Fast Bellman Updates for Robust MDPs

Chin Pang Ho¹, Marek Petrik², Wolfram Wiesemann¹

¹. Imperial College,
². University of New Hampshire
More Reliable Reinforcement Learning

- Medicine and other domains need policies with low failure probability

- Transition probabilities estimated from data $\Rightarrow$ errors

- Errors compound in reinforcement learning

- Small errors in probabilities $\Rightarrow$ large impact on policy quality (bad things happen)
Robust Markov Decision Processes

+ Flexible model of imprecise transition probabilities
+ Policies resistant to model errors
+ Computing policies is poly-time
  - Slow in practice

**Contribution**: Fast algorithms for common RMDPs
Robust Bellman Update

- Solve RMDPs using (approximate) value iteration

- Bellman update:

\[
Bv = \max_a \left( r_{s,a} + \gamma \cdot \bar{p}^T_{s,a} v \right)
\]

- Robust Bellman update:

\[
Lv = \max_a \min_p \left\{ r_{s,a} + \gamma \cdot p^T v : \|p - \bar{p}_{s,a}\| \leq \psi_{s,a} \right\}
\]
Robustness Flavors: Rectangularity

- **State-action-Rect:** Independent errors

\[ L_v = \max_a \min_p \left\{ r_{s,a} + \gamma \cdot p^T v : \| p - \bar{p}_{s,a} \| \leq \psi_{s,a} \right\} \]

- **State-Rect:** Correlated errors

\[ L_v = \max_\pi \min_{p_a} \left\{ \sum_a \pi(a) \left( r_{s,a} + \gamma \cdot p_a^T v \right) : \sum_a \| p_a - \bar{p}_{s,a} \| \leq \psi_s \right\} \]
Robustness Flavors: Distance Metric

$L_1$ Norm

\[ \| p - \bar{p}_{s,a} \|_1 \leq \psi \]

Weighted $L_1$ Norm

\[ \| p - \bar{p}_{s,a} \|_{1,w} \leq \psi \]
Computing Robust Bellman Update

- Find the worst-case probability $\min_p$?
- Linear programming: (weighted) $L_1$ norm as a distance metric

Timing Robust Bellman updates:
- Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver
- Bellman update: 0.04 s

Distance Metric
- Rectangularity
- $L_1$ Norm
- $w$-L$_1$ Norm
- State-action: 1.1 min, 1.2 min
- State: 16.7 min, 13.4 min

LP scales as $O(n^3)$.
Must solve for every state and iteration!
Computing Robust Bellman Update

- Find the worst-case probability $\min_p p$
- Linear programming: (weighted) $L_1$ norm as a distance metric

Timing Robust Bellman updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver

Bellman update: 0.04 s
Computing Robust Bellman Update

- Find the worst-case probability $\min_p$?
- Linear programming: (weighted) $L_1$ norm as a distance metric

Timing Robust Bellman updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver

Bellman update: 0.04 s

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
</tr>
<tr>
<td>State-action</td>
<td>1.1 min</td>
</tr>
<tr>
<td>State</td>
<td>16.7 min</td>
</tr>
</tbody>
</table>

LP scales as $\geq O(n^3)$. 
Computing Robust Bellman Update

- Find the worst-case probability $\min_p$?
- Linear programming: (weighted) $L_1$ norm as a distance metric

Timing Robust Bellman updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver

Bellman update: 0.04 s

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
<td>$w-L_1$ Norm</td>
<td></td>
</tr>
<tr>
<td>State-action</td>
<td>1.1 min</td>
<td>1.2 min</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>16.7 min</td>
<td>13.4 min</td>
<td></td>
</tr>
</tbody>
</table>

LP scales as $\geq O(n^3)$. Must solve for every state and iteration!
Prior Work: Fast Algorithms

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( L_1 ) Norm</td>
</tr>
<tr>
<td>State-action</td>
<td>( O(n \log n) )</td>
</tr>
<tr>
<td>State</td>
<td>?</td>
</tr>
</tbody>
</table>

*Problem size: \( n = \text{states} \times \text{actions} \)*

\( O(n \log n) \) algorithm:

- Robust dynamic programming (Iyengar 2006)
- MBIE (Strehl et al, 2008), used in UCRL2, …
- Does not extend to other robustness types
Prior Work: Fast Algorithms

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
</tr>
<tr>
<td>State-action</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>State</td>
<td>?</td>
</tr>
</tbody>
</table>

Problem size: $n = \text{states} \times \text{actions}$

Better solutions

$O(n \log n)$ algorithm:

- Robust dynamic programming (Iyengar 2006)
- MBIE (Strehl et al, 2008), used in UCRL2, ...
- Does not extend to other robustness types
Our Contribution: Fast Robust Updates

Worst-case complexity, new results highlighted

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
</tr>
<tr>
<td>State-action</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>State</td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>

Problem size: $n = \text{states} \times \text{actions}$

Structural constant: $k \leq \text{states}$
Our Contribution: Fast Robust Updates

Worst-case complexity, new results highlighted

| Rectangularity  | Distance Metric
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
</tr>
<tr>
<td>State-action</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>State</td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>

Problem size: $n = \text{states} \times \text{actions}$

Structural constant: $k \leq \text{states}$

- Homotopy Continuation Method
Our Contribution: Fast Robust Updates

Worst-case complexity, new results highlighted

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
</tr>
<tr>
<td>State-action</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>State</td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>

Problem size: $n = \text{states} \times \text{actions}$

Structural constant: $k \leq \text{states}$

- **Bisection + Homotopy Method**: randomized policies in combinatorial time!
Our Contribution: Practical Complexity

Timing Robust Bellman updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver / Homotopy + Bisection

<table>
<thead>
<tr>
<th>Rectangularity</th>
<th>Distance Metric</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1$ Norm</td>
<td>$w-L_1$ Norm</td>
<td></td>
</tr>
<tr>
<td>State-action</td>
<td>1.1 min / 0.6s</td>
<td>1.2 min / 0.8s</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>16.7 min / 0.7s</td>
<td>13.4 min / 1.2s</td>
<td></td>
</tr>
</tbody>
</table>

Bellman update: 0.04 s
How It Works

- **Homotopy Method**: Similar to LARS for LASSO, few linear segments, easy to trace
How It Works

- **Homotopy Method**: Similar to LARS for LASSO, few linear segments, easy to trace

- **Bisection**: Small dimensionality of the dual + fast homotopy
Summary of Contributions

• New fast methods for wider variety of robust Bellman Updates

• Pseudo-linear time complexity

• Computes primal solutions, not only duals (*skipped*)

• Empirical results: 500 – 40,000 × speedup over Gurobi LP (*skipped*)

• Also useful in model-based exploration (MBIE, UCRL2, …)

**Poster:** Hall B # 87