1983.

[2] S. Bromuri, M. Schumacher, K. Stathis, and J. Ruiz. Monitoring gestational diabetes mellitus with cognitive agents and agent environments. In *Proceedings of the 2011th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2011)*, Aug 2011.

[3] S. Bromuri and K. Stathis. Distributed Agent Environments in the Ambient Event Calculus. In *DEBS '09: Proceedings of the third international conference on Distributed event-based systems*, New York, NY, USA, 2009. ACM.

[4] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95, 1986.

[5] D. C. Serlin and R. W. Lash. Diagnosis and management of gestational diabetes mellitus. *Am Fam Physician*, 80:57–62, Jul 2009.

[6] U. Varshney. *Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring*. Springer Publishing Company, Incorporated, 2009.

Protos: A Cross-Organizational Business Modeling Tool (Demonstration)

Anup K. Kalia, Pankaj R. Telang, Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{akkalia,prtelang,singh}@ncsu.edu

ABSTRACT

Traditional approaches to cross-organizational business modeling use low-level abstractions such as data and control flow. These approaches result in rigid models that over-constrain business execution. Further, because such approaches ignore the underlying business relationships that drive process execution, they lack the notion of business level correctness.

Telang and Singh [5] propose a high-level business modeling approach based upon (social) commitments to address these shortcomings. The high-level model captures the business relationships in terms of commitments between the participants. Telang and Singh [5] develop a method for verifying if a low-level interaction model satisfies a high-level business model. They propose a top-down methodology in which a Business analyst first develops a high-level business model. An IT analyst then develops UML 2.0 sequence diagrams, and verifies if they satisfy the high-level model.

Protos is an Eclipse-based tool that implements Telang and Singh's [5] methodology. It enables: (a) the development of a high-level business model using reusable patterns, (b) the development of UML 2.0 sequence diagrams, as a low-level operational representation, and (c) the automated verification of the UML 2.0 sequence diagrams with respect to the high-level business model.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

Keywords

Software engineering, Commitment, Verification

1. ARCHITECTURE

1.1 Conceptual Architecture

Figure 1 shows the conceptual architecture of Protos. A business analyst starts with a desired cross-organizational scenario. The analyst selects an appropriate set of patterns from a pattern library, and composes them to develop a business model. Computation tree logic (CTL) specifications formalize each of the business pattern in the library. The union of the CTL specifications for all the patterns occurring in a business model constitute a formalization of the model.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

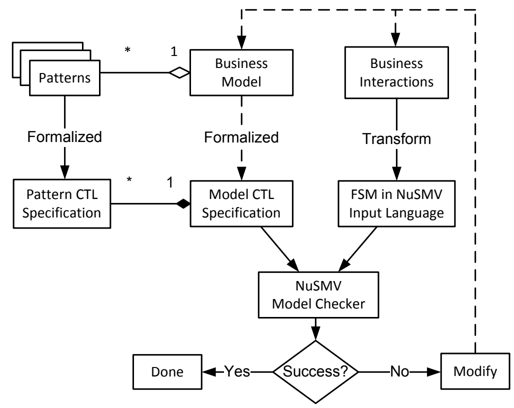


Figure 1: Conceptual architecture (verbatim from [5]).

An IT analyst starts from a business model to develop UML sequence diagrams (operational model). The analyst employs the standard operators of UML 2.0, such as alt(ernate) and opt(ion) with appropriate guards to specify the sequence diagrams. The sequence diagrams are transformed into a finite state machine in the NuSMV input language. The IT analyst runs the NuSMV model checker to verify if the sequence diagrams satisfy the business model. If the model-checker returns success, then the sequence diagrams satisfy the business model. Otherwise, IT or Business analyst inspect the NuSMV counterexample to identify the cause of the failure. They make appropriate changes to either or both of the models and rerun NuSMV.

1.2 Tool Architecture

Figure 2 shows the architecture of the Protos tool. The Protos tool consists of five key components: business modeler, UML sequence diagram modeler, Protos engine, Protos parser, and NuSMV model-checker.

The Business Modeler is implemented as an Eclipse plugin using the Eclipse Modeling Framework (EMF) [1] and the Graphical Modeling Framework (GMF) [2]. The concepts and constraints of the business metamodel are specified in an EMF ECore model. The graphical aspects of the tool such as the concept icons, connectors, and menus are specified in the GMF model. An Eclipse plugin of the tool is then generated using the GMF framework. The business modeler enables saving a business model as an ECore model instance file.

Figure 3 shows a screenshot of the business modeler presenting a model of a real-life Quote-to-Cash business process.

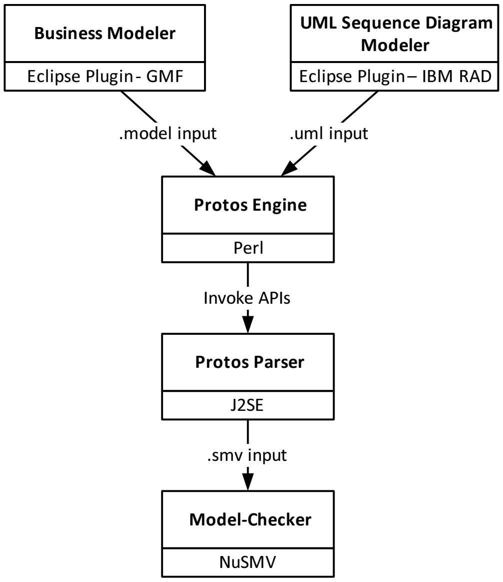


Figure 2: Protos tool architecture.

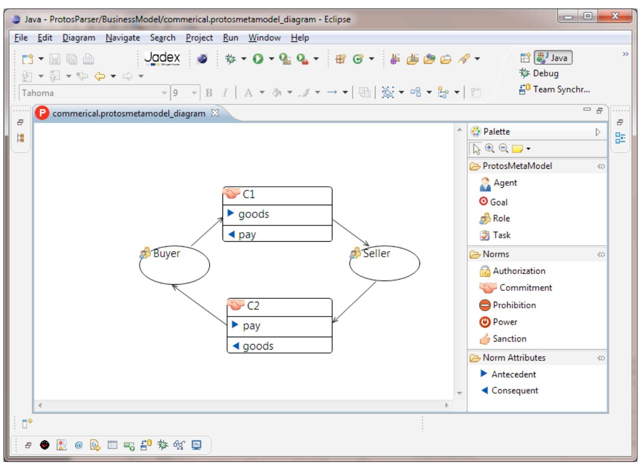


Figure 3: Protos business modeler.

The UML Sequence Diagram Modeler is part of IBM's Rational Software Architect (RSA) version 8.0 [3]. IBM RSA is an Eclipse-based tool that supports developing UML 2.0 sequence diagrams. The UML sequence diagram modeler can output a sequence diagram in a standard format.

Figure 4 shows a screenshot of the UML sequence diagram modeler from IBM RSA, presenting a sequence diagram with buyer and seller lifelines. We capture the meaning of a message as an annotation: for example, the accept message means the creation of the commitment C1 = C(Buyer, Seller, goods, pay).

NuSMV [4] is a well-known tool for model-checking. It verifies if a model specified as a finite state machine satisfies a given set of temporal logic formulae. It support computational tree logic (CTL), which we use. Protos employs CTL formulas to formalize a business model. If the finite state machine fails to satisfy a temporal logic formula, NuSMV returns a counterexample which indicates the cause of the failure.

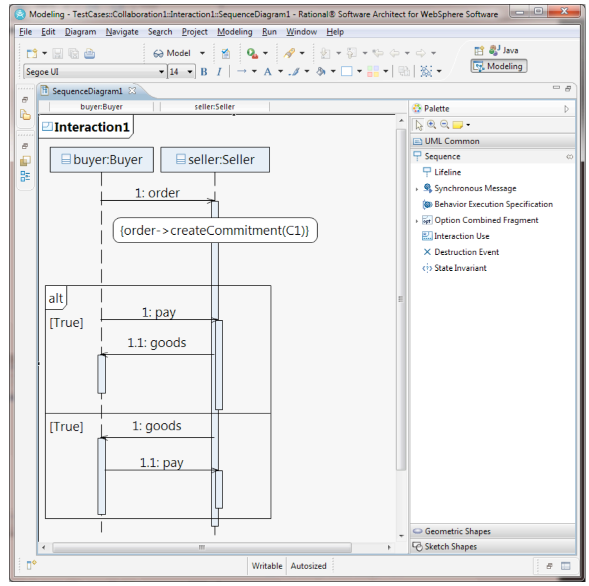


Figure 4: UML sequence diagram modeler.

The Protos engine is a Perl script that (a) invokes the Protos parser APIs to generate a NuSMV input file, (b) invokes the NuSMV model checker on the generated file, and (c) parses NuSMV's output to generate user-friendly output showing the result.

The Protos Parser is the heart of the Protos tool. The parser implements the algorithms from [5] to generate the CTL specifications from the business model ECore file, and to generate finite state machine in NuSMV input language from the UML model file.

2. CONCLUSION AND FUTURE WORK

This paper describes the architecture of Protos, a novel cross-organizational business modeling tool. In future, we plan to enhance usability of the tool by adding features such as drag and drop pattern selection, ability to invoke the Protos Perl script from within Eclipse, and improved user guidance in case of verification errors. We plan to conduct a user-study to evaluate the efficiency and quality of models produced using Protos.

Demonstration Video URL

http://www.youtube.com/watch?v=7UWb-89w6xE&feature=youtu.be
Note: Set the video quality to 720p.

3. REFERENCES

[1] Eclipse. Eclipse modeling project, 2012. http://www.eclipse.org/modeling/emf.

[2] Eclipse. Graphical modeling project, 2012. http://www.eclipse.org/modeling/gmp.

[3] IBM. Ibm rational software architect v8, 2012. http://www.ibm.com/developerworks/downloads/r/architect.

[4] NuSMV. A new symbolic model checker, 2012. http://nusmv.fbk.eu.

[5] Pankaj R. Telang and Munindar P. Singh. Specifying and verifying cross-organizational business models: An agent-oriented approach. *IEEE Transactions on Services Computing*, 4, 2011. In press; available online from IEEE.

Expectation and Complex Event Handling in BDI-based Intelligent Virtual Agents (Demonstration)

Surangika Ranathunga
Department of Information Science
University of Otago
Dunedin 9054, New Zealand
surangika@infoscience.otago.ac.nz

Stephen Cranefield
Department of Information Science
University of Otago
Dunedin 9054, New Zealand
scranefield@infoscience.otago.ac.nz

ABSTRACT

When operating in virtual communities, intelligent agents should maintain a high-level awareness of the physical and social environment around them in order to be more believable and capable. However, due to the inherent differences between virtual worlds and agent systems such as BDI, such a high-level of awareness has not been achieved for IVAs. In this paper we present a system that enables IVAs to maintain a high-level awareness of their environment by identifying complex events taking place in their environment, as well as by being able to monitor for the fulfilment and violation of their expectations.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent systems

General Terms

Design

Keywords

Intelligent Virtual Agents, Expectations, BDI, Complex Event Processing

1. INTRODUCTION

Intelligent Virtual Agents (IVAs) are present in many virtual communities alongside human participants. While interacting with human participants in virtual communities these IVAs are expected to exhibit an acceptable level of awareness of the environment they are operating in.

The task of dynamically perceiving and comprehending what is happening in an agent's surrounding environment is non-trivial, given the inherent differences between virtual worlds and agent systems. First and foremost, there is an information representation gap between agent systems and virtual worlds. Virtual worlds operate with low-level primitive data while agent systems such as those based on the BDI (Belief-Desire-Intention) architecture are declarative, operating at higher abstraction levels. Solutions implemented for this problem have mainly focused only on creating static abstractions of virtual environments. Moreover, virtual worlds operate at a much higher frequency and generate large amounts of low-level sensor data, when compared with agent systems that perceive their environment at a lower frequency. This results in a cognitive overload for the IVA.

In this paper, we present our solution that enables an IVA to dynamically comprehend the abstract events unfolding in its surrounding environment and make use of this knowledge to identify the fulfilments and violations of its expectations. In achieving this, the implemented framework has two main components: The first component is a data processing module that processes the low-level sensor data received from a virtual world to identify domain-specific abstract information that is of interest to an IVA. This process is based on a virtual environment formalism we have developed in previous research [4]. This high-level information is used by the expectation monitor component to identify fulfilments and violations of agent expectations. The agent can also use this high-level information as percepts in its deliberation process.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. SOLUTION OVERVIEW

The first step in dynamically comprehending the surrounding environment is to create a coherent snapshot of the virtual environment, based on the primitive sensor data. This step is important because a piece of sensor data received from a virtual world at a given time instant may not contain the state of all the entities in that environment. However, for the successful implementation of the subsequent steps, it is important that we have a complete view of the environment observable by the IVA at the time instant of the received piece of sensor data. The snapshot generated in the first step accomplishes this requirement. The second step is to identify the static relationships (e.g. spatial or structural) among the entities included in an individual snapshot. This provides the first level of abstraction over the sensor data. In the third step, snapshots enriched with the entity relationship information are subject to complex event recognition techniques to identify the dynamic (i.e. temporal) relationships between entities, thus further abstracting the low-level sensor data.

We have presented an interface that can be implemented by an agent platform to enable its agents to start and stop monitoring for their expectations [3]. Through this interface, monitoring for agent expectations can be delegated to a monitoring service provided by the local agent platform. This enables agents to monitor for the fulfilment and violation of their expectations without relying on a centralised monitoring mechanism. This way, it is possible for agents to have plans that respond to identified fulfilments and violations of their expectations, while being able to make use of well established expectation monitoring techniques.

3. IMPLEMENTATION

In Figure 1, the *Data Processing Module* is the central processing component that identifies high-level domain-specific complex events. The monitoring service contains the logic for identifying fulfilments and violations of expectations delegated by agents.

The *VW Connection Manager* provides an interface to connect agent systems with the given virtual world and our data processing module over a TCP/IP connection. It can accommodate multiple concurrent agents to be deployed in the virtual world[1]. Figure 1 shows how this interface is used to deploy Jason [1] agents in the popular multi-purpose virtual world Second Life[2]. The SL Client module contains logic specific to extracting sensor data from Second Life.

3.1 Data Processing Module

The data processing module has three main processing levels. First, data inference and data amalgamation mechanisms are employed on the received dynamic low-level sensor data of entities (objects and avatars), and snapshots of the virtual environment are created. A snapshot provides a complete view of the environment observable by the agent at a given time instant. Based on our virtual environment formalism, a snapshot at this level contains low-level dynamic property values of entities (e.g. their positions, velocities and the currently played animations), messages exchanged in the public chat channels, and primitive events generated by the changes of dynamic property values of entities.

In the second step, each snapshot is analysed to identify the non-temporal relations between entities. Such relations can include the location of entities with respect to given land marks in the environment, and entities close to a given entity.

The *Data Pre-Processor* is responsible for both these processing steps. It makes use of a *static relation identifier* that contains logic needed to identify relations for a given virtual simulation. This is implemented as an external rule-based dynamic script. Thus this logic is readily customisable for the specific needs of a simulation. It also utilises an external database to store the static information needed to identify these relations.

The third step is to identify the temporal relations included in these snapshots. The *Complex Event Detector (CED)* achieves this task. Currently, we employ the Esper complex event processing engine[3] to identify the high-level temporal relations between the snapshots generated by the *Data Pre-Processor*.

The *Data preparation sub module* processes the snapshots received from the *Data Pre-Processor* into a format suitable for the *CED* module. This way, a new complex event recognition mechanism can be easily employed. Finally, the *Data Post-Processor* amalgamates the identified high-level temporal relations with the original snapshot. It then converts the snapshot to a string representation to be sent over the TCP/IP connection. At this level, the snapshot contains three levels of abstractions of the virtual world sensor data. It is possible to eliminate the inclusion of low-level sensor data in the snapshot, thus reducing the number of percepts sent to the agent. If the snapshot is communicated to the agent only when an interesting temporal relation occurred, this further reduces the amount of information sent to the agent.

3.2 Monitoring for Expectations

Snapshot strings generated by the data processing module are used by the Jason Environment class to prepare percepts for Jason agents. Using a TCP/IP connection in the *EM Connection Manager*, the Environment class also forwards these snapshot strings to a separate monitoring service. The monitoring service implements instances of an expectation monitor developed in previous research [2]. This expectation monitor is implemented in Python, and is integrated with the C#-based monitoring service using IronPython[4]. A single expectation monitor is responsible for monitoring for a specific agent expectation defined as a rule in temporal logic. We have introduced two new Jason internal actions that enable agents to start and stop monitoring (i.e. starting and stopping of expectation monitors) for their expectations [3]. Fulfilments and violations identified by an expectation monitor are communicated back to the corresponding agent as events to be handled by plans.

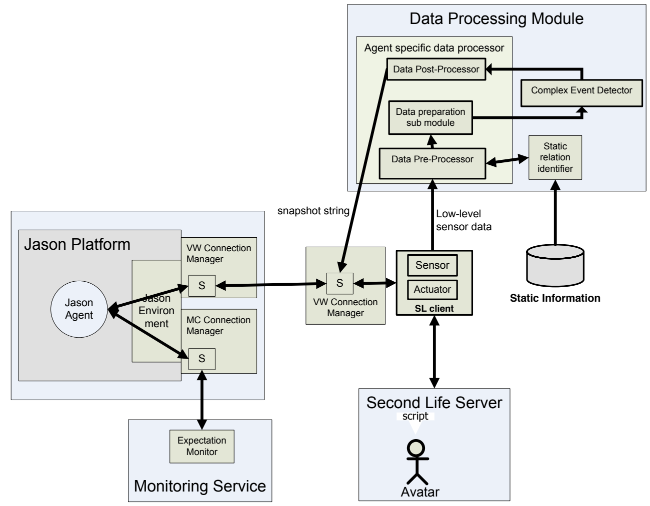


Figure 1: Framework

4. REFERENCES

[1] R. H. Bordini, J. F. Hubner, and M. Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons Ltd, England, 2007.

[2] S. Cranefield and M. Winikoff. Verifying social expectations by model checking truncated paths. *Journal of Logic and Computation*, 21(6):1217–1256, 2011.

[3] S. Ranathunga, S. Cranefield, and M. Purvis. Integrating Expectation Handling into Jason. In *International Workshop on Programming Multi-Agent Systems (ProMAS 2011)*, pages 105 – 120, 2011. `http://www.cs.huji.ac.il/~jeff/aamas11/workshops/ProMAS2011.pdf`.

[4] S. Ranathunga, S. Cranefield, and M. Purvis. Identifying events taking place in Second Life virtual environments. *Applied Artificial Intelligence*, 26:137–181, 2012.

[1]Only one connection is shown in the figure for clarity.

[2]`http://secondlife.com/`

[3]`http://esper.codehaus.org/`

[4]`http://ironpython.net/`

ARGOS: Simulating Migration Processes (Demonstration)

O. Alvarado, N. Ruiz, A. Giret, V. Julián, V. Botti
Universitat Politècnica de Valencia
oalvarado,nruiz,agiret,vinglada,vbotti@dsic.upv.es

V. Perez, R. M. Rodriguez
Boeing Research & Technology Europe
victor.perezvillar,rosamaria.rodriguez@boeing.com

ABSTRACT

In this paper a MAS simulation environment is proposed to simulate the migration process in order to observe dynamic behaviours that may emerge at macro level. As a result of this analysis, it has been possible to identify behaviour patterns that can be represented using agent-based models. Moreover, with this approach predictive techniques has been included in order to represent the complex environment of the migration process and its interaction with other processes like Labour and Financial Markets and Security Forces Management.

Categories and Subject Descriptors

I.6 [**Computing Methodologies**]: SIMULATION AND MODELING

General Terms

Experimentation

Keywords

Emergent behavior, Social simulation

1. INTRODUCTION

The Migration Process is a phenomenon that includes a variety of actors, societies and political issues at different levels. In the migration problem, it is then possible to observe complex interactions among different entities. These interactions have been traditionally represented by mathematical approaches that do not allow including flexibility, autonomy, adaptive and pro-activity features that are present into the dynamic and complex real life migration scenarios. On the other hand, the Multiagent System (MAS) paradigm has been successfully applied in studies related to mass movement in complex environments. In this paper a MAS simulation approach is proposed to simulate the migration process and to model micro-level interaction protocols among the participating entities in order to observe dynamic behaviours that may emerge at macro level. Thus, a MAS model allow to simulate simultaneous behavior of multiple agents to show and predict actions of complex phenomena. The Agent-based Social Simulation (ABSS) studies the social phenomenon by using computational models. These models may represent the people and their interactions with other people as agents [1]. The ABSS is focused on the emergence properties of big agent groups that react to its environment following a set of rules. The MAS paradigm has been applied in studies related to mass movement by defining agent-based models to simulate the rural-urban movement according to social learning [2] [3], an Ethnic Migration Model [4], and Schelling Segregation Model that shows the emergence of socio-spatial patterns [5].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ARGOS DEMOSTRATOR

2.1 The problem

The problem we are trying to simulate is an scenario in which there are complex links among the entities involved in the migration processes (migrants, transport service providers, security forces, labour market mediators, employer service providers and financial market managers) together with the pro-active behaviour of migrants that are trying to move from one country to another. The goal is to model the micro-level agent features of the migration interaction in order to observe the macro-level behaviour of the entire system. To do this we have implemented (i) a MAS supported simulator called ARGOS that is a MAS which simulates the migration scenario, (ii) a web-based simulation player, called ARGOS Player, that displays the simulation execution on top of Google maps, and (iii) ARGOS Data which exports simulation statistics in different formats in order to analyze the simulation execution. We have tested ARGOS with real migration routes from African countries to Spain. In this scenario there are migrants that try to reach Spain by mean of different transportation services, such as plains, boats and buses. The security forces are implemented in the frontier police controls of the country borders. On the other hand, migrants can settled temporarily on different cities of the migration routes to work and save enough money to continue the journey. To simulate this we have implemented labour markets in different route nodes (cities). In the following section ARGOS is detailed.

2.2 The simulator

The simulation environment called ARGOS includes the following key entities of the migration problem: Migrants, Transport Service Providers, Security Forces, Labour Market Mediators, Employer Services Providers, Financial Market Managers, and Radar Stations. ARGOS simulates the migration process in specific regions. The Regions are configured before the simulation is executed. We have tested ARGOS with two specific regions. In the first region, the main source of migrants are Nouakchott and Tidjikja (Mauritania) whose main goal is to reach Cadiz and Sevilla (Spain). In the second region, the main source of migrants are Tambacounda (Senegal), Gao (Mali), Kano (Nigeria), Agadez (Niger), Tamanrasset (Argelia), Maghnia (Algeria), and Oujda (Morroco) whose main goal is to reach Melilla and Ceuta (Spain). The regions include specific migration hubs called nodes in which it is possible to observe migration activities like departure/arrival of Migrants, Labour and Financial Markets Management, and Security Patrols. The simulation includes predefined routes that Migrants use. The routes connect specific region nodes. Thus, the demonstrator includes communication protocols that agents use according to specific scenarios. Financial Market is based on the Leontief Matrix and its behaviour influences the Labour Market behaviour. ARGOS also includes a weather information model between nodes. This model is used in order to compute the probability for successful arrival to nodes when the routes include maritime sectors. The control architecture for maritime borders include: coast modelling, radar stations, arrival paths, algorithms for calculating the detection probability of illegal boats reaching the coast and their graphical representation.

The main scenarios included are: a) Migrants that look for transport to move along migration routes, b) Negotiation among Transport Service Providers and Migrants, c) Movement of migrants with different means of transportation, d) Migrants that look for a job to earn money, e) Labour Market Mediators that execute auctions in order to link position vacancies to workers, f) Security Forces which patrol specific places (not maritime), g) Security Forces which capture illegal migrants that arrive to a specific place, h) Security Forces which verify documentations of captured Migrants, i) Automatic changes on labor demand that affects the Financial Market and influence the positions that Labour Market manages, j) Use of prospective model for decision-making processes made by Immigrants, k) Use of situational awareness by Immigrants during decision-making processes, l) Patrol Nodes by defining control stations that use radars for specific sea areas, m) Security Forces which apply specific algorithms for the evaluation of captured Migrants. ARGOS also allows the User to track the behaviour of agents in specific nodes by using graphic data that show arriving/departing Migrants, position vacancies and dynamic payment by capabilities, changes on Financial Markets and results from the security patrol execution (Migrants that have been retained and released), and data of radar stations (illegal transports detected or not). Moreover, ARGOS Data generates simulation data that is used for external analysis. This generated data can be used to analyze previous situations and predict future actions. The prospective model used by Immigrants takes into account the incomes, outcomes and her/him current wealth to determine if an Immigrant can stay in its current node or if it must try to move to another. Transport Service Providers and Labour Market Mediators publish Situational Awareness information about: transport proposals and labour vacancies. When an Immigrant is reasoning about to where he/she can move, he/she evaluates the situation applying a prospective decision model taking into account the posts that are in the Situational Awareness and its own attributes such as his/her own risk perception or available money. The Security Forces use radars on specific sea areas to detect illegal transports (boats) that are trying to reach Spain. A transport can be detected by one or more radars. It is possible to get the performance of radars (number of illegal transports detected). Therefore, Security Forces also include the use of Retention Centers to temporally retain Immigrants. Security Forces apply different algorithms to evaluate the people that have been retained. During simulation, Migrants share their experiences about the status of Labour Markets and the migration routes (as part of the Social Situational Awareness) with other migrants. The migrants take into account the shared information evaluating it according to the influence level of the migrant that have posted it. The information that migrant receives and the programmed events that the user can introduce in Argos (change on employment rate, change on cost living, change on security forces strictness) allows updating its environment knowledge as part of its Individual Situational Awareness. The migrant uses the Individual Situational Awareness during its decision making process.

3. CONCLUSIONS

In this paper a MAS simulation approach, called ARGOS[1] , has been presented, which allows the simulation of the migration process and to model micro-level interaction protocols in order to observe dynamic behaviours that may emerge at macro level. Some features that ARGOS includes are: decision algorithms that Security Forces uses on borders control, the activation of radars at specific maritime control stations, a blackboard for Social Situational Awareness that is fed by the Transport Services Suppliers and Labour Services Suppliers and consulted by Immigrants. Argos also includes too a prospective model for decision-making processes that Immigrants made when they have to decide if they can stay in the current place or if they have to move to another place, and a Real-life based algorithms that Security Forces use when Immigrants are captured. The proposed approach allows the improvement of maritime frontiers monitoring by using maritime control stations and the definition of a social network that allows connecting the immigrants by defining a specific influence level among them.

4. REFERENCES

[1] Gotts, N., Polhill, J. y Law, A.: Agent-based simulation in the study of social dilemmas. In: Artificial Intelligence Review, 19 (2003), p. 3-92
[2] Silveira, J.J., E.A., Penna, T.: Agent-based model to rural-urban migration analysis. Physica A: Statistical Mechanics and its Applications (364), 445-456 (2006)
[3] Espindola, A.L., Silveira, J.J., Penna, T.: A harris-todaro agent-based model to rural-urban migration. Brazilian Journal of Physics (36-3A), 603-609 (2005)
[4] Makowsky, M., et al.: An agent-based model of crisis-driven migration. Proc. of the Complex Systems Summer School. New Mexico. (2006)
[5] Schelling, T.: Models of segregation. American Economic Review, (59-2), 488-493 (1969)

[1] This research is partially funded by the Spanish Center for the Development of Industrial Technology as part of the Ingenio Program. This support is gratefully acknowledged.

CALU: Collision Avoidance with Localization Uncertainty (Demonstration)

Daniel Claes
Maastricht University
P.O. Box 616, 6200MD
Maastricht, The Netherlands
danielclaes@me.com

Daniel Hennes
Maastricht University
P.O. Box 616, 6200MD
Maastricht, The Netherlands
daniel.hennes@gmail.com

Karl Tuyls
Maastricht University
P.O. Box 616, 6200MD
Maastricht, The Netherlands
k.tuyls@maastrichtuniversity.nl

Wim Meeussen
Willow Garage
68 Willow Rd., Menlo Park
CA 94025, USA
meeussen@willowgarage.com

ABSTRACT

CALU is a multi-robot collision avoidance system based on the velocity obstacle paradigm. In contrast to previous approaches, we alleviate the strong requirement for perfect sensing (i.e. global positioning) using Adaptive Monte-Carlo Localization on a per-agent level.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Experimentation

Keywords

multi-robot systems, optimal reciprocal collision avoidance, adaptive monte-carlo localization

1. INTRODUCTION

Local collision avoidance is the task of steering free of collisions with static and dynamic obstacles, while following a global plan to navigate towards a goal location. Local collision avoidance differs from motion planning, global path planning and local path planning. In motion planning the environment of the robot is assumed to be deterministic and known in advance, thus allowing to plan a complete path to the goal. Global path planners usually operate on a static map and find either the minimum cost plan (e.g. using A* or Dijkstra's algorithm) or any valid plan (e.g. sample based planners). Local path planners, such as Trajectory Rollout and Dynamic Window Approaches (DWA), perform forward simulations for a set of velocity commands; each resulting trajectory is scored based on proximity to the goal location and a cost map built from current sensor data. In principle this allows to stay clear of dynamical obstacles; however, in multi-robot settings two problems arise:

1. Robots are not merely dynamic obstacles; each robot itself is a pro-active agent taking actions to avoid collisions. Neglecting this might lead to oscillations and thus highly inefficient trajectories or even collisions.
2. The sensor source (e.g. laser range finder) is usually mounted on top of the robot's base to allow for a maximal unoccluded viewing angle. In a system with homogenous robots this implies that there is very little surface area that can be picked up by the sensors of other robots and thus prevents the robots from observing each other.

Local collision avoidance addresses these challenges and is an important building block in any robot navigation system targeted at multi-robot systems. Although robot localization is a requirement for collision avoidance, most approaches assume perfect sensing and positioning and avoid local methods by using global positioning via an overhead tracking camera - or are purely simulation based. Nevertheless, to be able to correctly perform local collision avoidance in a realistic environment, a robot needs a reliable position estimation without the help of external tools.

Our approach, Collision Avoidance with Localization Uncertainty (CALU) builds on two techniques: Optimal Reciprocal Collision Avoidance (ORCA) [3] in combination with Adaptive Monte-Carlo Localization (AMCL) [1]. Thus effectively alleviating the need for global positioning by decentralized localization on a per-agent level. We provide a solution that is situated in between centralized motion planning for multi-robot systems and communication-free individual navigation. While actions remain to be computed independently for each robot, information about position and velocity is shared using local inter-robot communication. This keeps the communication overhead limited while avoiding problems like robot-robot detection. CALU bounds the error introduced by localization [2] and combines the computation for collision-free motion with localization uncertainty.

2. PROBLEM DESCRIPTION

ORCA (and all its variants) does not require any inter-robot negotiation to find optimal collision free motion trajectories and is hence in principal fully distributed. However, all methods require perfect information about the positions, velocities and shapes of all other robots. In order to preserve the distributed nature of this approach, robots need to

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

be able to accurately identify other robots using on-board sensors; furthermore, positions and velocities have to be deduced from the same data. The list of typical sensors for mobile robots includes stereo cameras, laser range finders and lately 3D image sensors (e.g. Microsoft Kinect). These sensors deliver large data-streams that require considerably computational power to process even for the detection and classification of static obstacles.

The computational requirement is not the only problem when considering robot-robot detection. As low-end laser range finders (e.g. Hokuyo URG-04LX) become widely available even for mobile robotic projects on a small budget, they are the preferred sensor choice due to there high accuracy, resolution and field of view. However, robot-robot detection based on laser range finders remains challenging.

Previous approaches have worked around these problems by providing global positioning to all robots based on an overhead tracking camera. Such a system is not distributed since a host computer connected to the camera needs to process the sensor data and communicate with all robots to provide position and velocity data.

3. APPROACH

We propose to utilize agent-based localization and inter-robot communication to provide a system that is more realistic in real-world scenarios (i.e. without the need for external positioning data) and also more robust (i.e. single component failure does not lead to system failure). Our approach, called Collision Avoidance with Localization Uncertainty (CALU), results in a fully decentralized system that uses local communication to share robot state information in order to ensure smooth collision free motion. Below we describe the four key components of this approach.

Platform: The robots are assumed to be differential drive robots. Required sensors are a laser range finder and wheel odometry. For simplicity we assume a circular footprint; other shapes can be approximated by the circumscribed radius. In order to connect the different subsystems, including device drivers and software modules, we use ROS[1].

Sensor processing and localization: Each robot integrates wheel odometry data which is in turn used to drive the motion model of AMCL , hence tracking the pose of the robot. Laser range finder scans are used in the update phase of AMCL. The uncertainty of the current localization, i.e. the spread and weight of the particles, is taken into account for the calculation of collision free velocities. We assume a prior static map that is used for localization and available to all robots, thus providing a consistent global coordinate frame.

Inter-robot communication: Each robot broadcasts its position and velocity information in the global coordinate frame on a common ROS topic. Each robot also subscribes to the same topic and caches position and velocity data of all other robots. Message delays are taken into account and positions are forward integrated in time according to the motion model of robots using the last known position and velocity information.

Collision avoidance: ORCA is used to compute optimal collision free velocities according to the aggregated position and velocity data of all surrounding robots. As a last step we incorporate localization uncertainty in the ORCA computation.

[1]For more information see: `http://www.ros.org`

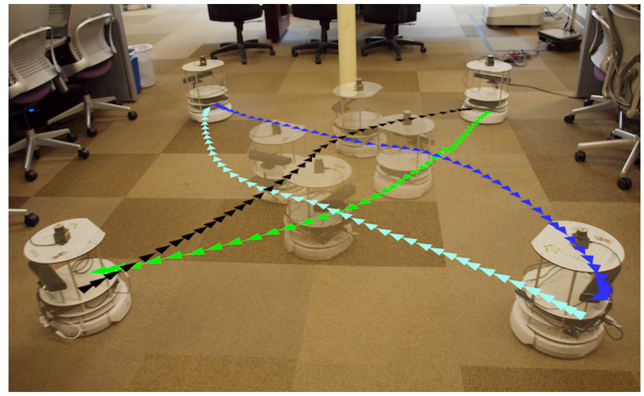


Figure 1: Real-world collision avoidance with four differential drive robots using CALU.

4. DEMONSTRATION

We will demonstrate our approach in simulation and a real-world setting. In simulation, robots are positioned on a circle and the goals located on the antipodal positions, i.e. each robot's shortest path is through the center of the circle. For experiments and detailed results of the proposed system, we refer to [2].

In addition to simulation runs, we present our approach on up to four differential drive Turtlebots[2]. The robots are based on the iRobots Create platform and have a diameter of 33.5 cm. In addition to the usual sensors, they are equipped with a Hokuyo URG laser-range finder to enable better localization in large spaces. All computation is performed on-board on a Intel Atom D525 1.8GHz dual core CPU netbook. Communication between the robots is realized via a 2.4 GHz WiFi link.

Figure 1 shows the trajectories of an example run of the four robots using CALU. The initial positions are approximately 3.5 meters apart; the goal location are set to the diagonally opposing start locations. The system successfully avoids collision and produces smooth paths; except for a small jump in the localization that can be observed in the path of robot starting in the upper right corner.

A demonstration video is available at:
`http://swarmlab.unimaas.nl/papers/aamas-2012-calu/`.

5. REFERENCES

[1] Dieter Fox. Kld-sampling: Adaptive particle filters. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[2] Daniel Hennes, Daniel Claes, Wim Meeussen, and Karl Tuyls. Multi-robot collision avoidance with localization uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, 2012.

[3] Jur van den Berg, Stephen Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, volume 70, pages 3–19, 2011.

[2]For more information see: `http://turtlebot.com`

Stigmergic Coverage Algorithm for Multi-Robot Systems (Demonstration)

Bijan Ranjbar-Sahraei
Maastricht University
PO Box 616, 6200 MD
Maastricht, The Netherlands
b.ranjbar@ieee.org

Gerhard Weiss
Maastricht University
PO Box 616, 6200 MD
Maastricht, The Netherlands
gerhard.weiss@maastrichtuniversity.nl

Ali Nakisaee
National Organization for
Development of Exceptional Talents
(NODET), Shiraz, Iran
ali.n123@gmail.com

ABSTRACT

We demonstrate the realization of stigmergic coverage for multi-robot systems. Compared to current state-of-the-art algorithms for multi-robot coverage, our Stigmergy-based Coverage algorithm (StiCo) has several key advantages. In particular, it does not need direct robot-robot communication. Moreover, this algorithm does not require any prior information about the environment. Simulation results illustrate robustness, scalability and simplicity of the algorithm.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics—*Autonomous vehicles, Commercial robots and applications*

General Terms

Algorithms, Design

Keywords

multi-robot coverage, stigmergy, multi-agent systems

1. INTRODUCTION

Recent years have shown a rapidly growing interest in the automated coverage of complex, large and unknown environments through teams of cooperating autonomous robots. The main reason for this interest in multi-robot coverage lies in its broad range of potential applications in civil, industrial and military domains.

Current research mainly focuses on graph-based approaches (e.g. [1–3]) and Voronoi-based approaches (e.g. [4, 5]). The basic idea underlying graph-based approaches is to model the subregions of an environment and connections between them with a graph and develop graph search algorithms (e.g. DFS, BFS) for exploration/coverage of this graph. A practical drawback of graph-based approaches, however, is that they require to map the environment to a graph-like structure, which is computationally expensive and inapplicable in complex large environments. In contrast, Voronoi-based approaches aim at spreading out the robots over an environment by positioning each robot at the centroid of its Voronoi cell. Unfortunately, Voronoi-based methods inherently suffer from high computational complexities, too. In addition, these methods require direct communication among robots which is not applicable in limited-communication environments.

This paper investigates an alternative approach to multi-robot coverage, called StiCo, which is based on the principle of stigmergic (pheromone-type) coordination as known from ant societies. Compared to graph-based approaches, our approach is a model-free coverage algorithm implemented on memory-less simple robots. Moreover, while our approach avoids the complexity of available Voronoi-based approaches, it achieves a Voronoi-like segmentation and coverage of the environment in a very robust way on the basis of indirect communication only. The main characteristics of our approach are its simplicity, robustness, scalability and flexibility, as described below and illustrated in the video demo available at:
http://swarmlab.unimaas.nl/papers/aamas-2012-demo-2

2. THE StiCo APPROACH

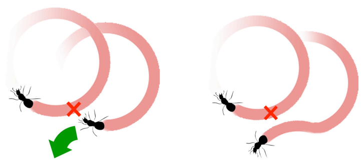
The StiCo approach follows the principle of indirect, pheromone-based coordination. StiCo assumes that there is a group of robots which have the capacity to communicate indirectly by depositing markers (also called pheromones) in the environment for noticing margins of their territories to the others. In addition, each robot is equipped with two simple sensors (in the front-left and front-right directions like an ant antenna), capable of detecting immediate pheromones.

It is demonstrated that the developed coverage algorithm causes the environment to be partitioned into smaller regions (called as *robot territory*), while margins of each region are guarded by an individual robot. StiCo uses pheromone detections to recognize the already covered areas and guide the robots to uncovered environments. This algorithm does not need any memory or computation ability.

In SitCo, each robot starts to move on a circle with a predetermined radius. Based on the circling direction (CW or CCW), one sensor would be considered as the interior sensor and the other as the exterior one. When the interior sensor detects pheromone, the robot changes circling direction immediately as shown in Figures 1a,1b. Otherwise, if exterior sensor detects pheromone, the robot rotates in the same direction until it doesn't detect pheromone any more. Moreover, the amount of pheromone deposited by each robot

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

is adjusted based on pheromone evaporation rate, in a way that robots do not collide with their own pheromones.



(a) (b)

Figure 1: StiCo coordination principle (a) before pheromone detection (b) after pheromone detection

This simple algorithm is detailed in Algorithm 1.

Algorithm 1 StiCo Algorithm

Require: Each robot can deposit/detect pheromone trails
  Initialize: Choose circling direction (CW/CCW)
  loop
    while (no pheromone is detected) **do**
      Circle around
      deposit pheromone
    end while
    if (interior sensor detects pheromone) **then**
      Reverse the circling direction
    else
      while (pheromone is detected) **do**
        Rotate
      end while
    end if
  end loop

3. SIMULATION RESULTS

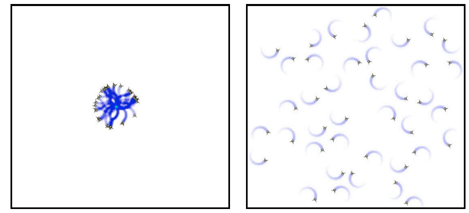
In this section, we demonstrate the evolution of our simple StiCo algorithm on a robotic swarm of identical members in a $40m \times 40m$ field. Simulations are implemented in Microsoft Visual C++. The pheromones are simulated with a high resolution, equal to 300×300 and the evaporation rate is $10units/s$. The linear velocity of each robot is $2m/s$, and the angular velocity is set to $\pm 1.0rad/s$. Each robot deposits 25units of pheromone in each iteration, and has two pheromone-sensors which can detect pheromones from a distance of $2m$. We pay careful attention to numerical accuracy and optimization issues in the pheromones update policy. Execution of coverage algorithm for 40 robots which move based on StiCo is illustrated in Figure 2.

In order to demonstrate potential capabilities of this simple algorithm, we consider a non-convex unknown environment as shown in Figure 3a. This environment can represent a devastated area after earthquake, or a street map in an emergency condition. 40 robots are initiated at the center of the environment. The coverage steps are illustrated in Figures 3a-3c. (In this simulation artificial pheromones are deposited on the margins of obstacles to make them detectable for robots).

4. CONCLUSIONS AND FUTURE WORK

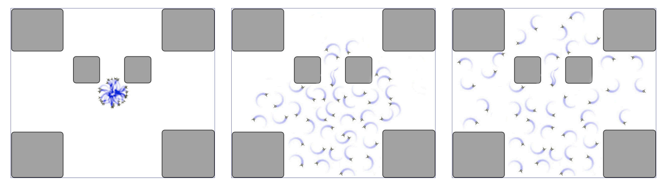
In this paper we addressed a coverage problem called StiCo for a group of robots which coordinate indirectly via ant-like, stigmergic communication. We assumed that robots can not communicate directly with each other. Therefore, a stigmergic communication through depositing pheromones in the environment were proposed. Fully distributed motion policies were designed which concluded to robust coverage of the unknown environment. Efficiency of StiCo algorithm was demonstrated with illustrative simulations.

As future work, we are planning to improve the behavior of presented algorithm and develop a comprehensive probabilistic framework for StiCo which can help us to prove its efficiency in mathematical form. Moreover, we are investigating how to implement StiCo on real swarms.



(a) (b)

Figure 2: Evolution of StiCo in a simple environment (Blue shadows are deposited pheromones) (a) Initial snapshot (b) Final snapshot



(a) (b) (c)

Figure 3: Evolution of StiCo in a complex environment. (a) Initial snapshot (b) Intermediate snapshot (c) Final snapshot

5. REFERENCES

[1] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 918–933, 1999.

[2] V. Yanovski, I. A. Wagner, and A. M. Bruckstein, "A distributed ant algorithm for efficiently patrolling a network," *Algorithmica*, vol. 37, pp. 165–186, 2003.

[3] Y. Elor and A. Bruckstein, "Autonomous multi-agent cycle based patrolling," in *Swarm Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6234, pp. 119–130.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 243 – 255, april 2004.

[5] M. Schwager, D. Rus, and J. J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *International Journal of Robotics Research*, vol. 30, no. 3, pp. 371–383, 2011.

Infraworld, a Multi-agent Based Framework to Assist in Civil Infrastructure Collaborative Design (Demonstration)

Jaume Domínguez Faus
Centre for 3D GeoInformation
Ålborg Universitet
9220 Ålborg, Denmark
jaume@land.aau.dk

Francisco Grimaldo
Departament d'Informàtica
Universitat de València
Av. de la Univesitat s/n
Burjassot (València), Spain 46100
francisco.grimaldo@uv.es

ABSTRACT

Infraworld is an experimental framework for Computer Aided Engineering (CAE) systems which is designed for distributed design. The framework is based on Multi-agent systems that allow engineers to synchronize their work by keeping track of their changes and facilitating the detection and management of semantic conflicts that arise when different actors are working in parallel. Conflicts are detected according of each engineers semantics which are defined by using OWL ontologies and SWRL rules. When they are detected, the framework allows solving them by negotiating possible alternatives. Then the alternatives are evaluated by expressing preferences and the picked alternative, being the one that maximizes the global welfare, is applied in all the models in the distributed environment. The system is completed with a machine learning module that allows the agents to suggest similar solutions to future conflicts with similar semantic context.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent Systems

Keywords

Distributed decision making, Semantic conflict detection

1. INTRODUCTION

Civil Infrastructure design has evolved from its very initial steps of paper design to Computer Aided Engineering tools that help in its complex tasks. These tools include sets of predefined features to compute concrete situations such as, e.g, the distribution of forces in a structure. However, these works are always carried out by sets of teams that specialize in some profile of the multidisciplinar Civil Infrastructure project. Unfortunatelly, most of the existing tools are geared to individual workplaces. Although there are some efforts to make this work more distributed like file repositories or the most advanced BIM [1] servers there is low support for handling conflict situations caused when several separated works have to be put together to align all the project designs.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

Unfortunately, despite previous works like [2] the process of alignment still a prominent manual one. The civil engineers meet regularly to align their works and in the best case they can do it using 3D representation of the models that are navigated and analyzed by the authors in order to find conflicts. Thus, there is a need for this process to be automated. When some error is not detected in design time and the project is already in the construction phase, it is very costly to fix it. And this situation is still happening today with the corresponding project overheads and delays. Studies estimate them at around 5-10% of the total budget in average [1]. To fulfill this gap, we present the Infraworld framework. Infraworld allows the definition of the semantics of a model on a per-engineering-profile basis. The semantics are defined by sets of OWL [3] ontologies from which the base knowledge is built, and the conflicts are detected by using SWRL rules. They are used by the JADE agents that control the evolution of the project in each workstation and allow, in front of a conflict, to negotiate how to solve it with the other stakeholders of the project.

2. THE INFRAWORLD FRAMEWORK

The Infraworld framework is composed of three main logical pieces: 1) a reasoning engine that can be used by Validator agents, 2) the collaboration module that defines the negotiation protocol that is carried out when solving conflicts, and 3) a learning module that, when the negotiation ends, captures the solution applied and the context of the conflict in order to infer solutions for future similar conflicts.

2.1 Reasoning Engine

Unlike the usual systems in which the conflict detection is based on pure geometric overlapping of the objects, also called Features, in the model, Infraworld framework extends the concept of conflict to the semantics. To do that, there is a Core Ontology that defines the concepts of Feature, Attribute, Geometry, and Relationship. A Feature represents an entity of the world and it is composed of the Attributes that parameterize it, the Geometry that gives its physical shape in the world and its Relationships with other Features in the model. The second level of abstraction is the FeatureCatalog ontology which gives the meaning of what Feature represents. For instance there is a Building concept in this ontology that when applied to a Feature defines it as a building.

Beyond these ontologies, each engineering profile provides with their own. This approach allows a feature to be treated differently depending on the point of view. For instance, a sewerage conduction might only be an obstacle for an engineer that is planing a gas supply conduction and only needs to ensure it does not overlap his designs. However, the engineer designing the sewerage has to ensure that there are no other conductions underneath. In other words, the sewer profile needs to take care of other specific problems than just regular geometry overlaps.

To complete the knowledge, SWRL rules are supported. They are also provided by each profile and they are meant for detecting the conflicts. These rules consist of an antecedent and a consequent. The antecedent is evaluated against the model and when it resolves to true, then the consequent is said to be also true. For instance a conflict like the one explained above could be captured with a SWRL rule as follows:

$$\begin{aligned}&Conduction(?c1) \wedge Sewerage(?c2) \wedge isBelow(?c1, ?c2)\\&\qquad \rightarrow PositionNotAllowedConflict(?c1, ?c2)\end{aligned}$$

This rule would mark features that match the condition expressed in the antecedent (first row) as a PositionNotAllowedConflict.

2.2 Collaboration Module

The collaboration module defines a Multiagent society (see Figure 1) composed of Validators, Negotiators and Coordinators. As the engineers work in parallel, changes are performed to the model. These changes are monitorized by the Validator agent that executes the Reasoning Engine when changes to the model are detected. When conflicts are detected as a result of the execution of the reasoner, the engineers have the possibility to solve the conflict by means of negotiation. The negotiation is based on MARA (Multi Agent Resource Allocation) as a general mechanism to make socially acceptable decisions and follows a ContractNet-like protocol that is executed when a Validator agent wants to solve a conflict. It consists of two round negotiation. In the first round, the Coordinator agent notifies all the Negotiators that a conflict has been detected and asks for alternatives to solve it. Then, Negotiators record the alternatives provided by the engineers and send them back to the Coordinator agent. The Coordinator agent collects all the alternatives and send them again, in a second round, to the Negotiators so that their engineers can provide with preferences. The engineers express their preferences by giving a score ranging from -5 to 5 to each alternative and the Negotiators send them to the Coordinator. The Coordinator picks the alternative that maximizes the global welfare as the solution and notifies this decision to the Negotiators. This solution is finally applied to all the models in the distributed environment.

2.3 Learning Module

The third module is aimed to learn from the engineers experience and behavior. After a conflict has been solved in a negotiation the context of the conflict, i.e. the Features and their Attributes that were in conflict, as well as the related Features this Feature may have by means of its Relationships, are registered for future processing. If the same conflict occurs in the future, the Validator agent might find coincidences in the history and suggest the past solution to help the engineers to find a solution for it.

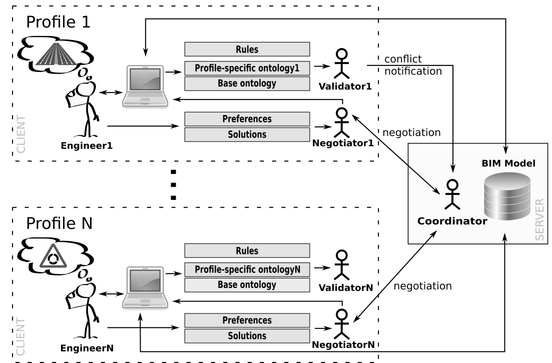


Figure 1: Overview of the system

3. EXPERIMENTS AND RESULTS

We applied our framework to two use cases to test it. The Reasoning Engine showed to be adequate to detect project-specific semantic problems. The Collaboration Module allowed to perform the negotiation. Finally, the Learning Module could suggest solutions for repeating problems.

- **Urban Development Use Case** consisting of a model with 4107 ontology instances covering the development of the city of Drammen in Norway. In this use case two profiles (a traffic engineer and a builder) were designing the model. The goal was to ensure that the road network was not exceeded by the population living in the buildings being planned.
- **Power Plant Electricity Installation Use Case** with 4592 ontology instances in which two engineering profiles in charge of the foundations and the wiring structure had to solve conflicts regarding the bolts connecting both elements that were misplaced. Since this conflict was repeating, the Learning Module helped solving it by automatically suggesting solutions.

4. ACKNOWLEDGEMENTS

This work was supported by Norwegian Research Council, Industrial PhD scheme case no: 195940/I40 through Vianova Systems AS, Norway; the Spanish MICINN, Consolider Programme and Plan E funds, European Commission FEDER and Universitat de València funds, under Grants CSD2006-00046, TIN2009-14475-C04-04, UV-INV-AE11 - 40990.62

5. REFERENCES

[1] C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM Handbook, a guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors.* John Wiley & Sons, Inc. New Jersey, 2008.

[2] F. Peña-Mora and C.-Y. Wang. Computer-supported collaborative negotiation methodology. *Journal of Computing in Civil Engineering*, pages 64–81, April 1998.

[3] W3C OWL Working Group. OWL 2 web ontology language document overview. Technical report, W3C, Oct. 2009.

AgentPolis: Towards a Platform for Fully Agent-based Modeling of Multi-Modal Transportation (Demonstration)

Michal Jakob, Zbyněk Moler, Antonín Komenda†, Zhengyu Yin, Albert Xin Jiang, Matthew P. Johnson*, Michal Pěchouček† and Milind Tambe*
† Agent Technology Center, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic
{jakob, moler, komenda, pechoucek}@agents.fel.cvut.cz
* Computer Science Department, University of Southern California, Los Angeles, CA. USA.
{zhengyuy, jiangx, matthepj, tambe}@usc.edu

ABSTRACT
AgentPolis is a fully agent-based platform for modeling multi-modal transportation systems. It comprises a high-performance discrete-event simulation core, a cohesive set of high-level abstractions for building extensible agent-based models and a library of predefined components frequently used in transportation and mobility models. Together with a suite of supporting tools, AgentPolis enables rapid prototyping and execution of data-driven simulations of a wide range of mobility and transportation phenomena. We illustrate the capabilities of the platform on a model of fare inspection in public transportation networks.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent Systems; I.6.3 [**Simulation and Modeling**]: Applications

General Terms
Algorithms, Design, Experimentation

Keywords
agent-based modeling, multi-modal mobility, simulation, platform, transportation, fare inspection

1. INTRODUCTION
Over the last two decades, high-level, equation-based transportation modeling has been being replaced by micro-simulation approaches, which achieve higher accuracy by representing transportation systems at the level of individual people and vehicles [2]. Micro-simulation is particularly popular for vehicle traffic modeling, where it is now part of several commercial packages. Adoption of micro-simulation is slower in *mobility models*, which aim to capture how *people* (and freight), rather than just vehicles, move around in space and time using different means of transportation. The state-of-the art *activity-based approaches* model mobility by encoding people's choices regarding the type, location, and time of their activities and transportation between them.

MATSim [1] is the best known platform for activity-based mobility micro-simulation. Although termed agent-based and supporting individual-level modeling, MATSim treats individuals as passive data structures which can only be updated synchronously by central modules at infrequent, predefined points in time. This reduces the ability to add new types of agents to the model and to represent dynamic and multi-agent behavior.

To overcome these limitations, AgentPolis adopts a *fully agent-based* modeling approach. Individual entities of a transportation system are represented as autonomous agents with continuous, asynchronous control modules and the ability to interact freely with the environment and other agents. The agent-based approach reduces coupling and allows modeling scenarios in which agents adjust their plans at any time during the day based on their observations of the environment and/or communication with other agents. Agent-centric design also makes AgentPolis models usable as testbeds for evaluating innovative multi-agent mechanisms for transportation control and management.

2. PLATFORM OVERVIEW
The Java-based AgentPolis platform (see Figure 1 for an architecture overview) was designed to provide maximum reusability of elements shared by most models of transportation and mobility phenomena while allowing maximum flexibility in implementing model-specific parts.

2.1 Simulation Core
In the core of AgentPolis is a discrete-event simulation platform based on the Alite multi-agent toolkit[1]. The platform consists of a high-performance discrete-event processing engine and a set of domain-independent abstractions for building discrete-event agent-based models of large systems. The discrete-event model used is more resource-efficient compared to fixed-time-step execution models used by most agent-based simulation platforms. The refined set of abstract classes and interfaces compliant with the agent-based design paradigm provides a coherent foundation for building diverse agent-based models. This contrasts with highly purpose-specific designs of existing transportation mo-

[1] http://agents.fel.cvut.cz/projects/#alite

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

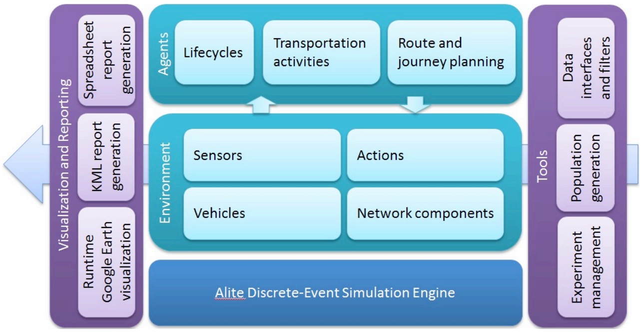


Figure 1: AgentPolis architecture.

dels, which make horizontal extension of models difficult.

2.2 Transportation Domain Model Library

Built on top of the base domain-independent abstractions, the transportation domain library provides a set of specific components for rapidly building transportation and mobility models. The library consists of the following software classes and modules:

- *transportation network components*—software classes for building transportation environments, in particular road, railway and metro networks, road intersections, public transportation stops and interchanges;
- *vehicles*—software classes representing vehicles (cars, buses, trains) and their properties;
- *transportation actions and sensors*—software classes mediating access between agents and their surrounding transportation environment, e.g., getting on/off a vehicle, moving along a street segment or detecting an arriving train;
- *transportation activities and lifecycles*—reactive control structures that can be composed to create a desired agent's behavior, e.g., travelling on a public transport or driving a vehicle between locations;
- *route and journey planning*—functional modules providing agent reasoning capabilities, in particular efficient route planning in road networks and multi-modal journey planning with public transportation services.

2.3 Simulation Tools

Rapid construction, execution and experimentation with AgentPolis models is supported by a range of tools:

- *Data interfaces and filters* allow to work directly with transportation-related data in standard formats, including OpenStreetMap format for map data, and Google Transit Feed Specification for public transportation networks and timetables.
- *Population generation tools* allow generating large numbers of agents with realistic distributions of demographic attributes (age, gender, income, car ownership etc.) based on real-world census data.
- *Experiment configuration, management and deployment tools* enable defining in a compact form experiment scenario batches and automatically executing them on available computing resources.
- *Visualization and reporting tools* allow viewing simulation runs as well as simulation results, including their geospatial and temporal context and aggregation over multiple runs and scenarios, in an interactive browser based on Google Earth.

3. APPLICATION TO FARE INSPECTION

Simulation of public transportation fare inspection is one of the models built using the AgentPolis platform. The model allows evaluating the effectiveness of fare inspection strategies provided by human experts or computational tools, in particular the TRUSTS (Tactical Randomization for Urban Security in Transit Systems)[3] system for scheduling randomized fare inspection patrols. TRUSTS adopts a game-theoretic approach, modeling the problem as a leader-follower Stackelberg game, and employs a novel compact representation of the mixed strategies as flows in a history-duplicate transition graph.

The agent-based model of fare inspection comprises two types of agents—the *FareAwarePassenger* has a standard daily travel pattern extended with ticket purchase logic determining whether a ticket should be purchased for a particular journey; the *TicketInspector* agent inspects passengers on specific trains and at specific stations according to a given patrolling schedule, provided, e.g., by the TRUSTS scheduler. The performance of each inspection strategy can be tested against passengers with different levels of rationality and observability of the environment. A number of performance metrics can be measured including fines collected, revenue lost and inspection coverage. So far, the fare inspection model was developed for the Los Angeles Metro system and involves the simulation of almost 400 thousand rides on five metro lines a day.

We are developing other models using the AgentPolis platform, including real-time ride sharing, auction-based taxi allocation and on-demand parcel delivery logistics. More information about new developments can be found on the platform's website[2].

Acknowledgements

Funded by Ministry of Education, Youth and Sports of Czech Republic (grant No. LD12044) and European Commission FP7 (grant agreement No. 289067). We acknowledge support from MURI grant W911NF-11-1-0332.

4. REFERENCES

[1] M. Balmer, K. Meister, M. Rieser, K. Nagel, and K. W. Axhausen. Agent-based simulation of travel demand: Structure and computational performance of MATSim-T. In *TRB Conference on Innovations in Travel Modeling*, 2008.

[2] K. Bernhardt. Agent-based modeling in transportation. *Artificial Intelligence in Transportation*, E-C113:72–80, 2007.

[3] A. Jiang, Z. Yin, C. Kietkintveld, K. Leyton-Brown, T. Sandholm, and M. Tambe. Towards optimal patrol strategies for fare inspection in transit systems. In *AAAI Spring Symposium on Game Theory for Security, Sustainability and Health*, 2012.

[2] `http://agents.fel.cvut.cz/projects/agentpolis/`

Distributed Consensus for Interaction between Humans and Mobile Robot Swarms (Demonstration)

Alessandro Giusti, Jawad Nagi, Luca M. Gambardella, Gianni A. Di Caro
Dalle Molle Institute for Artificial Intelligence (IDSIA)
Lugano, Switzerland
{alessandrog,jawad,luca,gianni}@idsia.ch

ABSTRACT

The purpose of the demonstrator is to present a novel system for gesture-based interaction between humans and a swarm of mobile robots. The human interacts with the swarm by showing hand gestures using an orange glove. Following initial hand glove detection, the robots move to adapt their positions and viewpoints. The purpose is to improve individual sensing performance and maximize the gesture information mutually gathered by the swarm as a whole. Using multi-hop message relaying, robots spread their opinions and the associated confidence about the issued hand gesture throughout the swarm. To let the robots in the swarm integrate and weight the different opinions, we developed a distributed consensus protocol. When a robot has gathered enough evidence, it takes a decision for the hand gesture, and sends it into the swarm. Different decisions compete with each other. The one assessed with the highest confidence eventually wins. When consensus is reached about the hand gesture, the swarm acts accordingly, for example by moving to a location, or splitting into groups.

The working of the system is shown and explained in the video accessible at the following address: `http://www.idsia.ch/~gianni/SwarmRobotics/aamasdemo.zip`.

Categories and Subject Descriptors

I.2.9 [**Robotics**]; I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination; C.2.4 [**Computer Communication Networks**]: Distributed applications

General Terms

Algorithms

Keywords

Gesture recognition, Distributed consensus, Swarm robotics

1. INTRODUCTION

We consider the problem of the *interaction between humans and robotic swarms*. The purpose is to let a human communicating commands to be executed by the swarm (e.g., split in two groups). The problem is particularly challenging since the robots in the swarm can be spread in different positions in the environment and be engaged in tasks of their own when the command is issued. Moreover, the robots typically used in swarm robotics are relatively simple and have limited processing capabilities. The task of the robots is to *detect* and *understand* the command, and collectively reach a *distributed consensus* about it in order to actuate its execution.

We use *hand gestures* as mean for human-swarm communication. In our scenario, a hand gesture encodes a command, that the swarm will execute. Hand gestures are a powerful and intuitive way to communicate, and do not require the use of additional devices. However, real-time *vision-based* recognition of hand gestures is a challenging task for the single robot, due to the limited processing power and field of view of robots that we use, the *foot-bots* (see next section).

We investigated how to exploit *robot mobility*, *swarm spatial distribution*, and *multi-hop wireless communications*, to let the robots in the swarm: (i) implement a *distributed and cooperative sensing* of hand gestures, and (ii) robustly reach a *consensus* about a gesture.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2. THE ROBOTS IN THE SWARM

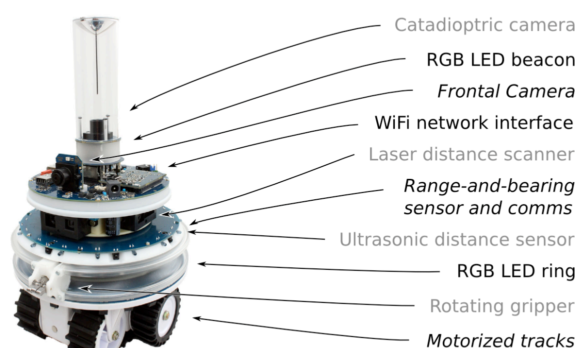


Figure 1: The foot-bot mobile platform. Italicized text indicates features we use; remaining features are either for monitoring (black), or are not used (gray).

We use *foot-bot* robots (Figure 1), developed in the *Swarmanoid* project [1] specifically for swarm robotics applications. The foot-bot is based on an on-board ARM 11 533MHz with a Linux-based operating environment.

We use a subset of the sensors and actuators available on such platform. In particular, the *frontal camera* is used for recognizing gestures, and acquires 512×384 RGB images. *Motorized track-based wheels* allow robots to move at speeds up to 5cm per second. The infrared-based *range-and-bearing* sensor and actuator allows a robot to detect its line-of-sight neighbors up to a range of few meters, and to recover their distance and bearing; messages can be broadcast to neighbors through a low-bandwidth (100 bytes/sec), low-reliability communication channel; in our implementation, all messages propagate to the swarm using multi-hop communication on this system. *RGB LEDs* are used to display the state of the system and for notifying the user about the decision the swarm has taken.

3. GESTURE RECOGNITION AND DISTRIBUTED CONSENSUS

We consider the two sets of gestures represented in Figure 2, namely finger counts (from 0 to 5) and four ad-hoc gestures representing furniture-like shapes, designed for interacting with *roombot* robots [2].

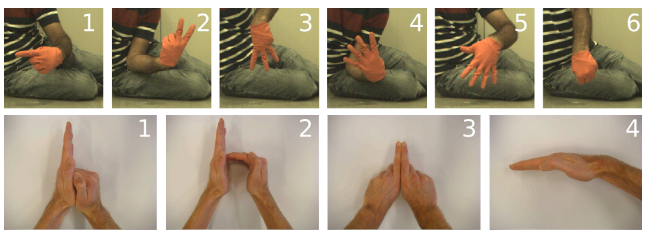


Figure 2: The six finger-count gestures (first row), and the four furniture-like gestures (second row).

3.1 Training of the Gesture Classifier

The first step was to use 13 foot-bots to collect a total of 70,000 hand gesture images from 65 different points of view. Figure 3 shows the acquisition setup.

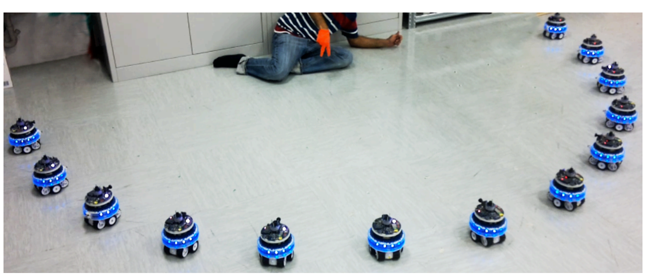


Figure 3: Setup for the training dataset acquisition.

For each acquired image, robots use color-based segmentation to detect the glove and obtain a binary mask, from which 20 shape features are computed.

With this data set we trained a *Support Vector Machine*, which is used by the robots for *individual gesture classification* and generation of an *opinion vector*, assigning a probability to each known gesture [3]. The resulting classifier performs correctly independently on the orientation of the hand since it was trained from images obtained from different points of view.

3.2 Distributed Gesture Recognition

In our scenario, robots search for the glove in an environment. When a robot detects the glove, the rest of the robots *moves* to adapt their viewpoint for better sensing.

Robots then start acquiring hand images at a rate of roughly one per second. Immediately after each acquisition, the image is processed as described above. The resulting opinions (i.e., classification vectors representing the probability of each gesture) are spread throughout the swarm through *multi-hop message relaying*. Each robot records its own opinions (deriving from successive acquisitions) as well as opinions received from the rest of the robots in the swarm.

The full set of available opinions (which may be conflicting) are additively taken into account by each robot, that incrementally builds a decision vector $\mathbf{D}$ as the component-wise *sum* of all the classification vectors (opinions) it has locally generated and/or received from other robots. $\mathbf{D}$'s component with the highest value, i', indicates the gesture class in favor of which most evidence is available at the moment to the robot. The robot also calculates a measure of its *confidence* about the true class being i' as $\lambda = \mathbf{D}_{i'} - \mathbf{D}_{i''}$, where i'' is the index of the second highest component of $\mathbf{D}$.

When a robot has gathered enough evidence, i.e., when λ exceeds a predefined threshold, it takes a *decision* for the hand gesture, and sends it into the swarm, where it is propagated through wireless line-of-sight multi-hop communication. Robots receiving a decision immediately adopt it. If different decisions are generated in a swarm, the one assessed with the highest confidence overrides and shutdowns the propagation of the others.

The linked video shows two examples of command *execution*. For the furniture-like shapes, robots send their decisions to the simulated *Roombot* system, where modular robots 'build' the furniture [2]. For the finger count, after receiving a 'two', the swarm splits in two groups moving in opposite directions.

4. DEMONSTRATION

The system will be demonstrated with a swarm composed by at least 5 foot-bots, as shown in the video. The full scenario described above will be operational and the participants will be able to interact with the swarm (one person at a time) using an orange glove.

5. ACKNOWLEDGMENTS

This research was supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Research (NCCR) Robotics (`www.nccr-robotics.ch`). The Authors participate to Project 4: *Distributed Robotics*, sub-project 4.5, *Symbiotic human-swarm cooperation*.

6. REFERENCES

[1] *Swarmanoid: Towards humanoid robotic swarms.* FET-OPEN project funded by the European Commission, `http://www.swarmanoid.org`.

[2] *Roombots: Modular robotics for adaptive and self-organizing furniture.* `http://biorob.epfl.ch/roombots`.

[3] J. Nagi et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Proc. of IEEE ICSIPA*, Kuala Lumpur, Malaysia, 2011.

Team-It: Location-Based Mobile Games for Multi-Agent Coordination and Negotiation (Demonstration)

Spencer Frazier
Alex Newnan
Univ. of Southern California
4676 Admiralty Way #1001
Marina Del Rey, CA, USA.
sfrazier@usc.edu
newnan@usc.edu

Yu-Han Chang
Rajiv Maheswaran
Univ. of Southern California
4676 Admiralty Way #1001
Marina Del Rey, CA, USA.
ychang@isi.edu
maheswar@usc.edu

Categories and Subject Descriptors

I.1.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms,Experimentation

Keywords

Multi-Agent Systems, Location-based Games, Coordination, Negotiation, Human-Agent Interaction

1. INTRODUCTION

The multi-agent systems community has made great strides investigating issues such as coordination and negotiation. When addressing human or human-agent behavior, very few approaches have addressed a feature that people are embodied in the real-world and act in geospatial environments. In the past, it has been difficult to perform experiments and collect data for such domains. However, with the spread of mobile technology that can run sophisticated applications and return location-based data, we are now in a position to investigate such questions.

TEAM-IT allows researchers to run mobile-games for a variety of location-based experiments for multi-agent coordination and negotiation with real-world movement as well as competitive experiments such as pursuit-evasion games. We will provide a brief description of TEAM-IT , its capabalities, its applications and our plan for the demonstration.

2. TEAM-IT

TEAM-IT allows for multiple teams each which can be composed of one or more players. Players can be human or software agents. Human players interact with the game through an iOS interface shown in Figures below. All players have a location which is tracked and shared through the interface. If players move, their locations are tracked and updated. Human players are currently being tracked via GPS location updates. Figure 1 shows various human and software agents on the game interface.

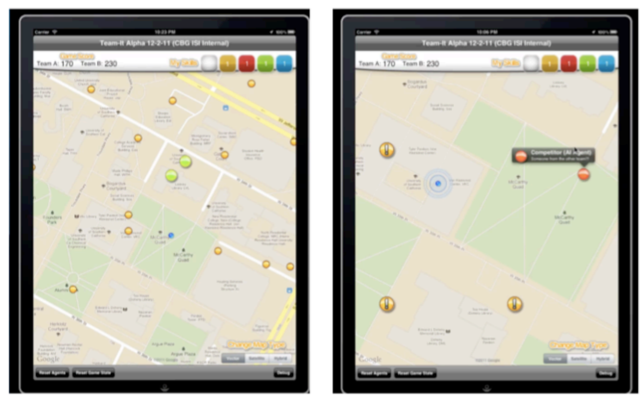


Figure 1: Human agents (larger green circles in the left figure) and software agents (red circles in the right figure) are shown in a map-based interface.

Each player is give a a collection of *cards* with various colors. These cards are abstractions that can represent skills or resources at their disposal. Players can have multiple cards of the same color to indicate greater skill or more resources. Cards can be persistent indicating that they are skill that can be applied repeatedly or consumable indicating they are a resource that gets used up when applied.

The game also has multiple *task collections*, each of which are situated in a specific geospatial location. These collections are only discovered if players are within a certain discovery radius of the task collection location. When discovered, a collection reveals a set of *tasks*. Each task requires a certain set of cards to be applied simultaneously for a stated duration in order to be completed successfully. The cards can be applied to a task only when an agent is within a particular application radius from the collection location. This application radius can be smaller than the discovery radius. Multiple players can apply cards to the same task.

Once all the required cards are applied for the required skills for the required duration, the task is completed. If a player leaves the application radius or cancels the application of a card, the task is incomplete and must be started again in order to be completed, i.e., tasks cannot be suspended. When a task is completed, each team receives

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

Figure 2: Players can see tasks as yellow circles that show an exclamation point when the player is within the discovery radius (left figure). At the top right, players can see their card sets. By pressing the task collection icon, players go to the card application interface (right figure) where the can apply cards to a chosen task.

points which can differ by team. The points received are known when the task collection is discovered. Figure 2 shows tasks in the map interface as well as the card application interface.

Human players may not even as a team have the cards required to complete tasks. Team-It also has software agent which are parts of other teams with whom players can negotiate to obtain cards. The negotiation only involves consumable cards, i.e., resources, that can change ownership. Software agents can be endowed with arbitrary negotiation algorithms and policies can be as heterogeneous. The policies are generated by hidden valuations over various cards. The interface for negotiation is shown in Figure 3.

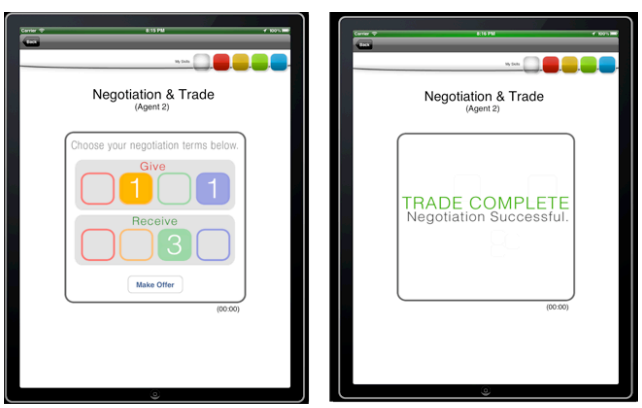


Figure 3: The human players can see what card types the software agent is willing to offer and accept as well as options for negotiation (left figure). A successful trade is also shown (right figure).

3. APPLICATIONS OF TEAM-IT

Team-It gives researchers the ability to run real-world human-agent geospatial experiments in a variety of multi-agent contexts. We can investigate multi-agent coordination by focusing on games with single team and only persistent cards where the team must decide how to best combine, path plan and schedule discovery and application of diverse skills to optimize their performance within a given time interval. We can add competition or motivational diversity with multiple teams where players must negotiate to get skills from other teams to accomplish their goals. We can introduce negotiation and trading with resources in a mixed-initiative environment. Team-It also enables experiments in pursuit-evasion games with the location identification, discovery and card application features.

4. DEMONSTRATION

Figure 4: A Team-It game on the USC campus showing players coordinating to complete tasks.

We will bring several iPads on which participants will be able to use Team-It . We will adapt the location-based services to work indoors by using one of two features: step-based tracking using the accelerometer where participants can physically move or acceleromater-controlled motion where participants move by manipulating the iPad. We will choose the best option based on the location and structure of the demo facility. The Team-It demo will involve participants joining teams and working with their team, other teams and software agents to complete tasks and gain points. We will display a live scoreboard of the best performing teams. Participants will be able to join and leave the game at any time. A movie of Team-It can be found at: *http://youtube.com/*.

5. REFERENCES

GaTAC: A Scalable and Realistic Testbed for Multiagent Decision Making (Demonstration)

Ekhlas Sonu and Prashant Doshi
Dept. of Computer Science
University of Georgia
Athens, GA,30602, USA
esonu@uga.edu, pdoshi@cs.uga.edu

ABSTRACT

In an attempt to bridge the gap between the theoretical advances in multiagent decision making algorithms and their application in real world scenario, we present the *Georgia testbed for autonomous control of vehicles (GaTAC)*. GaTAC provides a low-cost, open-source and flexible environment for realistically simulating and evaluating policies generated by multi-agent decision making algorithms in real world problem domains pertaining to control of autonomous uninhabbited aerial vehicles (AUAVs). We describe GaTAC in detail and shall demonstrate how GaTAC could be used to simulate an example AUAV problem. We expect GaTAC to facilitate the development and evaluation of scalable decision making algorithms with results that have immediate practical implications.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

scalability, testbed, autonomous vehicles

1. INTRODUCTION

With advances in multi-agent sequential decision making algorithms, there is a need to move beyond traditional toy problems to scalable problems with real world implications that may be used to evaluate performance of various decision making algorithms. We think that desired problem domains should, (a) be scalable to naturally allow for greater numbers of physical states, actions, observations, and agents while maintaining the plausibility of the problem; (b) be flexible to accomodate different types of multi-agent settings such as co-operative, competitive or mixed; (c) produce solutions that are rich in structure and which have practical implications; and (d) be realistic and have popular appeal. In this paper, we introduce a problem domain that meets these criteria.

Unmanned agents such as uninhabited aerial vehicles (UAVs) are used in fighting forest fires, law enforcement, and wartime reconnaissance. They operate in environments characterized by multiple parameters that affect their decisions, including other agents with common or antagonistic preference. The task is further complicated as the vehicles may possess noisy sensors and unreliable actuators. In such complex and unreliable settings, an autonomous UAV must choose navigational and surveillance actions that are expected to optimize its objective of say, timely reconnaissance of target while avoiding detection. UAV operation theaters may be populated by a single reconnaissance target or a host of other agents including UAVs working together as a team, or other hostile UAVs. Depending on the type of the agents present in the environment, this would involve application of decision-theoretic frameworks such as interactive POMDPs [2] and decentralized POMDPs [1].

In order to facilitate application of multiagent decision making to the problem domains pertaining to AUAVs and its evaluation, we have developed the *Georgia testbed for autonomous control of vehicles (GaTAC)*. GaTAC is a computer simulation framework for evaluating autonomous control of aerial robotic vehicles such as UAVs. It provides a low-cost and open-source alternative to highly complex and expensive simulation architecture. GaTAC uses a free, open-source and multi-platform flight simulator software called *FlightGear*. GaTAC deploys multiple instances of the flight simulator on a networked cluster of computing platforms using a scalable architecture. It is flexible in allowing the interchange of instances of manually controlled vehicles with autonomous ones. It can be extended to include complex scenarios involving multiple UAVs performing complex tasks.

In this paper we describe GaTAC in detail focusing on its architecture and its components and provide an introduction to our demonstration of its applicability on a simple example problem.

2. TESTBED FOR AUTONOMOUS CONTROL

As we mentioned previously, the objective behind the development of GaTAC is to provide a realistic and scalable testbed for algorithms on multiagent decision making. GaTAC facilitates this by providing an intuitive and easy to deploy architecture that makes use of powerful, open-source software components. Successful demonstrations of algorithms in GaTAC would not only represent tangible gains but also have the potential for practical applications toward designing autonomous UAVs. We think that multiagent decision making could make significant contributions in this area.

2.1 Architecture

We show a simplified design of the GaTAC architecture in Fig. 1, where a manually controlled UAV is interacting with an autonomous one. Briefly, GaTAC employs multiple instances of an open-source flight simulator possibly on different networked platforms that communicate with each other via external servers, and an autonomous control module that interacts with the simulator instances using a communication module. GaTAC can be deployed on most platforms including Linux and Windows with moderate hardware requirements, and the entire source code is available. GaTAC is implemented using C++ programming language.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

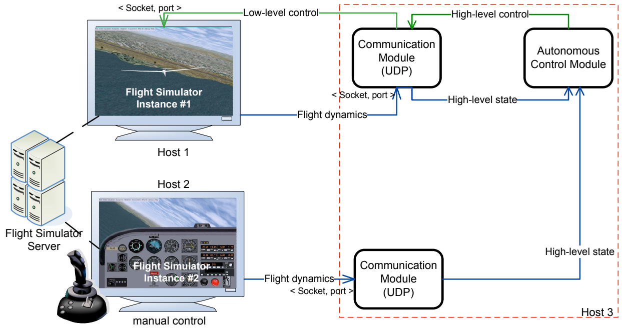


Figure 1: Design of GaTAC showing two networked instances of a flight simulator (FlightGear with 3D scenery from TerraGear), one autonomously and other manually controlled. GaTAC is extensible and more instances may be added.

Each agent may be simulated on a separate instance of GaTAC that may be running on different computing platforms connected through the internet. GaTAC doesn't apply any limit to the problem size or the number of agents. We describe the individual components of a GaTAC instance next.

2.1.1 Flight Simulator

We utilize *FlightGear* [3] as the flight simulator in GaTAC. FlightGear flight simulator project is an open-source, multi-platform, hyperrealistic flight simulator with a goal to develop a low cost sophisticated flight simulator for use in academic and research environments. The entire source code of FlightGear written in C++ is available under GNU General Public License, allowing full extensibility. It provides a flexible platform with options to choose from multiple aircrafts, including UAVs (e.g., Predator), which could be operated manually or guided automatically by external programs. FlightGear uses a generic, six degrees-of-freedom flight dynamics model for simulating the motion of aerial vehicles. It simulates the effect of airflow on different part of the aircraft making it possible to perform the simulation based on geometry and mass information combined with more commonly available performance numbers for an aircraft. FlightGear utilizes realistic 3-dimensional scenery available from TerraGear, which virtually maps many parts of the world including models of the sky.

FlightGear also provides multiple views of the flying aircraft, including external views from different viewpoints and an internal cockpit view which allows for a realistic flying experience. Finally, multiple instances of FlightGear may be run on different hosts and are linked together through external servers located in different countries. This multi-player mode allows for multiple aircrafts to fly simultaneously and see each other if the aircrafts are in visual range. This is a crucial functionality for its use in multiagent systems research.

2.1.2 Communications Module

FlightGear allows remote control of the aircrafts through UDP socket based communication channels. The communication module in GaTAC (see Fig. 1) establishes UDP sockets that are used to communicate with instances of FlightGear. Control data at a low level is sent to FlightGear in order to remotely pilot the UAV. This data includes values for more than 30 flight parameters including the throttle, rudder, elevator and aileron settings. The communications module receives the aircraft's flight dynamics in real time from FlightGear. This includes data about the current latitude, longitude and altitude location of the aircraft, the values of the different flight surfaces, and current fuel level. During flight, the communication module continuously sends and receives data from the FlightGear instance at a pre-specified baud rate. GaTAC associates a communication module with every instance of FlightGear regardless of whether the corresponding aircraft is autonomously or manually controlled. If the aircraft is manually controlled, the communication module simply receives the flight dynamics of the aircraft in order to remain informed about the state of that aircraft. The communication module also provides a way for UAVs to communicate with each other. This may be useful in team settings with communication.

2.1.3 Autonomous Control Module

In order to allow algorithmic control of the aircraft, GaTAC implements an autonomous control module (see Fig. 1). This module implements low-level control actions such as setting values of various flight parameters including throttle, rudder, elevator and aileron settings. Using these low level actions, we have constructed high-level control actions such as *takeoff*, *fly straight*, *change heading*, move to an adjacent grid, etc. We may utilize these actions to construct a set of agent actions for any decision making problem. The GaTAC library is extensible to include additional actions. Additionally, GaTAC allows users to define their own grids of any size.

Because we intend to utilize GaTAC with multiagent decision making frameworks, it implements methods that read *policy tree* files in different formats generated by various algorithms. We have made effort to make GaTAC independent of any particular type of decision-theoretic framework. It may be easily integrated with existing implementations by simply providing it with the behavioral policies generated by the various algorithms for decision making.

3. DEMO

In demo, we show how GaTAC may be used to simulate and evaluate the policies obtained for a few example UAV reconnaissance problem using various decision making algorithms. The evaluation criteria may differ according to various problem. Some of the evaluation criteria may be the number of successful achievement of goal (number of successful reconnaissance), the cumulative reward obtained, etc.

More information on GaTAC including the source code, a demo video and an informative powerpoint presentation may be obtained from the following link: http://thinc.cs.uga.edu/thinclabwiki/index.php/GaTAC_:_Georgia_Testbed_for_Autonomous_Control_of_Vehicles

4. DISCUSSION

GaTAC provides a low cost, open source scalable platform for a satisfactory simulatory experience of a problem domain that has popular appeal, and is extensible. GaTAC represents a realistic testbed for multiagent decision making research, and a first step in our knowledge toward enabling decision-making algorithms to cross over to domains of practical import. We hope that GaTAC could be further improved with inputs from users.

Acknowledgement

This research is supported in part by AFOSR through grant no. #FA9550-08-1-0429 and in part by a NSF CAREER award with grant no. #IIS-0845036.

5. REFERENCES

[1] 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.

[2] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*. 24:49–79, 2005.

[3] A. R. Perry. The flightgear flight simulator. In UseLinux, 2004.

Author Index