Fast Bellman Updates for Robust MDPs

Chin Pang Ho[1], **Marek Petrik**[2], Wolfram Wiesemann[1]

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 ⇒ errors
- Errors compound in reinforcement learning
- Small errors in probabilities ⇒ 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

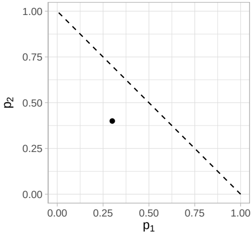
Contribution: Fast algorithms for common RMDPs

Robust Bellman Update

- Solve RMDPs using (approximate) value iteration

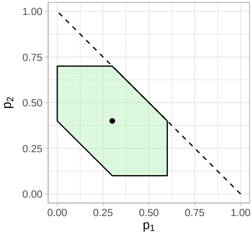
- Bellman update:

$$Bv = \max_a \left(r_{s,a} + \gamma \cdot \bar{p}_{s,a}^{\mathsf{T}} v \right)$$



- Robust Bellman update:

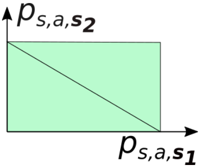
$$Lv = \max_a \min_p \left\{ r_{s,a} + \gamma \cdot p^{\mathsf{T}} v \ : \ \|p - \bar{p}_{s,a}\| \le \psi_{s,a} \right\}$$



Robustness Flavors: Rectangularity

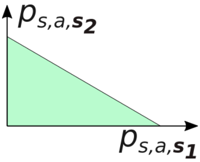
- State-action-Rect: Independent errors

$$Lv = \max_a \min_p \left\{ r_{s,a} + \gamma \cdot p^\top v \ : \ \|p - \bar{p}_{s,a}\| \le \psi_{s,a} \right\}$$



- State-Rect: Correlated errors

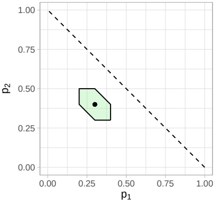
$$Lv = \max_\pi \min_{p_a} \left\{ \sum_a \pi(a) \left(r_{s,a} + \gamma \cdot p_a^\top v \right) \ : \ \sum_a \|p_a - \bar{p}_{s,a}\| \le \psi_s \right\}$$

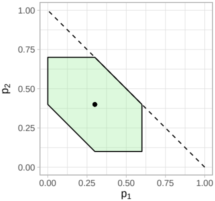


Robustness Flavors: Distance Metric

L_1 Norm

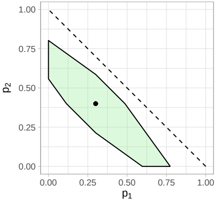
$$\|p - \bar{p}_{s,a}\|_1 \leq \psi$$

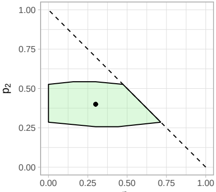




Weighted L_1 Norm

$$\|p - \bar{p}_{s,a}\|_{1,w} \leq \psi$$





Computing Robust Bellman Update

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

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

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

Rectangularity	Distance Metric	
	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 $\geq O(n^3)$.

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

	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 $\geq O(n^3)$. Must solve for every state and iteration!

Prior Work: Fast Algorithms

Rectangularity	**Distance Metric** L_1 Norm	w-L_1 Norm
State-action	$O(n \log n)$	?
State	?	?

Problem size: n = states $\times$ 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

Rectangularity	**Distance Metric** L_1 Norm	w-L_1 Norm
State-action	$O(n \log n)$	?
State	?	?

Problem size: n = states $\times$ 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

	Distance Metric	
Rectangularity	L_1 Norm	w-L_1 Norm
State-action	$O(n \log n)$	$O(k\, n \log n)$
State	$O(n \log n)$	$O(k\, n \log n)$

Problem size: n = states $\times$ actions
Structural constant: $k \leq$ states

Our Contribution: Fast Robust Updates

Worst-case complexity, new results highlighted

Rectangularity	Distance Metric L_1 Norm	Distance Metric w-L_1 Norm
State-action	$O(n \log n)$	$O(k\, n \log n)$
State	$O(n \log n)$	$O(k\, n \log n)$

Problem size: n = states $\times$ actions
Structural constant: $k \leq$ states

- Homotopy Continuation Method

Our Contribution: Fast Robust Updates

Worst-case complexity, new results highlighted

Rectangularity	Distance Metric: L_1 Norm	Distance Metric: w-L_1 Norm
State-action	$O(n \log n)$	$O(k\, n \log n)$
State	$O(n \log n)$	$O(k\, n \log n)$

Problem size: n = states $\times$ actions
Structural constant: $k \leq$ states

- Bisection + Homotopy Method: randomized policies in combinatorial time!

Our Contribution: Practical Complexity

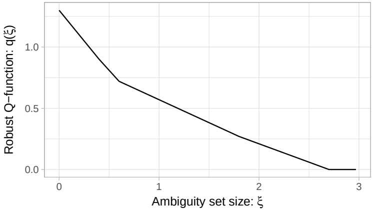
Timing Robust Bellman updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver / Homotopy + Bisection

	Distance Metric	
Rectangularity	L_1 Norm	w-L_1 Norm
State-action	1.1 min / 0.6s	1.2 min / 0.8s
State	16.7 min / 0.7s	13.4 min / 1.2s

Bellman update: 0.04 s

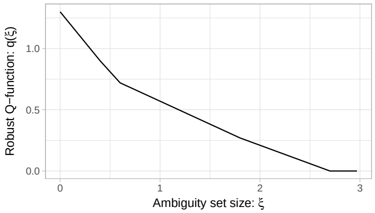
How It Works

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

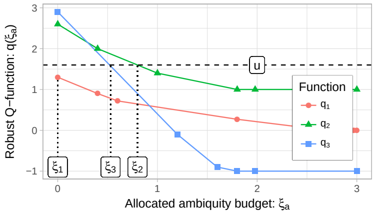


How It Works

- 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 $\times$ speedup over Gurobi LP (*skipped*)
- Also useful in model-based exploration (MBIE,UCRL2,...)

Poster: Hall B # 87