Convex Coverage Set Methods for Multi-Objective Collaborative Decision Making

(Doctoral Consortium)

Diederik M. Roijers*
Institute for Informatics
University of Amsterdam
The Netherlands
d.m.roijers@uva.nl

ABSTRACT

My research is aimed at finding efficient coordination methods for multi-objective collaborative multi-agent decision theoretic planning. Key to coordinating efficiently in these settings is exploiting loose couplings between agents. We proposed two algorithms for the case in which the agents need to make a single collective decision: *convex multi-objective variable elimination (CMOVE)* and *variable elimination linear support (VELS)*. While CMOVE deals with the multiple objectives on a single agent level, VELS deals with the multiple objectives on a joint decision level, leading to different trade-offs. We also proposed a naive way (*approximate optimistic linear support (AOLS)*) to apply the scheme of VELS to sequential settings, which does not yet fully exploit loose couplings. The next step in this line of research is to extend the distributed single-shot methods to distributed sequential settings, while better exploiting loose couplings.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: multi-agent systems

General Terms

Algorithms

Keywords

Multiple objectives, Coordination graphs

1. INTRODUCTION

In cooperative multi-agent decision problems, agents work together to maximize a common utility. Many real-world problems, such as computer networks [1, 10] and traffic network maintenance planning [4] are naturally expressed as cooperative multi-agent problems. Key to coordinating in these domains efficiently, is exploiting *loose couplings*; each agent's behavior directly affects only a subset of the other agents. Graphical models, in multi-agent literature referred to as *coordination graphs*, can capture such independence and can be used by methods such as *variable elimination* to exploit the independence [2, 3]. An example coordination graphs is shown in Figure 1, in which there are three agents (circles), that connect to other agents via *local payoff functions* (squares). Each local payoff function **u**, takes the actions of the directly connected agents (via lines) as input.

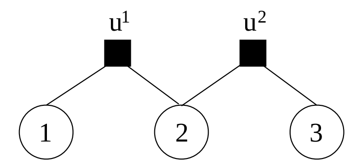

Figure 1: A coordination graph

My research focusses on cooperative loosely coupled multi-agent settings in which there are multiple objectives. This is important, because many real-world problems are inherently multi-objective [5]. For example, consider a computer network in which messages need to be passed on (as in [1]), where we both care about maximizing a computer network's performance and minimizing power consumption (as in [10]). Or, consider a traffic network in which several contractors need to perform maintenance, where we care both about the costs of doing the maintenance and about the hinder for traffic [4].

Even though a problem may have multiple objectives, special solution methods are sometimes not required. If the problem can be *scalarized*, i.e. the vector-valued utility function transformed to a scalar function, the problem may be solvable with existing single-objective methods. However, when the *weights* of the *scalarization function* cannot be known in advance or difficult to quantify, this approach does not apply. For example, consider a company that produces different metals whose market prices vary. If there is not enough time to re-plan for each price change, we need explicit multi-objective methods. Such methods compute a *set of solutions* that contain an optimal solution for all possible scalarizations.

2. THE SINGLE SHOT SETTING

For the setting in which agents need to take a single joint action, we model the problem as an extension to coordination graphs, we refer to as *multi-objective coordination graphs* (MO-CoGs) [6, 8].[1]

*This research is supported by the NWO DTC-NCAP (#612.001.109) and NWO CATCH (#640.005.003) projects.

[1]We previously used the term *multi-objective collaborative*

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

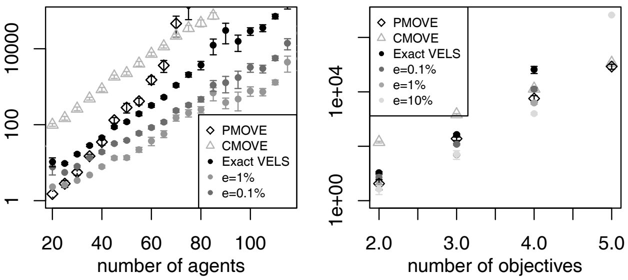

Figure 2: The runtimes of PMOVE, CMOVE and VELS, for varying numbers of agents (and 2 objectives) (left) and objectives (and 25 agents) (right).

Several methods already existed for MO-CoGs. For instance, Rollón and Larrosa [9] introduce an algorithm that we refer to as *Pareto multi-objective variable elimination* (PMOVE), which solves MO-CoGs by iteratively solving local subproblems to eliminate agents from the graph. However, because these existing methods all compute the *Pareto coverage set (PCS)* (i.e., the Pareto front), which grows very rapidly with the number of agents, they do not scale well [6].

Fortunately, computing the PCS is often not required. In the highly prevalent case where we know a *linear scalarization function* (with unknown weights) will be applied, the *convex coverage set (CCS)* suffices. Also, when the joint strategies can be *stochastic* all optimal value-vectors can be attained by mixing strategies from the CCS [11]. In the case of linear scalarization, the scalarized value becomes a *piecewise linear and convex (PWLC)* function in the scalarization weights. PWLC value functions are well-known from POMDP literature, which enables us to use insights from POMDPs on multi-objective decision problems.

We first extended PMOVE, to compute the CCS. The resulting algorithm we call *convex MOVE (CMOVE)* [6]. We do this by using POMDP pruners to compute local CCSs as the solutions to the local subproblems of PMOVE, rather than local PCSs. We show, both theoretically and empirically, that CMOVE is much more efficient than PMOVE (Figure 2), accross all numbers of objectives. However, the CCS can still grow rapidly with the number of agents.

To address this difficulty, we proposed *variable elimination linear support (VELS)* [8], again using POMDP insights, in order to build the CCS incrementally. We show that for moderate numbers of objectives, its complexity is better than that of previous methods (including CMOVE), and also that VELS is empircally much faster than existing methods (Figure 2 (left)). However, that is not the only advantage of VELS: because VELS builds up the CCS incrementally, the intermediate solutions computed by VELS can serve as approximations to the CCS, and we can compute an upper bound on the maximum error ε in scalarized payoff. We show emprically (Figure 2) that VELS can compute an ε-CCS much faster than the full CCS, even for small ε.

3. THE SEQUENTIAL SETTING

A broader, more expressive setting is when we consider sequential decision making. In this setting, we have to deal with a state that can change on the basis of the actions of the agents, which then affects the next state. Such *multi-objective (loosely coupled) multi-agent markov decision processes* (MOMMDPs) are difficult, because even though the reward functions might be loosely coupled and expressable with a MO-CoG, agents that are far away from each other might still influence each over time via the state. For example, in a traffic network, agents that work on local intersections may still influence far away agents via their effects on the cars that pass through the entire network.

We applied a naive approach in [4], by centralizing the decisions, resulting in a large MOMDP [5]. For this method we extended the outer loop of VELS, to be able to use approximate planning methods, yielding *approximate optimistic linear support (AOLS)*, and applied existing MDP planning methods as a subroutine. However, this approach does not scale, because the centralized joint actions grow exponentially in the number of agents.

In future research, we will try to develop an efficient planning method by better exploiting the loosely couplings. First, we will attempt to develop an ε-approximate planning version of sparse-cooperative Q-learning [3]. In general, this may not possible, as the effects of an agent on other agents via the state is unboundable in general. Therefore, we hope to identify a broadly applicable subclass of MOMMDPs for which an ε-approximate planning method yields substantial speed-ups with respect to exact planning methods.

Then we will attempt to extend the ε-approximate planning method to the multi-objective setting. One option is to use this method as a subroutine in AOLS. Alternatively, we can see an MOMMDP as a series of MO-CoGs, and use either CMOVE or VELS to replace the max operator in the Bellman equation.

graphical game for this setting [7].

4. REFERENCES

[1] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *NIPS*, pages 671–678, 1993.

[2] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *NIPS*, 2002.

[3] J. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *J. Mach. Learn. Res.*, 7:1789–1828, Dec. 2006.

[4] D. M. Roijers, J. Scharpff, M. T. Spaan, F. A. Oliehoek, M. de Weerdt, and S. Whiteson. Bounded approximations for linear multi-objective planning under uncertainty. In *ICAPS*, 2014. Yet to appear.

[5] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 47:67–113, 2013.

[6] D. M. Roijers, S. Whiteson, and F. A. Oliehoek. Computing convex coverage sets for multi-objective coordination graphs. In *ADT*, Nov. 2013. To appear.

[7] D. M. Roijers, S. Whiteson, and F. A. Oliehoek. Multi-objective variable elimination for collaborative graphical games. In *AAMAS*, 2013. Extended Abstract.

[8] D. M. Roijers, S. Whiteson, and F. A. Oliehoek. Linear support for multi-objective coordination graphs. In *AAMAS*, 2014. Yet to appear.

[9] E. Rollón and J. Larrosa. Bucket elimination for multiobjective optimization problems. *Journal of Heuristics*, 12:307–328, 2006.

[10] G. Tesauro, R. Das, H. Chan, J. O. Kephart, C. Lefurgy, D. W. Levine, and F. Rawson. Managing power consumption and performance of computing systems using reinforcement learning. In *NIPS*, 2007.

[11] P. Vamplew, R. Dazeley, E. Barker, and A. Kelarev. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In *Advances in Artificial Intelligence*, pages 340–349. 2009.