Optimistic Linear Support
and Multi-objective POMDPs

Diederik M. Roijers
Informatics Institute
University of Amsterdam

in collaboration with:
Shimon Whiteson and Frans Oliehoek

October 20, 2015
Multiple objectives

Maximize coverage while minimizing damage

Observation
• Multi-objective decision problems
• Convex coverage sets
• Optimistic Linear Support
• Approximate single-objective solvers
• Multi-objective POMDPs
• OLS for MOPOMDPs
• Scalarized Perseus
• $\alpha$-matrix reuse
• Experimental results
Do we need multi-objective models?

*Sutton’s Reward Hypothesis:* “All of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received *scalar* signal (reward).”

Source: http://rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html

- $V : \Pi \rightarrow \mathbb{R}$
- $V^\pi = E_\pi[\sum_t r_t]$
- $\pi^* = \max_\pi V^\pi$
Sutton’s Reward Hypothesis: “All of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward).”

Source: http://rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html

- $V : \Pi \rightarrow \mathbb{R}$
- $V^\pi = E_\pi[\sum_t r_t]$
- $\pi^* = \max_\pi V^\pi$
**Sutton’s Reward Hypothesis:** “All of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received *scalar* signal (reward).”

Source: http://rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html

- \( V : \Pi \rightarrow \mathbb{R} \)
- \( V^\pi = E_\pi[\sum_t r_t] \)
- \( \pi^* = \max_\pi V^\pi \)
Why Multi-Objective Decision Making?

- **The weak argument**: real-world problems are multi-objective!

\[ \mathbf{V} : \Pi \rightarrow \mathbb{R}^n \]

- Objection: why not just scalarize?

- Scalarization function projects multi-objective value to a scalar:

\[ V_w^\pi = f(V^\pi, w) \]

- Linear case:

\[ V_w^\pi = \sum_{i=1}^{n} w_i V_i^\pi = w \cdot V^\pi \]

- A priori prioritization of the objectives

- *The weak argument is necessary but not sufficient
Why Multi-Objective Decision Making?

- **The weak argument**: real-world problems are multi-objective!

\[ V : \Pi \rightarrow \mathbb{R}^n \]

- Objection: why not just *scalarize*?

- Scalarization function projects multi-objective value to a scalar:

\[ V_w^\pi = f(V^\pi, w) \]

- Linear case:

\[ V_w^\pi = \sum_{i=1}^{n} w_i V_{i}^\pi = w \cdot V^\pi \]

- A priori prioritization of the objectives

- *The weak argument is necessary but not sufficient*
Why Multi-Objective Decision Making?

- **The weak argument**: real-world problems are multi-objective!

  \[ V : \Pi \rightarrow \mathbb{R}^n \]

- Objection: why not just *scalarize*?

- Scalarization function projects multi-objective value to a scalar:

  \[ V_w^\pi = f(V^\pi, w) \]

- Linear case:

  \[ V_w^\pi = \sum_{i=1}^{n} w_i V_i^\pi = w \cdot V^\pi \]

- A priori prioritization of the objectives
  - *The weak argument is necessary but not sufficient*
Why Multi-Objective Decision Making?

- **The weak argument**: real-world problems are multi-objective!

\[ \mathbf{V} : \Pi \rightarrow \mathbb{R}^n \]

- Objection: why not just **scalarize**?

- Scalarization function projects multi-objective value to a scalar:

\[ V_w^\pi = f(\mathbf{V}^\pi, \mathbf{w}) \]

- Linear case:

\[ V_w^\pi = \sum_{i=1}^{n} w_i V_i^\pi = \mathbf{w} \cdot \mathbf{V}^\pi \]

- A priori prioritization of the objectives

- **The weak argument is necessary but not sufficient**
Why Multi-Objective Decision Making?

- The strong argument: a priori scalarization is sometimes impossible, infeasible, or undesirable

- Instead produce the coverage set of undominated solutions

- Three scenario’s
Motivating scenarios

MO decision problem → algorithm → solution set → single solution

weights

planning or learning phase

selection phase

execution phase

MO decision problem → algorithm → solution set → single solution

user selection

selection phase

execution phase

MO decision problem + weights → algorithm → single solution

planning or learning phase

execution phase
Utility-based approach

- Scalarization is *explicit* or *implicit*, but always happens

- Scalarization function: $V_w = f(V, w)$

- Choose the solution set by:
  - What do we know about $f$?
  - Stochastic policies allowed?
  - Non-stationary policies allowed?
Utility-based approach

- Scalarization is *explicit* or *implicit*, but always happens.

- Scalarization function: $V_w = f(V, w)$

- Choose the solution set by:
  - What do we know about $f$?
  - Stochastic policies allowed?
  - Non-stationary policies allowed?
Convex coverage set (CCS)

- Scalarization function: \( f(V^\pi, w) = w \cdot V^\pi \)
- Scalarized value function: \( V^*_{\text{CCS}}(w) = \max_{\pi} w \cdot V^\pi \)
- Piece-wise linear and convex (PWLC) function
## Problem Taxonomy

<table>
<thead>
<tr>
<th>Linear Scalarization</th>
<th>Deterministic</th>
<th>Stochastic</th>
<th>Deterministic</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Policy (Known Weights)</td>
<td>one deterministic stationary policy</td>
<td>one mixture policy of two or more deterministic stationary policies</td>
<td>Pareto coverage set of deterministic non-stationary policies</td>
<td>convex coverage set of deterministic stationary policies</td>
</tr>
<tr>
<td>Multiple Policies (Unknown Weights or Decision Support)</td>
<td>deterministic</td>
<td>stochastic</td>
<td>deterministic</td>
<td>stochastic</td>
</tr>
</tbody>
</table>

D.M. Roijers (UvA)
Outline

- Multi-objective decision problems
- Convex coverage sets

- Optimistic linear support
- Approximate single-objective solvers

- Multi-objective POMDPs
- OLS for MOPOMDPs
- Scalarized Perseus
  - $\alpha$-matrix reuse
  - Experimental results
Optimistic Linear Support (OLS)

Outer loop approach: series scalarized instances with different $w$
- Terminates after checking only a finite number of weights \( w \)
- Exact solutions if single-objective solver is exact
- Anytime
Optimistic Linear Support

- Terminates after checking only a finite number of weights $w$
- Exact solutions if single-objective solver is exact
- Anytime
- Terminates after checking only a finite number of weights $w$
- Exact solutions if single-objective solver is exact
- Anytime
- Terminates after checking only a finite number of weights $\mathbf{w}$
- Exact solutions if single-objective solver is exact
- Anytime
Optimistic linear support

- $\varepsilon$-approximate single-objective solver

- OLS produces an $\varepsilon$-CCS
Outline

• Multi-objective decision problems
• Convex coverage sets

• Optimistic linear support
• Approximate single-objective solvers

• Multi-objective POMDPs
• OLS for MOPOMDPs
• Scalarized Perseus
• $\alpha$-matrix reuse
• Experimental results
Multi-objective Partially Observable MDPs

- Multiple objectives
- Vector-valued policy values
- Set of all possibly optimal policies

\[ V_w^\pi = w \cdot V^\pi = w_1 V_{\text{coverage}}^\pi + w_2 V_{\text{damage}}^\pi \]
Multi-objective Partially Observable MDPs

- Multiple objectives
- Vector-valued policy values
- Set of all possibly optimal policies

\[ V_w^\pi = w \cdot V^\pi = w_1 V_{\text{coverage}}^\pi + w_2 V_{\text{damage}}^\pi \]
Approach

- Optimistic Linear Support
- Opens way to efficient MOPOMDP planning
- Solve as series of scalarized POMDPs
- Point-based POMDP planners
- Smart choices of scalarized instances
Point-based methods represent value by $\alpha$-vectors

\[
\alpha = \begin{pmatrix} V(s_1) \\ V(s_2) \\ V(s_3) \\ V(s_4) \end{pmatrix}
\]

$V^\alpha(b_0) = b_0 \cdot \alpha$

Adapt point-based methods to return $\alpha$-matrices

\[
A = \begin{pmatrix} obj 1 : & obj 2 : \\ V_1(s_1) & V_2(s_1) \\ V_1(s_2) & V_2(s_2) \\ V_1(s_3) & V_2(s_3) \\ V_1(s_4) & V_2(s_4) \end{pmatrix}
\]

$V^A(b_0) = b_0 A$

Adapted point-based backups
Optimistic linear support for POMDPs

- Point-based methods represent value by $\alpha$-vectors

\[
\alpha = \begin{pmatrix} V(s_1) \\ V(s_2) \\ V(s_3) \\ V(s_4) \end{pmatrix}
\]

- Adapt point-based methods to return $\alpha$-matrices

\[
A = \begin{pmatrix} \text{obj 1}: & \text{obj 2}: \\ V_1(s_1) & V_2(s_1) \\ V_1(s_2) & V_2(s_2) \\ V_1(s_3) & V_2(s_3) \\ V_1(s_4) & V_2(s_4) \end{pmatrix}
\]

- $V^\alpha(b_0) = b_0 \cdot \alpha$

- $V^A(b_0) = b_0A$

- Adapted point-based backups
Multi-objective Point-based backups

**Back-projection** of $\alpha$-vectors $\alpha_i \in A_k$:  

$$g_i^{a,o}(s) = \sum_{s' \in S} O(a, s', o) T(s, a, s') \alpha_i(s')$$

$$\alpha_{k+1}^{b,a} = r^a + \gamma \sum_{o \in \Omega} \arg \max_{b} b \cdot g^{a,o}$$

backup$(A_k, b) = \arg \max_{\alpha_{k+1}^{b,a}} b \cdot \alpha_{k+1}^{b,a}$

**Back-projection** of $\alpha$-matrices $A_i \in A_k$, for a given $w$:  

$$G_i^{a,o}(s) = \sum_{s' \in S} O(a, s', o) T(s, a, s') A_i(s')$$

$$A_{k+1}^{b,a} = r^a + \gamma \sum_{o \in \Omega} \arg \max_{b} b \cdot G^{a,o} w$$

backupMO$(A_k, b, w) = \arg \max_{A_{k+1}^{b,a}} b A_{k+1}^{a,b} w$
Multi-objective Point-based backups

**Back-projection** of $\alpha$-vectors $\alpha_i \in \mathcal{A}_k$:

$$g^{a,o}_i(s) = \sum_{s' \in S} O(a, s', o) T(s, a, s') \alpha_i(s')$$

$$\alpha^{b,a}_{k+1} = r^a + \gamma \sum_{o \in \Omega} \arg \max b \cdot g^{a,o}$$

$$\text{backup}(\mathcal{A}_k, b) = \arg \max b \cdot \alpha^{b,a}_{k+1}$$

**Back-projection** of $\alpha$-matrices $A_i \in \mathcal{A}_k$, for a given $w$:

$$G^{a,o}_i(s) = \sum_{s' \in S} O(a, s', o) T(s, a, s') A_i(s')$$

$$A^{b,a}_{k+1} = r^a + \gamma \sum_{o \in \Omega} \arg \max b \cdot G^{a,o}$$

$$\text{backupMO}(\mathcal{A}_k, b, w) = \arg \max b A^{a,b}_{k+1} w$$

D.M. Roijers (UvA) Optimistic Linear Support October 20, 2015 18 / 22
Starting from scratch for each \( \mathbf{w} \) is inefficient.

Intuition: when \( \mathbf{w} \) and \( \mathbf{w}' \) are close, so are the optimal policies and values.

**Hot start** point-based planner using \( \alpha \)-matrices.

More and more effective as \( \mathbf{w} \)'s lie closer together.
Theoretical results

Theorem

**OLSAR** requires a finite number of calls to the point-based solver to converge.

Theorem

**OLSAR** produces an \(\varepsilon\)-approximate solution set. \(\varepsilon\) is inherited from the single-objective method.
Sample of results: 3-objective tiger

D.M. Roijers (UvA)
Optimistic Linear Support
October 20, 2015
Conclusions

• Use point-based methods for MOPOMDPs

• First method that reasonably scales

• Bounded approximation

• Alpha reuse is key to keeping MOPOMDPs tractable
OLS\( (m, \text{SolveSingleObjective}, \epsilon) \) // Without reuse

\[
S \leftarrow \emptyset \quad //\text{partial CCS} \\
Q \leftarrow \text{an empty priority queue} \\
\textbf{foreach extremum of the weight simplex } w_e \text{ do} \\
\quad \text{Q.add}(w_e, \infty) \quad //\text{add extrema with infinite priority} \\
\textbf{while } \neg Q.\text{isEmpty()} \land \neg \text{timeOut} \text{ do} \\
\quad w \leftarrow Q.\text{pop()} \\
\quad V \leftarrow \text{SolveSingleObjective}(m, w) \\
\quad \textbf{if } V \notin S \text{ then} \\
\quad \quad S \leftarrow S \cup \{V\} \\
\quad \quad \text{delete obsolete corner weights from } Q \\
\quad \quad \mathcal{W}_V \leftarrow \text{the new corner weights that involve } V \\
\quad \textbf{foreach } w \in \mathcal{W}_V \text{ do} \\
\quad \quad \Delta_r(w) \leftarrow \text{max. possible rel. improvement at } w \\
\quad \quad \textbf{if } \Delta_r(w) > \epsilon \text{ then} \\
\quad \quad \quad \text{Q.add}(w, \Delta_r(w)) \\
\textbf{return } S \text{ and the highest } \Delta_r(w) \text{ left in } Q
\[
\begin{align*}
\mathcal{A}' & \leftarrow \mathcal{A}; \\
\mathcal{A} & \leftarrow \{-\infty\}; \\
\text{while } & \max_b \max_{\mathcal{A}' \in \mathcal{A}'} b\mathcal{A}'w - (\max_{\mathcal{A} \in \mathcal{A}} b\mathcal{A}w) > \eta \text{ do } \\
& \quad \mathcal{A} \leftarrow \mathcal{A}'; \quad \mathcal{A}' \leftarrow \emptyset; \quad B' \leftarrow B; \\
& \quad \text{while } B' \neq \emptyset \text{ do } \\
& \quad \quad \text{Randomly select } b \text{ from } B'; \\
& \quad \quad \mathcal{A} \leftarrow \text{backupMO}(\mathcal{A}, b, \mathbf{w}); \\
& \quad \quad \mathcal{A}' \leftarrow \mathcal{A}' \cup \{ \arg \max_{\mathcal{A}' \in \mathcal{A}' \cup \mathcal{A}} b\mathcal{A}'w \}; \\
& \quad \quad B' \leftarrow \{ b \in B' : \max_{\mathcal{A}' \in \mathcal{A}'} b\mathcal{A}'w < \max_{\mathcal{A} \in \mathcal{A}} b\mathcal{A}w \}; \\
\text{return } & \mathcal{A}'; \\
\end{align*}
\]
OLSAR($b_0, \eta$) // With Reuse

\[ X \leftarrow \emptyset; \] // partial CCS of multi-objective value vectors $V_{b_0}$
\[ WV_{old} \leftarrow \emptyset; \] // searched weights and scalarized values
\[ Q \leftarrow \text{priority queue with weights to search}; \]
Add extrema of the weight simplex to $Q$ with infinite priority;
\[ A_{all} \leftarrow \text{a set of } \alpha\text{-matrices forming a lower bound on the value}; \]
\[ B \leftarrow \text{set of sampled belief points (e.g., by random exploration)}; \]
\[ \text{while } \neg Q.\text{isEmpty}() \land \neg \text{timeOut} \text{ do} \]
\[ w \leftarrow Q.\text{dequeue}(); \] // Retrieve a weight vector
\[ A_r \leftarrow \text{select the best } A \text{ from } A_{all} \text{ for each } b \in B, \text{ given } w; \]
\[ A_w \leftarrow \text{solveScalarizedPOMDP}(A_r, B, w, \eta); \]
\[ V_{b_0} \leftarrow \max_{A \in A_w} b_0 A w; \]
\[ A_{all} \leftarrow A_{all} \cup A_w; \]
\[ WV_{old} = WV_{old} \cup \{(w, w \cdot V_{b_0})\}; \]
\[ \text{if } V_{b_0} \notin X \text{ then} \]
\[ X \leftarrow X \cup \{V_{b_0}\}; \]
\[ W \leftarrow \text{compute new corner weights and maximum possible improvements} \]
\[ (w, \Delta w) \text{ using } WV_{old} \text{ and } X; \]
\[ Q.\text{addAll}(W); \]
return $X$;
Theorem

(Cheng 1988) The maximum value of:

$$\max_{w,u \in CCS} \min_{v \in S} w \cdot u - w \cdot v,$$

i.e., the maximal improvement to $S$ by adding a vector to it, is at one of the corner weights.
An optimistic hypothetical CCS, \( \overline{CCS} \) is a set of payoff vectors that yields the highest possible scalarized value for all possible \( w \) consistent with finding the vectors \( S \) at the weights in \( \mathcal{W} \).

For a given \( w \), the scalarized value of \( u^*_{\overline{CCS}}(w) \) can be found by solving the following linear program:

\[
\begin{align*}
\text{max} \quad & w \cdot v \\
\text{subject to} \quad & \mathcal{W}v \leq u^*_{S,\mathcal{W}},
\end{align*}
\]

where \( u^*_{S,\mathcal{W}} \) is a vector containing \( u^*_S(w') \) for all \( w' \in \mathcal{W} \).