Bounded Approximations for Linear Multi-Objective Planning under Uncertainty

(Extended Abstract)

Diederik M. Roijers \textsuperscript{a} \hspace{1cm} Joris Scharpff \textsuperscript{b} \hspace{1cm} Matthijs T.J. Spaan \textsuperscript{a}

Frans A. Oliehoek \textsuperscript{a} \hspace{1cm} Mathijs M. De Weerdt \textsuperscript{b} \hspace{1cm} Shimon Whiteson \textsuperscript{a}

\{d.m.roijers, f.a.oliehoek, s.a.whiteson\}@uva.nl
\textsuperscript{a} University of Amsterdam, The Netherlands

\{j.c.d.scharpff, m.t.j.spaan, m.m.deweerdt\}@tudelft.nl
\textsuperscript{b} Delft University of Technology, The Netherlands

Abstract

Planning under uncertainty poses a complex problem in which multiple objectives often need to be balanced. When dealing with multiple objectives, it is often assumed that the relative importance of the objectives is known a priori. However, in practice human decision makers often find it hard to specify such preferences exactly, and would prefer a decision support system that presents a range of possible alternatives. We propose two algorithms for computing these alternatives for the case of linearly weighted objectives. First, we propose an anytime method, approximate optimistic linear support (AOLS), that incrementally builds up a complete set of $\epsilon$-optimal plans, exploiting the piecewise-linear and convex shape of the value function. Second, we propose an approximate anytime method, scalarised sample incremental improvement (SSII), that employs weight sampling to focus on the most interesting regions in weight space, as suggested by a prior over preferences.

We show empirically that our methods are able to produce (near-)optimal alternative sets orders of magnitude faster than existing techniques, thereby demonstrating that our methods provide sensible approximations in stochastic multi-objective domains.

1 Introduction

Many real-world planning problems involve both uncertainty as well as multiple objectives. This type of problems is expressed naturally using the multi-objective Markov decision process (MOMDP) framework [4]. Following [4] we assume the existence of a scalarisation function, i.e. a function that translates multi-dimensional rewards into a scalar value. However, using such a function for planning requires complete knowledge of its parameters, or weights, beforehand. When such knowledge is not available, solving an MOMDP requires finding the set of optimal solutions for all possible weights.

In this paper\textsuperscript{1}, we consider only linear scalarisation functions. Therefore, it suffices to focus on the Convex Coverage Set of an MOMDP. Existing methods such as optimistic linear support (OLS) [5] exploit the value piecewise-linear convexity in the optimal value function over all weights, present when the scalarisation function is linear, to minimise the number of scalarised MDPs that need to be solved. However, OLS can only guarantee this when the scalarised MDPs are solved optimally and is therefore not directly applicable to large realistic planning problems.

2 Our Contributions

We propose new methods that rely on approximate MDP solving techniques to produce (near-)optimal CCSs. The first algorithm we propose is approximate optimistic linear support (AOLS) that, given an

\textsuperscript{1}This is an extended abstract of our paper [3] at ICAPS 2014.
\( \epsilon \)-bounded MDP approximation, is guaranteed to produce an \( \epsilon \)-approximate CCS. The second algorithm, scalarised sample-based iterative improvement (SSII), exploits available prior knowledge on the distribution of weights and concentrates its effort within such a prior.\(^2\) Although SSII can in practice produce a better approximate CCS over this prior, we cannot provide a bound on the CCS quality because SSII relies on sampling.

Both AOLS and SSII use an approximate single-objective solver as a subroutine. AOLS can use any solver. SSII requires an anytime method. In this paper we use UCT* [2] for both.

### 3 Evaluation

We performed experiments on instances of the maintenance planning problem [6], a 2-objective, probabilistic and numerical planning domain, and compared the optimal OLS method with our approximate AOLS and SSII. The optimal solutions have been computed using SPUDD [1] and approximations using the UCT* algorithm from PROST [2]. We compared the outcomes in terms of runtime, average CCS error \( \epsilon_{\text{exp}} \) and maximal CCS \( \epsilon_{\text{max}} \) error. The results are presented in Table 1 below.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Runtime</th>
<th>[CCS]</th>
<th>( [0, 1] )</th>
<th>( [0.5, 1] )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \epsilon_{\text{exp}} )</td>
<td>( \epsilon_{\text{max}} )</td>
</tr>
<tr>
<td>OLS + SPUDD</td>
<td>2390.819</td>
<td>9.250</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AOLS + UCT* 0.01s</td>
<td>8.612</td>
<td>3.389</td>
<td>0.701</td>
<td>325.354</td>
</tr>
<tr>
<td>AOLS + UCT* 1s</td>
<td>19.940</td>
<td>4.111</td>
<td>0.119</td>
<td>65.668</td>
</tr>
<tr>
<td>AOLS + UCT* 10s</td>
<td>65.478</td>
<td>4.528</td>
<td>0.084</td>
<td>56.439</td>
</tr>
<tr>
<td>AOLS + UCT* 20s</td>
<td>165.873</td>
<td>5.694</td>
<td>0.044</td>
<td>38.667</td>
</tr>
<tr>
<td>SSII 1s, no prior</td>
<td>18.795</td>
<td>4.306</td>
<td>0.118</td>
<td>70.244</td>
</tr>
<tr>
<td>SSII 10s, no prior</td>
<td>59.336</td>
<td>3.889</td>
<td>0.061</td>
<td>51.800</td>
</tr>
<tr>
<td>SSII 1s, prior</td>
<td>17.892</td>
<td>3.944</td>
<td>0.221</td>
<td>95.189</td>
</tr>
<tr>
<td>SSII 10s, prior</td>
<td>59.154</td>
<td>4.083</td>
<td>0.141</td>
<td>71.290</td>
</tr>
</tbody>
</table>

Table 1: Comparison of averaged performance of the algorithms presented in this paper for various parameters, shown for two regions of the scalarised reward space. Runtimes are in seconds, the expected error \( \epsilon_{\text{exp}} \) and maximum error \( \epsilon_{\text{max}} \) are relative to the optimum CCS and %OPT denotes the fraction of instances that were solved optimally.

From the table we can conclude that both AOLS and SSII are able to produce reasonable, and sometimes even optimal, solutions much faster than OLS. Also, SSII is competitive with AOLS without exploiting additional knowledge but when SSII uses the prior it produces a slightly better CCS within the targeted weight region \( w_1 \in [0.5, 1] \) and \( w_2 = 1 - w_1 \).

### References


\(^2\)Note that such a prior expresses some — but not complete — knowledge about the weights.