Efficient Methods for Multi-Objective Decision-Theoretic Planning

Diederik M. Roijers
Institute for Informatics
University of Amsterdam
The Netherlands

1 Introduction

In decision-theoretic planning problems, such as (partially observable) Markov decision problems [Wiering and Van Otterlo, 2012] or coordination graphs [Guestrin *et al.*, 2002], agents typically aim to optimize a scalar value function. However, in many real-world problems agents are faced with multiple possibly conflicting objectives, e.g., maximizing the economic benefits of timber harvesting while minimizing ecological damage in a forest management scenario [Bone and Dragicevic, 2009]. In such multi-objective problems, the value is a vector rather than a scalar [Roijers *et al.*, 2013a].

Even when there are multiple objectives, it might not be necessary to have specialized multi-objective methods. When the problem can be *scalarized*, i.e., converted to a single-objective problem before planning, existing single-objective methods may apply. Unfortunately, such a priori scalarization is not possible when the *scalarization weights*, i.e., the parameters of the scalarization, are not known in advance. For example, consider a company that mines different metals whose market prices vary. If there is not enough time to re-solve the decision problem for each price change, we need specialized multi-objective methods that compute a *coverage set*, i.e., a set of solutions optimal for all scalarizations. What constitutes a coverage set depends on the type scalarization.

Much existing research assumes the *Pareto coverage set (PCS)*, or Pareto front, as the optimal solution set. However, we argue that this is not always the best choice. In the highly prevalent case when the objectives will be *linearly weighted*, the *convex coverage set (CCS)* suffices. Because CCSs are typically much smaller, and have exploitable mathematical properties, CCSs are often much cheaper to compute than PCSs. Futhermore, when policies can be *stochastic*, all optimal value-vectors can be attained by mixing policies from the CCS [Vamplew *et al.*, 2009]. Thefore, this project focuses on finding planning methods that compute the CCS.

2 Computing the CCS

A CCS is a set of policies that is optimal for each possible weight vector $\mathbf{w}$ of a linear scalarization function, i.e., when the set of all possible policies is Π, the CCS is a subset of Π such that $\forall \mathbf{w} \max_{\pi \in \Pi} \mathbf{w} \cdot \mathbf{V}^{\pi} = \max_{\pi' \in CCS} \mathbf{w} \cdot \mathbf{V}^{\pi'}$, where $\mathbf{V}^{\pi}$ denotes the multi-objective value of a policy π. The CCS is a sufficient set to identify the so-called *scalarized value function*, i.e., the function that gives the maximal scalarized value for each $\mathbf{w}$: $V^*(\mathbf{w}) = max_{\pi \in CCS} \mathbf{w} \cdot \mathbf{V}^{\pi}$. $V^*(\mathbf{w})$ is a *piecewise-linear and convex (PWLC)* function in the scalarization weights. Finding $V^*(\mathbf{w})$, and thus the CCS, is solving the multi-objective decision problem.

In this research we distinguish two approaches to computing the CCS. In the *inner loop* approach we solve a multi-objective decision problem as a series of simpler/smaller multi-objective problems. In the *outer loop* approach we solve a multi-objective decision problem as a series of single-objective problems. Specifically, we propose an outer loop scheme called *optimistic linear support (OLS)*, that calls a single-objective solver as a subroutine to solve a finite series of *scalarized* problem instances to produce the CCS.

While inner loop methods are typically faster for large numbers of objectives, OLS typically scales better in the size of the problem (e.g., the number of agents in a multi-objective coordination graph) and can use any single-objective solver as a subroutine. An important advantage of this is that an improvement to the single-objective state-of-the-art directly applies to the multi-objective case.

3 Optimistic Linear Support

Our outer loop method is called *optimistic linear support (OLS)* [Roijers *et al.*, 2014b]. OLS finds the CCS by solving a series of scalarized problems. For each scalarized problem, OLS calls a single-objective solver to find the optimal policy. OLS retrieves the multi-objective value of this policy and adds it to a partial CCS. Such a partial CCS induces an approximation to $V^*(\mathbf{w})$ which is also a PWLC function.

OLS uses a priority queue to make smart choices about which scalarized problem instances to solve. In particular, OLS selects so-called *corner weights* that lie at the intersections of line segments of the approximate scalarized value function resulting from a partial CCS. The priority of each corner weight is the maximal possible improvement that can result from finding a new multi-objective value-vector for this $\mathbf{w}$, which can be calculated using a linear program.

Due to a theorem by Cheng [Cheng, 1988] we know that highest maximal possible improvement is at one of the corner weights. Therefore, if we have checked all corner weights and have not found an improvement, we can stop. When the single-objective solver that OLS calls is exact, OLS is guaranteed to find the exact CCS within a finite number of calls to

this single-objective solver. When the single-objective solver is ε-approximate, OLS finds an ε-CCS [Roijers *et al.*, 2014a].

When there is not enough time to let OLS converge, the approximate CCS can be used as a bounded approximation to the full CCS. I.e., the maximal possible improvement of the corner weight that is the head of the priority queue is a bound on the quality of the approximate CCS.

OLS is typically faster than inner loop methods for problems with small numbers of objectives. For two and three objective problems, the number of times OLS needs to call the single-objective solver is linear in the size of the CCS. Because many real-world problems have only a small number of objectives, OLS is often preferable.

4 Multi-Objective Decision Problems

We investigated different multi-objective decision problems. In particular, *multi-objective coordination graphs (MO-CoGs)*, (multi-agent) *multi-objective Markov decision processes (MOMDPs)*, and *multi-objective partially observable Markov decision processes (MOPOMDPs)*.

In MO-CoGs, a team of agents needs to perform a single joint action to optimize the team value. For this problem we created an inner loop method called *convex multi-objective variable elimination (CMOVE)* [Roijers *et al.*, 2013b]. This method follows the same scheme as single-objective *variable elimination (VE)*, i.e., it solves a series of local subproblems that follows from eliminating agents from the coordination graph. However, rather than a single optimal local action, CMOVE computes a local CCS for each local subproblem. We compared this to our outer loop method, called *variable elimination linear support (VELS)* [Roijers *et al.*, 2014b], which combines OLS and VE. Futhermore, we also created memory-efficient inner and outer loop methods based on AND/OR tree search [Roijers *et al.*, 2015b]. The experiments on MO-CoGs indicate that the inner loop method, CMOVE, scales better in the number of objectives, while the outer loop method, VELS, scales better in the number of agents and can compute an ε-CCS, leading to large additional speedups. Furthermore, VELS is more memory-efficient than CMOVE. In fact, VELS uses little more memory than VE. When memory is very restricted and VELS cannot be applied, the memory-efficient outer loop method provides an alternative. Although it is considerably slower than VELS, some of this loss can be compensated by allowing some error (ε).

In MOMDPs, we combined OLS with the exact single-objective solver SPUDD and the approximate single-objective solver UCT* and tested it on a complex planning problem with a very high number of states called the *maintenance planning problem* [Roijers *et al.*, 2014a]. We show experimentally that good approximations to the CCS can be found, even when we allow relatively little time for UCT*.

In MOPOMDPs [Roijers *et al.*, 2015a], we improve upon OLS by *reusing* policies and values found for earlier scalarized problem instances in calls to the single-objective solver later in the sequence, drastically improving computation time.

In future research, we will try to develop efficient multi-objective planning algorithms for multi-agent sequential settings. We aim to find an ε-approximate CCS planning method for fully obserable multi-agent MDPs. In order to achieve this, we would first need to find an ε-approximate single objective planning method that exploits sparse interactions between agents in multi-agent MDPs. Then, we can extend this to the multi-objective case using both the inner and the outer loop approach, and compare the resulting methods.

Next to fully observable settings, we aim to find multi-objective planning methods for *decentralized* problems such as Dec-POMDPs. In these problems, agents receive only local observations about the state, making it harder to coordinate. For decentralized problems we aim to exploit recent insights that enable the use of POMDP methods in this setting [Oliehoek and Amato, 2014].

References

[Bone and Dragicevic, 2009] Christopher Bone and Suzana Dragicevic. GIS and intelligent agents for multiobjective natural resource allocation: A reinforcement learning approach. *Trans. in GIS*, 13(3):253–272, 2009.

[Cheng, 1988] Hsien-Te Cheng. *Algorithms for partially observable Markov decision processes*. PhD thesis, UBC, 1988.

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

[Oliehoek and Amato, 2014] Frans A. Oliehoek and Christopher Amato. Dec-POMDPs as non-observable MDPs. IAS technical report IAS-UVA-14-01, Amsterdam, The Netherlands, 2014.

[Roijers *et al.*, 2013a] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 47:67–113, 2013.

[Roijers *et al.*, 2013b] Diederik M. Roijers, Shimon Whiteson, and Frans A. Oliehoek. Computing convex coverage sets for multi-objective coordination graphs. In *ADT*, pages 309–323, 2013.

[Roijers *et al.*, 2014a] Diederik M. Roijers, Joris Scharpff, Matthijs T.J. Spaan, Frans A. Oliehoek, Mathijs de Weerdt, and Shimon Whiteson. Bounded approximations for linear multi-objective planning under uncertainty. In *ICAPS*, pages 262–270, 2014.

[Roijers *et al.*, 2014b] Diederik M. Roijers, Shimon Whiteson, and Frans A. Oliehoek. Linear support for multi-objective coordination graphs. In *AAMAS 2014: Proceedings of the Thirteenth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1297–1304, May 2014.

[Roijers *et al.*, 2015a] Diederik Roijers, Shimon Whiteson, and Frans Oliehoek. Point-based planning for multi-objective POMDPs. In *IJCAI 2015: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, July 2015. To appear.

[Roijers *et al.*, 2015b] Diederik M Roijers, Shimon Whiteson, and Frans A Oliehoek. Computing convex coverage sets for faster multi-objective coordination. *Journal of Artificial Intelligence Research*, 52:399–443, 2015.

[Vamplew *et al.*, 2009] 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.

[Wiering and Van Otterlo, 2012] Marco Wiering and Martijn Van Otterlo. Reinforcement learning: State-of-the-art. In *Adaptation, Learning, and Optimization*, volume 12. Springer, 2012.