Efficient Value Iteration for $\mathtt{s}$-rectangular Robust Markov Decision Processes

Navdeep Kumar [1] **Kaixin Wang** [1] **Kfir Levy** [1] **Shie Mannor** [1 2]

Abstract

We focus on $\mathtt{s}$-rectangular robust Markov decision processes (MDPs), which capture interconnected uncertainties across different actions within each state. This framework is more general compared to $\mathtt{sa}$-rectangular robust MDPs, where uncertainties in each action are independent. However, the introduced interdependence significantly amplifies the complexity of the problem. Existing methods either have slow performance guarantees or are inapplicable to even moderately large state spaces. In this work, we derive optimal robust Bellman operators in explicit forms. This leads to robust value iteration methods with significantly faster time complexities than existing approaches, which can be used in large state spaces. Further, our findings reveal that the optimal policies demonstrate a novel threshold behavior, selectively favoring a limited set of actions based on their respective advantage functions. Additionally, our study uncovers a noteworthy connection between the robustness of a policy and the variance in its value function, highlighting that policies with lower variance exhibit greater resilience.

1. Introduction

In Markov Decision Processes (MDPs), an agent interacts with the environment and learns to behave optimally in it (Sutton & Barto, 2018). However, the MDP solution can be highly sensitive to even minor alterations in model parameters (Mannor et al., 2004). Consequently, caution is warranted when applying MDP solutions in scenarios involving dynamic model changes or parameter uncertainties to avoid catastrophic failures. On the other hand, robust MDPs consider a possible set of environments (uncertainty sets) instead of a single environment, and the goal is to obtain a solution whose worst performance over the uncertainty set is optimal. Hence, robust MDPs' solutions are more resilient in the face of model uncertainty (Hanasusanto & Kuhn, 2013; Tamar et al., 2014; Iyengar, 2005). The study of robust MDPs is further motivated by their potential to yield superior generalization compared to non-robust solutions (Xu & Mannor, 2010; Zhao et al., 2019; Packer et al., 2018).

Unfortunately, solving robust MDPs is a complex problem in general. Consequently, most existing solutions exist for a special class of uncertainty sets, which are $\mathtt{sa}$-rectangular (Nilim & Ghaoui, 2005; Iyengar, 2005; Wang & Zou, 2021; 2022). This setting models various problems, where model uncertainty in one state-action is independent from uncertainty in other state-actions. However, in many cases, the model uncertainty in one action can be coupled to the other actions in the same state, which are captured by $\mathtt{s}$-rectangular uncertainty sets (Wiesemann et al., 2013). However, this coupling comes with additional complexities and presents a great difficulty in obtaining the solutions, and hence, only a handful of existing work exists on the topic (Wiesemann et al., 2013; Ho et al., 2020; Derman et al., 2021). Unfortunately, capturing the further coupling in uncertainties across different states is proven to make the problem strongly NP-hard (Wiesemann et al., 2013).

Related works. Most of the works in robust MDPs are for $\mathtt{sa}$-rectangular uncertainty sets (Iyengar, 2005; Nilim & Ghaoui, 2005; Kaufman & Schaefer, 2013; Bagnell et al., 2001; Hanasusanto & Kuhn, 2013; Abdullah et al., 2019). Notably, for $\mathtt{sa}$-rectangular R-contamination robust MDPs, (Wang & Zou, 2021) derived robust Bellman operators, which are equivalent to value-regularized-non-robust Bellman operators, enabling efficient robust value iteration. Building upon this work, (Wang & Zou, 2022) derived a robust policy gradient equivalent to a non-robust policy gradient with regularizer and correction terms. Unfortunately, these methods cannot be naturally generalized to $\mathtt{s}$-rectangular robust MDPs.

A naive LP (linear programming) can evaluate $\mathtt{s}$-rectangular optimal robust Bellman operators with uncertainty sets constrained by L_1 norm, with time complexity $O(S^{5.5}A^{4.5})$, where S, A is the cardinality of state space, and action space respectively. This brute force method is prohibitively expensive and restrictive to only polyhedral uncertainty sets.

[1]Technion, Haifa, Israel [2]Nvidia, Haifa, Israel. Correspondence to: Navdeep Kumar <navdeepkumar@alum.iisc.ac.in>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Table 1. Related Work: Computation of Optimal s-rectangular L_p constrained Robust Bellman Operator $\mathcal{T}_{\mathcal{U}^s_p} v$

	LP	(Wiesemann et al., 2013)	(Ho et al., 2020)	(Derman et al., 2021)	Ours	Ours	Ours
p	1	$1, 2$	1	p	$1, \infty$	2	p
Method	LP	SDP	Bisection	Gradient	Close form	Algorithm	Binary Search
Solution	Approx	Approx	**Exact**	Approx	**Exact**	**Exact**	Approx
Complexity $\tilde{O}$	$S^{\frac{11}{2}}A^{\frac{9}{2}}$	$S^{\frac{11}{2}} + S^3A$	S^3A	NA	$\mathbf{S^2A}$	$\mathbf{S^2A}$	$\mathbf{S^2A}$
Contraction factor	γ	γ	γ	$\gamma(1+\sqrt{S}\beta^*)$	γ	γ	γ
Valid Kernels	Yes	Yes	Yes	**No**	Yes	Yes	Yes
Optimal Policy Characterization	No	No	No	No	**Yes**	**Yes**	**Yes**

SDP and LP stand for semi-definite programming and linear programming, respectively, $\beta^* = \max_{s\in\mathcal{S}} \beta_s$, $\tilde{O}$ hides logarithmic factors in S, A, ϵ.

In further generality, (Wiesemann et al., 2013) uses the SDP (semi-definite programming) methods for uncertainty set which are finite intersections of closed half-spaces and ellipsoids, as described in (3a) and (3b) of (Wiesemann et al., 2013). This approach can approximately evaluate optimal s-rectangular Bellman operator up to ϵ-accuracy, with time complexity of $O(m^3 l^{3/2} S \log \epsilon^{-1} + mS^2A)$ (Corollary 1 of (Wiesemann et al., 2013)), where m, l is degree of freedom and number of constraints respectively. This framework can describe s-rectangular L_2 constrained uncertainty sets for the degree of freedom $m \geq S$, and number of constraints $l = S$ as a special case. This makes their worst case complexity $O(S^{5.5} \log \epsilon^{-1} + S^3A)$ which is very expensive.

On the other hand, focusing only on L_1 constrained s-rectangular robust MDPs, (Ho et al., 2020) for the first time, evaluated the optimal robust Bellman operator exactly. The approach uses a combination of homotopy and bisection methods which has the time complexity of $O(CSA\log(SA) + A\log CS \log CSA)$ at each state, where $0 \leq C \leq S$ is the number of states where uncertainty is present. Hence, the worst-case time complexity to evaluate robust optimal Bellman operator for all states is $O(S^3A\log(SA) + SA\log S^2 \log S^2A)$ which has a considerable speedup over existing LP and SDP methods. However, how this method can be extended for other L_p norms is unclear.

Additionally, some works have explored robust MDPs from a regularization perspective (Derman et al., 2021; Derman & Mannor, 2020; Husain et al., 2021; Eysenbach & Levine, 2021). Specifically, (Derman et al., 2021) showed that s-rectangular L_p robust MDPs are equivalent to reward-value-policy regularized MDPs, and proposed R^2 Bellman operators. These optimal R^2 Bellman operators are obtained via gradient-based policy iteration methods. However, the work has the following limitations: a) It makes unrealistic assumptions on kernel noise (Corollary 4.1 of (Derman et al., 2021)); b) It assumes an extra condition on the uncertainty radius (assumption 5.1 of (Derman et al., 2021)), and; c) The contraction factor of R^2 Bellman operator is greater than $\gamma(1 + \sqrt{|\mathcal{S}|}\max_{s\in\mathcal{S}} \beta_s)$ where $\gamma, \mathcal{S}, \beta_s$ is discount factor, state space, and uncertainty radius in transition kernel in state s respectively (Assumption 5.1 and Proposition 5.1 of (Derman et al., 2021)). The above contraction factor quickly grows to 1, making the R^2 Bellman operators inapplicable for even moderately large state-space problems.

As summarized in Table 1, the existing methods are either slow or inapplicable for large state problems. Furthermore, it is well established that the robust policies in s-rectangular robust MDPs can be stochastic (Wiesemann et al., 2013). Unfortunately, nothing is known about the nature of its stochasticity.

We summarize our **contributions** as follows with a summary in Table 1:

- Our work introduces a regularizer, providing a novel insight that policies with lower variance in value function are more robust to the change in transition kernels.
- We unveil the exact nature of optimal robust policies in s-rectangular robust MDPs and show it as threshold policies. It plays only the top few actions proportional

to their advantage functions and avoids playing very bad actions. This policy class contrasts with the widely used soft-max policies that play all the actions (Mai & Jaillet, 2021; Grill et al., 2019; Yang et al., 2019).

- Our methods evaluate the optimal robust Bellman operators for $\mathtt{s}$-rectangular uncertainty sets constrained by L_p norms, with time complexity of $O(S^2A + SA\log A)$ for $p = 1, 2$ and $O(S^2A + SA\log\frac{A}{\epsilon})$ for general p, which is faster than existing methods by at least a factor of S, which leads to efficient value iteration which can be used in large state space problems.

Further, (Kumar et al., 2023; Gadot et al., 2023) build upon our work to derive policy gradient methods for robust MDPs with uncertainty set constrained by L_p norms, then (Zhou et al., 2023; Wang et al., 2023) utilized the insights to develop pessimistic sampling methods for robust MDPs. Additionally, there exist different lines of work such as soft-robust MDPs (Derman et al., 2018), k-rectangular robust MDPs (Mannor et al., 2016), r-rectangular robust MDPs (Goyal & Grand-Clément, 2018). However, this work focuses on $\mathtt{s}$-rectangular uncertainty sets.

2. Preliminary

A Markov Decision Process (MDP) can be described as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is a transition kernel mapping $\mathcal{S}\times\mathcal{A}$ to $\Delta_{\mathcal{S}}$, R is a reward function mapping $\mathcal{S}\times\mathcal{A}$ to $\mathbb{R}$, μ is an initial distribution over states in $\mathcal{S}$, and γ is a discount factor in $[0,1)$ (Puterman, 1994; Sutton & Barto, 2018). A policy $\pi : \mathcal{S} \to \Delta_{\mathcal{S}}$ is a decision rule that maps state space to a probability distribution over action space. Further, $\pi(a|s), P(s'|s,a)$ denotes the probability of taking action a in state s by policy π, and the probability of transition to state s' from state s under action a respectively. In addition, we denote $P^\pi(s'|s) = \sum_a \pi(a|s)P(s'|s,a)$ and $R^\pi(s) = \sum_a \pi(a|s)R(s,a)$ as short-hands. The return of a policy π, is defined as $\rho^\pi_{(P,R)} = \langle \mu, v^\pi_{(P,R)}\rangle$ where $v^\pi_{(P,R)} := (I - \gamma P^\pi)^{-1}R^\pi$ is value function (Puterman, 1994).

A robust Markov Decision Process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mu)$ which generalizes the standard MDP, by containing a set of transition kernels $\mathcal{P}$ and set of reward functions $\mathcal{R}$. Let uncertainty set $\mathcal{U} = \mathcal{P}\times\mathcal{R}$ be a set of tuples of transition kernels and reward functions (Iyengar, 2005; Nilim & Ghaoui, 2005). The robust return of a policy π, is its performance over the uncertainty set $\mathcal{U}$, defined as $\rho^\pi_{\mathcal{U}} = \min_{(P,R)\in\mathcal{U}} \rho^\pi_{(P,R)}$. The objective is to find an optimal robust policy $\pi^*_{\mathcal{U}}$ that achieves the optimal robust performance $\rho^*_{\mathcal{U}}$, defined as

$$\rho^*_{\mathcal{U}} = \max_\pi \rho^\pi_{\mathcal{U}}. \tag{1}$$

Unfortunately, a general solution to the robust objective (1), is proven to be strongly NP-hard for both non-convex sets and convex ones (Wiesemann et al., 2013). Hence, it is common practice to take the $\mathtt{sa}$-rectangular uncertainty sets $\mathcal{U}^{\mathtt{sa}}$, where ambiguity in each state and action are independent (Iyengar, 2005; Nilim & Ghaoui, 2005; Wang & Zou, 2021; 2022). Formally defined as $\mathcal{U}^{\mathtt{sa}} = \mathcal{P}\times\mathcal{R}$ is decomposed over state-action-wise as $\mathcal{R} = \times_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{R}_{s,a}$ and $\mathcal{P} = \times_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{P}_{s,a}$ where $\mathcal{R}_{s,a}, \mathcal{P}_{s,a}$ components sets.

However, many classes of uncertainty sets arise in practice, where ambiguities in a given state are correlated. This type of uncertainty sets are captured by $\mathtt{s}$-rectangular uncertainty sets $\mathcal{U}^{\mathtt{s}}$, defined as $\mathcal{U}^{\mathtt{s}} = \mathcal{P}\times\mathcal{R}$, where $\mathcal{R}$ and $\mathcal{P}$ can be decomposed state-wise as $\mathcal{R} = \times_{s\in\mathcal{S}}\mathcal{R}_s$ and $\mathcal{P} = \times_{s\in\mathcal{S}}\mathcal{P}_s$ (Wiesemann et al., 2013). Note that its special case is $\mathtt{sa}$-rectangular uncertainty sets.

The robust value function $v^\pi_{\mathcal{U}}$ and optimal robust value function $v^*_{\mathcal{U}}$ (Iyengar, 2005; Nilim & Ghaoui, 2005), for any uncertainty set $\mathcal{U}$ can be defined state wise, for all π, as

$$v^\pi_{\mathcal{U}}(s) = \min_{(P,R)\in\mathcal{U}} v^\pi_{(P,R)}(s), \qquad v^*_{\mathcal{U}}(s) = \min_{(P,R)\in\mathcal{U}} v^\pi_{\mathcal{U}}(s).$$

Fortunately, the decoupling structure in $\mathtt{s}$-rectangular uncertainty sets allows the existence of a kernel and a reward function that minimizes the value function over the uncertainty set for each state for any given policy. Similarly, it allows the existence of an optimal optimal robust policy that maximizes the robust value in each state (Wiesemann et al., 2013). Mathematically, the robust value function is to be rewritten as

$$v^*_{\mathcal{U}^{\mathtt{s}}} = \max_\pi v^\pi_{\mathcal{U}^{\mathtt{s}}}, \qquad v^\pi_{\mathcal{U}^{\mathtt{s}}} = \min_{(P,R)\in\mathcal{U}^{\mathtt{s}}} v^\pi_{(P,R)}.$$

Hence, the robust return can be rewritten as $\rho^\pi_{\mathcal{U}} = \langle \mu, v^\pi_{\mathcal{U}}\rangle$, and $\rho^*_{\mathcal{U}} = \langle\mu, v^*_{\mathcal{U}}\rangle$. Most importantly, this rectangularity implies the existence of contractive robust Bellman operators, which are pivotal same as non-robust MDPs (Wiesemann et al., 2013). Specifically, the robust value function $v^\pi_{\mathcal{U}}$, and the optimal robust value function $v^*_{\mathcal{U}}$ is the fixed point of the robust Bellman operator $\mathcal{T}^\pi_{\mathcal{U}}$ and the optimal robust Bellman operator $\mathcal{T}^*_{\mathcal{U}}$ respectively (Wiesemann et al., 2013; Iyengar, 2005), defined as

$$\mathcal{T}^\pi_{\mathcal{U}} v := \min_{(P,R)\in\mathcal{U}} T^\pi_{(P,R)} v, \quad \text{and} \quad \mathcal{T}^*_{\mathcal{U}} v := \max_\pi \mathcal{T}^\pi_{\mathcal{U}} v,$$

where $T^\pi_{(P,R)}v := R^\pi + \gamma P^\pi v$ is non-robust Bellman operator (Puterman, 1994). Moreover, these robust Bellman operators are γ contraction maps (Wiesemann et al., 2013), that is $\|\mathcal{T}^*_{\mathcal{U}}v - \mathcal{T}^*_{\mathcal{U}}u\|_\infty \le \gamma\|u - v\|_\infty$, and $\|\mathcal{T}^\pi_{\mathcal{U}}v - \mathcal{T}^\pi_{\mathcal{U}}u\|_\infty \le \gamma\|u - v\|_\infty, \quad \forall \pi, u, v$. So for all initial values v^π_0, v^*_0, sequences defined as $v^\pi_{n+1} := \mathcal{T}^\pi_{\mathcal{U}} v^\pi_n, v^*_{n+1} := \mathcal{T}^*_{\mathcal{U}} v^*_n$ converge linearly to their respective fixed points, that is $v^\pi_n \to v^\pi_{\mathcal{U}}$ and $v^*_n \to v^*_{\mathcal{U}}$. Given this optimal robust value

Table 2. p-variance

p	$\kappa_p(v)$	Remark
∞	$\frac{\max_s v(s)-\min_s v(s)}{2}$	Semi-norm
2	$\sqrt{\sum_s \left(v(s) - \frac{\sum_s v(s)}{S} \right)^2}$	Variance
1	$\sum_{i=1}^{\lfloor (S+1)/2 \rfloor} v(s_i)$ - $\sum_{i=\lceil (S+1)/2 \rceil}^{S} v(s_i)$	Top half - lower half

where v is sorted, i.e. $v(s_i) \geq v(s_{i+1}) \quad \forall i$.

function, the optimal robust policy can be computed as: $\pi^*_{\mathcal{U}} \in \arg\max_\pi \mathcal{T}^\pi_{\mathcal{U}} v^*_{\mathcal{U}}$ (Wiesemann et al., 2013). This makes the robust value iteration an attractive method for solving s-rectangular robust MDPs.

3. Method

We consider uncertainty sets constrained by L_p norm around the nominal values that occur naturally in practice (Derman et al., 2021; Ho et al., 2020; Auer et al., 2008). We derive robust Bellman operators in concrete forms for these uncertainty sets, producing efficient robust value iteration methods. Additionally, we explicitly obtain the exact nature of the robust optimal policies.

We begin with a more straightforward case of sa-rectangular robust MDPs as a warm-up to the more complex s-rectangular counterpart.

Here, we make a few useful definitions that will be used throughout the paper. We reserve q for Holder conjugate of p, i.e. $\frac{1}{p} + \frac{1}{q} = 1$. Let p-variance function $\kappa_p : \mathcal{S} \to \mathbb{R}$ be defined as

$$\kappa_p(v) := \min_{\omega \in \mathbb{R}} \|v - \omega \mathbb{1}\|_p, \tag{2}$$

where $\mathbb{1}$ is all-ones vector. For $p = 1, 2, \infty$, the p-variance function κ_p has intuitive closed forms as summarized in Table 2. For general p, it can be calculated by binary search in the range $[\min_s v(s), \max_s v(s)]$ (see appendix for proofs).

3.1. sa-rectangular L_p robust MDPs

Let P_0, R_0 be any valid transition kernel and reward function, which we call nominal values. In accordance with (Derman et al., 2021), we define sa-rectangular L_p constrained uncertainty set $\mathcal{U}^{\mathtt{sa}}_p$ as

$$\mathcal{U}^{\mathtt{sa}}_p := (P_0 + \mathcal{P}) \times (R_0 + \mathcal{R})$$
$$\mathcal{R} = \left\{ R \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \mid |R(s,a)| \leq \alpha_{s,a} \right\}, \qquad \text{and}$$
$$\mathcal{P} = \{P \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} : \underbrace{\sum_{s'} P_{s,a}(s') = 0}_{\text{simplex condition}}, \|P_{s,a}\|_p \leq \beta_{s,a}\}$$

where $\mathcal{P}$, $\mathcal{R}$ are noise sets around nominal kernel P_0 and nominal reward R_0 respectively, component wise bounded by radius vectors α, β. Note that the noise sets, by construction, are sa-rectangular. To ensure, all the kernels in $(P_0 + \mathcal{P})$ are valid, we assume the radius vector β is small enough and further impose the 'simplex condition'. The condition requires all the elements of $\mathcal{P}$ to have a sum of zero across each row. Hence, it ensures each transition kernel in $(P_0 + \mathcal{P})$ has the sum of each row equal to one, as P_0 is a valid transition kernel that already has the sums to be ones.

Our setting differs from (Derman et al., 2021) as they did not impose this simplex condition on the kernel noise. It renders their setting unrealistic because the probabilities do not sum to one for some transition kernels in their uncertainty set. Imposing the simplex condition makes our reward regularizer dependent on the q-variance of the value function $\kappa_q(v)$, instead of the q-th norm of value function $\|v\|_q$ in (Derman et al., 2021). Observing that the two regularizers differ significantly can produce contrasting results when applied in practice.

Note that noise in one action is independent of the noise in other actions in any given state (Iyengar, 2005; Nilim & Ghaoui, 2005). This independence allows us to evaluate the robust Bellman operators comfortably using only nominal values and regularizers, as shown in the result below.

Theorem 3.1. *sa-rectangular L_p robust Bellman operators are equivalent to reward-value regularized (non-robust) Bellman operators:*

$$(\mathcal{T}^\pi_{\mathcal{U}^{sa}_p} v)(s) = \sum_a \pi(a|s) \Big[-\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big],$$

$$(\mathcal{T}^*_{\mathcal{U}^{sa}_p} v)(s) = \max_{a \in \mathcal{A}} \Big[-\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big],$$

where q is Holder's conjugate of p.

Proof. The proof is in the appendix, mainly consisting of two parts: a) Separating the noise from nominal values. b) The reward noise to yields the term $-\alpha_{s,a}$ and noise in kernel yields $-\gamma\beta_{s,a}\kappa_q(v)$. $\square$

The reward penalty is proportional to the uncertainty radiuses and a novel variance function $\kappa_p(v)$.

The variance regularizer $\kappa_p(v)$ penalizes the policy with high variance in their value function. In other words, the policies having low variance in their value functions are less sensitive to the perturbation of the transition kernel, and vice versa. The significant variance in value function makes the adversary more powerful, as it can choose the kernel to send the agent to relatively bad value states.

Further, we recover non-robust value iteration by putting uncertainty radiuses (i.e., $\alpha_{s,a}, \beta_{s,a}$) to zero in the above results. Furthermore, the same is true for all subsequent robust results in this paper.

Further discussion on Q-value iteration, etc, can be found in the appendix. Now, we move to $\mathtt{s}$-rectangular case, which is the core contribution of the paper.

3.2. s-rectangular L_p robust MDPs

This subsection discusses the evaluation of robust Bellman operators for the $\mathtt{s}$-rectangular uncertainty set. We begin by defining $\mathtt{s}$-rectangular L_p constrained uncertainty set $\mathcal{U}^{\mathtt{s}}_p$ as

$$\mathcal{U}^{\mathtt{s}}_p := (P_0 + \mathcal{P}) \times (R_0 + \mathcal{R})$$

where each component are bounded by L_p norm with radius vectors α and small enough β as

$$\mathcal{R} = \{R \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} : \|R(s,\cdot)\|_p \le \alpha_s\}, \qquad \text{and}$$
$$\mathcal{P} = \{P \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} : \|P_s\|_p \le \beta_s, \sum_{s'} P_s(s',a) = 0\},$$

where $P_s(s',a)$ is shorthand for $P(s'|s,a)$. Note that this setting allows the noise in one action to be coupled with other actions in a given state (Wiesemann et al., 2013). The result below shows that this coupling presents an extra dependence on the policy term in the policy evaluation for the $\mathtt{s}$-rectangular robust Bellman operator compared to the $\mathtt{sa}$-rectangular counterpart.

Theorem 3.2. *(Policy Evaluation) $\mathtt{s}$-rectangular L_p robust Bellman operator is equivalent to reward-value-policy regularized (non-robust) Bellman operator:*

$$(\mathcal{T}^{\pi}_{\mathcal{U}^{\mathtt{s}}_p} v)(s) = -\Big[\,\alpha_s + \gamma\beta_s\kappa_q(v)\,\Big]\,\|\pi_s\|_q + \sum_a \pi(a|s)\Big[\,R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')\,\Big],$$

where $\|\pi_s\|_q$ is q-norm of the vector $\pi(\cdot|s) \in \Delta_{\mathcal{A}}$.

Proof. The proof in the appendix: the techniques are similar to its $\mathtt{sa}$-rectangular counterpart. □

In this case, the reward penalty has an additional dependence on the q-norm of the policy ($\|\pi_s\|_q$). Note that $\|\pi_s\|_q$ is maximum for deterministic policies and attains the minimum value for uniform policies in state s. Hence, the policy norm term $\|\pi_s\|_q$ is conceptually similar to entropy regularization $\sum_a \pi(a|s)\ln(\pi(a|s))$, which is widely studied in the literature (Mai & Jaillet, 2021; Grill et al., 2019; Yang et al., 2019; Haarnoja et al., 2017; Schulman et al., 2017). As we can see, in the above result, the policy norm regularizer $\|\pi_s\|_q$ penalizes the deterministic policies, hence encouraging the stochasticity in the policy.

Note: It is widely believed that the stochasticity in the policy improves exploration during learning. In addition, the above result shows another benefit of policy stochasticity: they can improve robustness, leading to better generalization.

In summary, value variance and policy norm penalty terms in policy evaluation indicate that the policies with more stochasticity and low variance in their value function are more robust to environmental perturbations and vice versa.

Now, we move to policy improvement, which is challenging due to the policy regularizer term, thus presenting a richer theory.

The result below states that the optimal robust Bellman operator is the solution of a polynomial equation, which we discuss next how to obtain it.

Theorem 3.3. *(Policy improvement) For any vector v and state s, $(\mathcal{T}^{*}_{\mathcal{U}^{\mathtt{s}}_p} v)(s)$ is the minimum value of x that satisfies*

$$\Big[\sum_a \Big(\,Q(s,a) - x\,\Big)^p\,\mathbf{1}\Big(\,Q(s,a) \ge x\,\Big)\Big]^{\frac{1}{p}} = \sigma, \quad (3)$$

where $\mathbf{1}$ is indicator function, $Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$, and $\sigma = \alpha_s + \gamma\beta_s\kappa_q(v)$.

Proof. The proof is in the appendix. The main steps are: From definition and Theorem 3.2, we have

$$\begin{aligned}(\mathcal{T}^{*}_{\mathcal{U}^{\mathtt{s}}_p} v)(s) &= \max_\pi (\mathcal{T}^{\pi}_{\mathcal{U}^{\mathtt{s}}_p} v)(s)\\ &= \max_\pi \Big[\,(\mathcal{T}^{\pi}_{(P_0,R_0)} v)(s) - \Big[\,\alpha_s + \gamma\beta_s\kappa_q(v)\,\Big]\,\|\pi_s\|_q\,\Big]\\ &= \max_{\pi_s\in\Delta_{\mathcal{A}}} \langle \pi_s, Q_s\rangle - \sigma\|\pi_s\|_q \quad (\text{where } Q_s = Q(\cdot|s)).\end{aligned}$$

The solution to the above optimization problem is technically complex. Specifically, for $p = 2$, the solution is known as the water filling/pouring lemma (Anava & Levy, 2016), and we generalize it to the L_p case. □

To better understand the nature of (3), we look at the 'sub-optimality distance' function g,

$$g(x) := \Big[\sum_a \Big(\,Q(s,a) - x\,\Big)^p\,\mathbf{1}\Big(\,Q(s,a) \ge x\,\Big)\Big]^{\frac{1}{p}}.$$

Table 3. Optimal robust Bellman operator and optimal robust policy

$\mathcal{U}$	$(\mathcal{T}^*_{\mathcal{U}}v)(s)$	$\pi^*_{\mathcal{U}}(a\|s) \propto$
$\mathcal{U}^{\mathtt{s}}_p$	$\min x: \quad \Big\lVert \big(Q_s - x\mathbf{1} \big) \circ \mathbf{1} \big(Q_s \geq x \big) \Big\rVert_p = \sigma_q(v,s)$	$A(s,a)\mathbf{1}(A(s,a)\geq 0)$
$\mathcal{U}^{\mathtt{s}}_1$	$\max_k \frac{\sum_{i=1}^{k} Q(s,a_i) - \sigma_\infty(v,s)}{k}$	$\mathbf{1}(A(s,a)\geq 0)$
$\mathcal{U}^{\mathtt{s}}_2$	By algorithm 1	$A(s,a)\mathbf{1}(A(s,a)\geq 0)$
$\mathcal{U}^{\mathtt{s}}_\infty$	$\max_{a\in\mathcal{A}} Q(s,a) - \sigma_1(v,s)$	$\mathbf{1}(A(s,a) = 0)$
$\mathcal{U}^{\mathtt{sa}}_p$	$\max_{a\in\mathcal{A}} \Big[Q(s,a) - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v) \Big]$	$\mathbf{1}(A(s,a) = \max_a A(s,a))$
(P_0, R_0)	$\max_a Q(s,a)$	$\mathbf{1}(A(s,a) = 0)$

where $Q(s,a) = R_0(s,a) + \gamma P_0(\cdot|s,a)v$, sorted Q-value: $Q(s,a_1) \geq \cdots \geq Q(s,a_A)$, $\sigma_q(v,s) = \alpha_s + \gamma\beta_s\kappa_q(v)$, $Q_s = Q(s,\cdot)$, $A(s,a) = Q(s,a) - v^*_{\mathcal{U}}(s)$, $\circ$ is Hadamard product.

Observe that $g(x)$ captures the difference between x and the Q-values, summed over actions whose Q-value is greater than x. Note that the function g monotonically decreases in x as each term in the sum decreases. Further, it attains its lowest value of zero at $x = \max_a Q(s,a)$ as no action can have a Q-value greater than the maximum. On the other hand, the function value is at least σ for $x = \max_a Q(s,a) - \sigma$, as the term corresponding to the best action alone makes enough contribution.

Since, $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s)$ is the value of x at which the "sub-optimality distance" $g(x)$ is equal to the "uncertainty penalty" σ. Hence, (3) can be approximately solved using a binary search between a narrow interval $[\max_a Q(s,a) - \sigma, \quad \max_a Q(s,a)]$.

Now, we examine the following dependence of $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s)$ on p, α_s, and β_s:

- For $\alpha = \beta = 0$ then $\sigma = 0$, implying $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s) = \max_a Q(s,a)$, same as non-robust case.
- For $p = \infty$, we have, $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s) = \max_a Q(s,a) - \sigma$, same as $\mathtt{sa}$-rectangular case.
- For $p = 1$, (3) becomes linear, which can solved in closed form as $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s) = \max_k \frac{\sum_{i=1}^{k} Q(s,a_i) - \sigma}{k}$ where Q-values are sorted as $Q(s,a_1) \geq Q(s,a_2) \cdots$.
- For $p = 2$, (3) becomes quadratic, and $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s)$ can be be obtained in at-most A steps by Algorithm 1.
- As α_s and β_s increase, σ increases, resulting in a decrease in $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s)$ with a decaying rate. Moreover for sufficiently small σ, we have $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p}v)(s) = \max_a Q(s,a) - \sigma$.

Algorithm 1 $\mathtt{s}$-rectangular L_2 robust Bellman operator (see algorithm 1 of (Anava & Levy, 2016)

Input: $\sigma = \alpha_s + \gamma\beta_s\kappa_q(v), \quad k = 0, \quad \lambda = Q(s,a_1) - \sigma.$
Output: $\lambda = (\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_2}v)(s).$

```
1: Sort such that $Q(s,a_1) \geq Q(s,a_2), \cdots \geq Q(s,a_A)$.
2: while $k \leq A-1$ and $\lambda \leq Q(s,a_k)$; $k = k+1$ do
3:     $\lambda = -\sqrt{\frac{\sigma^2}{k} + (\sum_{i=1}^{k} \frac{Q(s,a_i)^2 - kQ(s,a_i)}{k^2})^2} + \frac{\sum_{i=1}^{k} Q(s,a_i)}{k}$
4: end while
```

To summarize, (3) can be solved in the closed form $p = 1, \infty$, by Algorithm 1 for $p = 2$, and approximately by binary search for general p, as summarized in table 3 with proofs in appendix.

In this section, we demonstrated that robust Bellman operators can be efficiently evaluated for both $\mathtt{sa}$ and $\mathtt{s}$ rectangular L_p robust MDPs, thus enabling efficient robust value iteration.

In the following sections, we discuss optimal policies' nature and robust value iteration's time complexity. Finally, we present experiments validating the time complexity of robust value iteration.

4. Optimal Robust Policies Characterization

Here, we study the optimal policies in $\mathtt{s}$-robust MDPs, which are known to be stochastic, in contrast to deterministic optimal policies in non-robust and $\mathtt{sa}$-rectangular robust MDPs (Wiesemann et al., 2013). Unfortunately, nothing further is known about its stochastic nature.

In the above sections, we discussed computing the robust optimal value functions. Using that robust value function, the optimal robust policies can be derived by the following relation

$$\pi^*_{\mathcal{U}} \in \arg\max_{\pi} \mathcal{T}^{\pi}_{\mathcal{U}} v^*_{\mathcal{U}}, \qquad (Wiesemannet\,al.,\ 2013).$$

This implies, the robust optimal policy $\pi^*_{\mathcal{U}}(\cdot|s)$ at state s, is the policy π that maximizes

$$\sum_a \pi(a|s) \min_{(P,R)\in\mathcal{U}} \Big[\ R(s,a) + \gamma \sum_{s'} P(s'|s,a) v^*_{\mathcal{U}}(s')\ \Big]\ .$$

For **sa-rectangular robust MDPs**, from Theorem 3.1, it is clear that a $\mathtt{sa}$-rectangular L_p robust MDP admits a deterministic optimal robust policy just like non-robust MDPs. This policy takes an action that maximizes the regularized Q-value $Q(s,a) = -\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v) + R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a) v^*_{\mathcal{U}^{\mathtt{sa}}_p}(s')$ in state s.

However, for **s-rectangular robust MDPs**, from Theorem 3.2, we get the optimal policy $\pi^*_{\mathcal{U}^{\mathtt{s}}_p}(\cdot|s)$ in state s is the maximizer of $-\ \big[\ \alpha_s + \gamma\beta_s\kappa_q(v)\ \big]\ \|\pi_s\|_q + \sum_a \pi(a|s)\ \big[\ R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a) v^*_{\mathcal{U}^{\mathtt{s}}_p}(s')\ \big]$ over π_s. The result below solves this maximization problem and provides the first explicit characterization of robust optimal policies.

Theorem 4.1. *The optimal robust policy $\pi^*_{\mathcal{U}^s_p}$ can be computed using optimal robust value function as:*

$$\pi^*_{\mathcal{U}^{\mathtt{s}}_p}(a|s) \propto [Q(s,a) - v^*_{\mathcal{U}^{\mathtt{s}}_p}(s)]^{p-1}\mathbf{1}\ \big(\ Q(s,a) \geq v^*_{\mathcal{U}^{\mathtt{s}}_p}(s)\ \big)$$

where $Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a) v^*_{\mathcal{U}^{\mathtt{s}}_p}(s)$.

Notice that the above optimal policy is a threshold policy, exclusively selecting actions with a positive advantage function. The selection is proportionate to the advantage values, assigning greater significance to actions with higher advantages while avoiding actions that offer little utility. It is worth highlighting that this policy diverges from the optimal policy in soft-Q learning with entropy regularization. In the latter, the policy follows a softmax distribution, expressed as $\pi(a|s) \propto e^{\eta(Q(a|s)-v(s))}$, where actions are selected based on a combination of the action's Q-value and the state's value (Haarnoja et al., 2017; Mai & Jaillet, 2021; Schulman et al., 2017).

Notably, the parameter p in the above robust optimal policy acts similarly to temperature variable η in soft-max policy, controlling skewness of the policy distribution over actions. The robust optimal policy and the soft-max policy move towards the best action deterministically as their temperature variables increase. For concreteness, the exceptional cases of the above theorem for $p = 1, 2, \infty$, and others are summarized in table 3.

Algorithm 2 Online $\mathtt{s}$-rectangular L_p robust value iteration

```
Input: Initialize $Q, v$ randomly, $s_0 \sim \mu$.
Output: $v = v^*_{\mathcal{U}^{\mathtt{s}}_p}$.
1: while not converged; $n = n + 1$ do
2:     Estimate $\kappa_p(v)$ using table 2. Then approximate $(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p} v)(s_n)$ using table 3 and update
           $v(s_n) = v(s_n) + \eta_n[(\mathcal{T}^*_{\mathcal{U}^{\mathtt{s}}_p} v)(s_n) - v(s_n)]$.
3:     Play action $a_n = a$ with probability proportional to
           $[Q(s_n,a) - v(s_n)]^{p-1}\mathbf{1}(Q(s_n,a) \geq v(s_n))$,
4:     Get the next state $s_{n+1}$ from the environment and update Q-value:
           $Q(s_n,a_n) = Q(s_n,a_n) + \eta'_n[R(s_n,a_n) + \gamma v(s_{n+1}) - Q(s_n,a_n)]$.
5: end while
```

	Total cost O
(P,R)	$\log(1/\epsilon)S^2A$
$\mathcal{U}^{\mathtt{sa}}_1$	$\log(1/\epsilon)S^2A$
$\mathcal{U}^{\mathtt{sa}}_2$	$\log(1/\epsilon)S^2A$
$\mathcal{U}^{\mathtt{sa}}_\infty$	$\log(1/\epsilon)S^2A$
$\mathcal{U}^{\mathtt{s}}_1$	$\log(1/\epsilon)(S^2A + SA\log(A))$
$\mathcal{U}^{\mathtt{s}}_2$	$\log(1/\epsilon)(S^2A + SA\log(A))$
$\mathcal{U}^{\mathtt{s}}_\infty$	$\log(1/\epsilon)S^2A$
$\mathcal{U}^{\mathtt{sa}}_p$	$\log(1/\epsilon)\ \big(\ S^2A + S\log(S/\epsilon)\ \big)$
$\mathcal{U}^{\mathtt{s}}_p$	$\log(1/\epsilon)\ \big(\ S^2A + SA\log(A/\epsilon)\ \big)$
$\mathcal{U}_{convex}$	Strongly NP-Hard

Table 4. Time Complexity of Robust Value Iteration

5. Complexity and Experiments

In this section, we examine the time complexity of robust value iteration: $v_{n+1} := \mathcal{T}^*_{\mathcal{U}} v_n$ for different L_p robust MDPs assuming the knowledge of nominal values (P_0, R_0). Since, the optimal robust Bellman operator $\mathcal{T}^*_{\mathcal{U}}$ is γ-contraction operator (Wiesemann et al., 2013), meaning that it requires only $O(\log(\frac{1}{\epsilon}))$ iterations to obtain an ϵ-close approximation of the optimal robust value. The main challenge is to calculate the cost of one iteration.

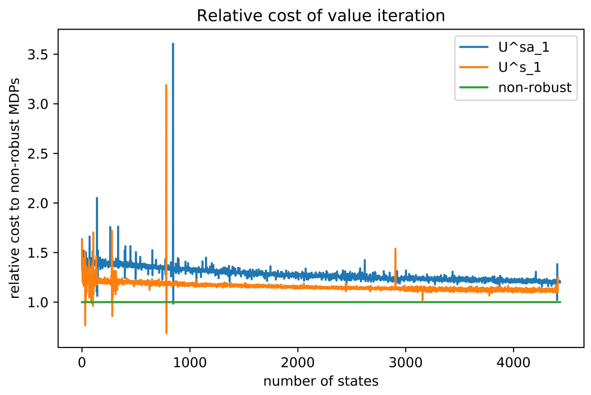

Figure 1. Relative cost of value iteration w.r.t. non-robust MDP at different S with fixed $A = 10$.

Table 5. Running cost (time) for value iteration relative w.r.t. non-robust MDP

		Linear Program		Ours			
S	A	$\mathcal{U}_1^{sa}$	$\mathcal{U}_1^{s}$	$\mathcal{U}_1^{sa}$	$\mathcal{U}_2^{sa}$	$\mathcal{U}_1^{s}$	$\mathcal{U}_2^{s}$
10	10	1438	72625	1.7	1.5	1.4	2.6
30	10	6616	629890	1.3	1.4	1.5	2.8
50	10	6622	4904004	1.5	1.9	1.2	2.4
100	20	16714	N/A	1.4	1.5	1.1	2.1

Evaluating the optimal robust Bellman operators in Theorem 3.1 and Theorem 3.3 has three main components. A) Computing $\kappa_p(v)$, which can be done differently depending on the value of p, as shown in Table 2. B) Computing the Q-value from v, which requires $O(S^2A)$ in all cases. Finally, C) Evaluating optimal robust Bellman operators from Q-values requires different operations, such as sorting the Q-value, calculating the best action, and performing a binary search, as shown in Table 3. The overall complexity of the evaluation is presented in Table 4, with the proofs provided in the Appendix M.

Table 4 and Figure 1 demonstrate the relative cost (time) of robust value iteration compared to non-robust MDP for randomly generated kernel and reward functions with varying numbers of states S and actions A. The Appendix H shows more results and details.

For large state spaces ($S \geq A\log(A\epsilon^{-1})$), the computational complexities for each robust MDP in Table 4 is the same as non-robust MDPs ($O(S^2A\log\epsilon^{-1})$), which is also confirmed empirically in Figure 1.

Further, Table 5 presents empirical relative time w.r.t. non-robust MDPs for value iteration using our methods and LP (linear programming). Our method scales very well with state and action spaces, while LP methods become primitively expensive in large state-action spaces.

All experiments are repeated 500 times, resulting in some stochasticity in the results. However, the standard deviations were found to be 1-10%, making the trend clear.

6. Discussion

We presented an efficient robust value iteration for s-rectangular L_p-robust MDPs, which is not only faster than existing methods but also can be easily adapted to an online setting, as illustrated by Algorithm 2. Note that the algorithm is a two-time-scale algorithm, where the Q-values are approximated at a faster time scale, and the value function is approximated from the Q-values at a slower time scale. The p-variance function κ_p can be estimated online using batches or other sophisticated methods. The algorithm's convergence can be guaranteed from (Borkar, 2022).

We also introduce a novel value regularizer (κ_p) and a novel threshold policy, which may help obtain more robust and generalizable policies.

We consider uncertainty sets constrained by the L_p norm, which is an essential and natural family of norms to consider. Moreover, adaptation to a general norm for sa-rectangular case is trivial (only κ changes in Theorem 3.1) but challenging for s-rectangular case, as discussed in the appendix.

Finally, a take-home message is that the policies with low variance in their value function are more robust, which can be ensured by using a value-variance regularizer. Further, the work reinforces the existing belief that stochastic policies are better.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement

This research was partially supported by Israel PBC-VATAT, by the Technion Artificial Intelligent Hub (Tech.AI) and by the Israel Science Foundation (grant No. 447/20).

Additionally, this project has received funding from the European Union's Horizon Europe Programme under grant agreement No.101070568.

References

Abdullah, M. A., Ren, H., Ammar, H. B., Milenkovic, V., Luo, R., Zhang, M., and Wang, J. Wasserstein robust reinforcement learning, 2019. URL `https://arxiv.org/abs/1907.13196`.

Anava, O. and Levy, K. k*-nearest neighbors: From global to local. *Advances in neural information processing systems*, 29, 2016.

Asadi, M., Talebi, M. S., Bourel, H., and Maillard, O. Model-based reinforcement learning exploiting state-action equivalence. *CoRR*, abs/1910.04077, 2019. URL `http://arxiv.org/abs/1910.04077`.

Auer, P. and Ortner, R. Logarithmic online regret bounds for undiscounted reinforcement learning. In Schölkopf, B., Platt, J., and Hoffman, T. (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper/2006/file/c1b70d965ca504aa751ddb62ad69c63f-Paper.pdf`.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper/2008/file/e4a6222cdb5b34375400904f03d8e6a5-Paper.pdf`.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 263–272. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/azar17a.html`.

Bagnell, J. A., Ng, A. Y., and Schneider, J. G. Solving uncertain markov decision processes. Technical report, Carnegie Mellon University, 2001.

Borkar, V. S. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency, 2022. doi: 10.1007/978-81-951961-1-1. URL `https://doi.org/10.1007\%2F978-81-951961-1-1`.

Derman, E. and Mannor, S. Distributional robustness and regularization in reinforcement learning, 2020. URL `https://arxiv.org/abs/2003.02894`.

Derman, E., Mankowitz, D. J., Mann, T. A., and Mannor, S. Soft-robust actor-critic policy-gradient, 2018.

Derman, E., Geist, M., and Mannor, S. Twice regularized mdps and the equivalence between robustness and regularization, 2021.

Eysenbach, B. and Levine, S. Maximum entropy rl (provably) solves some robust rl problems, 2021. URL `https://arxiv.org/abs/2103.06257`.

Gadot, U., Derman, E., Kumar, N., Elfatihi, M. M., Levy, K., and Mannor, S. Solving non-rectangular reward-robust mdps via frequency regularization, 2023.

Goyal, V. and Grand-Clément, J. Robust markov decision process: Beyond rectangularity, 2018. URL `https://arxiv.org/abs/1811.00215`.

Grill, J.-B., Darwiche Domingues, O., Menard, P., Munos, R., and Valko, M. Planning in entropy-regularized markov decision processes and games. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/50982fb2f2cfa186d335310461dfa2be-Paper.pdf`.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies, 2017. URL `https://arxiv.org/abs/1702.08165`.

Hanasusanto, G. A. and Kuhn, D. Robust data-driven dynamic programming. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/ef575e8837d065a1683c022d2077d342-Paper.pdf`.

Ho, C. P., Petrik, M., and Wiesemann, W. Partial policy iteration for l1-robust markov decision processes, 2020. URL `https://arxiv.org/abs/2006.09484`.

Husain, H., Ciosek, K., and Tomioka, R. Regularized policies are reward robust, 2021. URL `https://arxiv.org/abs/2101.07012`.

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, May 2005. ISSN 1526-5471. doi: 10.1287/moor.1040.0129. URL `http://dx.doi.org/10.1287/MOOR.1040.0129`.

Kaufman, D. L. and Schaefer, A. J. Robust modified policy iteration. *INFORMS J. Comput.*, 25:396–410, 2013.

Kumar, N., Derman, E., Geist, M., Levy, K. Y., and Mannor, S. Policy gradient for rectangular robust markov decision processes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=NLpXRrjpa6`.

Mai, T. and Jaillet, P. Robust entropy-regularized markov decision processes, 2021. URL `https://arxiv.org/abs/2112.15364`.

Mannor, S., Simester, D., Sun, P., and Tsitsiklis, J. N. Bias and variance in value function estimation. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pp. 72, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015402. URL https://doi.org/10.1145/1015330.1015402.

Mannor, S., Mebel, O., and Xu, H. Robust mdps with k-rectangular uncertainty. *Math. Oper. Res.*, 41(4): 1484–1509, nov 2016. ISSN 0364-765X.

Nilim, A. and Ghaoui, L. E. Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.*, 53:780–798, 2005.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., and Song, D. Assessing generalization in deep reinforcement learning, 2018. URL https://arxiv.org/abs/1810.12282.

Puterman, M. L. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft q-learning, 2017. URL https://arxiv.org/abs/1704.06440.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Tamar, A., Mannor, S., and Xu, H. Scaling up robust mdps using function approximation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pp. 181–189. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/tamar14.html.

Wang, K., Gadot, U., Kumar, N., Levy, K., and Mannor, S. Robust reinforcement learning via adversarial kernel approximation, 2023.

Wang, Y. and Zou, S. Online robust reinforcement learning with model uncertainty, 2021. URL https://arxiv.org/abs/2109.14523.

Wang, Y. and Zou, S. Policy gradient method for robust reinforcement learning, 2022.

Wiesemann, W., Kuhn, D., and Rustem, B. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/23358653.

Xu, H. and Mannor, S. Robustness and generalization, 2010. URL https://arxiv.org/abs/1005.2243.

Yang, W., Li, X., and Zhang, Z. A regularized approach to sparse optimal policy in reinforcement learning. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5938–5948, 2019.

Zhao, C., Sigaud, O., Stulp, F., and Hospedales, T. M. Investigating generalisation in continuous deep reinforcement learning, 2019. URL https://arxiv.org/abs/1902.07015.

Zhou, R., Liu, T., Cheng, M., Kalathil, D., Kumar, P., and Tian, C. Natural actor-critic for robust reinforcement learning with function approximation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=wxkBdtDbmH.

A. How to read appendix

1. Section B contains additional properties that couldn't be included in the main section for the sake of clarity and space.
2. Section C contains the discussion on zero transition kernel (forbidden transitions).
3. Section D contains a possible connection this work to UCRL.
4. Section H contains additional experimental results and a detailed discussion.
5. All the proofs of the main body of the paper is presented in the section L and M.
6. Section J contains helper results for section L. Particularly, it discusses p-mean function ω_p and p-variance function κ_p.
7. Section K contains helper results for section L. Particularly, it discusses L_p water pouring lemma, necessary to evaluate robust optimal Bellman operator (learning) for s-rectangular L_p robust MDPs.
8. Section M contains time complexity proof for model based algorithms.
9. Section E develops Q-learning machinery for (sa)-rectangular L_p robust MDPs based on the results in the main section. It is not used in the main body or anywhere else, but this provides a good understanding for algorithms proposed in section F for (sa)-rectangular case.
10. Section F contains model-based algorithms for s and (sa)-rectangular L_p robust MDPs. It also contains, remarks for special cases for $p = 1, 2, \infty$.

B. Revisiting S-rectangular Robust MDPs

Here we outline some intriguing properties of s-rectangular robust MDPs, particularly their Q-value and greedy policies. We begin with some useful definitions in Table 6.

Table 6. Useful Definitions

Notation	Definition	Remark
Q^v	$R_0 + \gamma P_0 v$	Q-value at value function v
$Q^*_{\mathcal{U}}$	$R_0 + \gamma P_0 v^*_{\mathcal{U}}$	Optimal Q-value
$\pi^v_{\mathcal{U}}$	$\arg\max_\pi \mathcal{T}^\pi_{\mathcal{U}} v$	Greedy policy at value function v
$\chi_p(s)$	$\left\lvert \{a \mid \pi^*_{\mathcal{U}^s_p}(a\|s) \geq 0\} \right\rvert = \left\lvert \{a \mid Q^*_{\mathcal{U}^s_p}(s,a) \geq v^*_{\mathcal{U}^s_p}(s)\} \right\rvert$	Number of active actions in state s of optimal policy (From Theorem 4.1)
$\chi_p(v,s)$	$\left\lvert \{a \mid \pi^v_{\mathcal{U}^s_p}(a\|s) \geq 0\} \right\rvert$	Number of active actions in state s, of greedy policy at value function v
$Q(s,a_i)$	$Q(s,a_1) \geq Q(s,a_2) \geq, \cdots, \geq Q(s,a_A)$	i-th best Q-value in state s.

B.1. Relation Between Optimal Q-value and Optimal Value function

The optimal value function and optimal Q-value, in non-robust MDPs, have a very straightforward relation,

$$v^*_{(P,R)}(s) = \max_a Q^*_{(P,R)}(s,a), \qquad \forall s.$$

This relation simply extends to sa-rectangular case, where the value function is best regularized Q-value, that is,

$$v^*_{\mathcal{U}^{sa}_p}(s) = \max_a [\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v^*_{\mathcal{U}^{sa}_p}) - Q^*_{\mathcal{U}^{sa}_p}(s,a)]$$

However, in s-rectangular case, the value function is no longer given by the best action. Specifically, from Theorem 3.2, we have the following relation,

$$v^*_{\mathcal{U}^s_p}(s) = \sum_a \pi^*_{\mathcal{U}^s_p}(a|s) \Big[-\big(\alpha_s + \gamma\beta_s\kappa_q(v) \big) \|\pi^*_{\mathcal{U}^s_p}(\cdot|s)\|_q + Q^*_{\mathcal{U}^s_p}(s,a) \Big] . \tag{4}$$

From this relation, we get the value function sandwiched between the Q-value of lowest active action and highest inactive Q-value, as stated by the property below.

Property B.1. *(Optimal Value vs Q-value) $v^*_{\mathcal{U}^s_p}(s)$ is bounded by the Q-value of $\chi_p(s)$th and $(\chi_p(s)+1)$th actions, that is ,*

$$Q^*_{\mathcal{U}^s_p}(s, a_{\chi_p(s)+1}) < v^*_{\mathcal{U}^s_p}(s) \leq Q^*_{\mathcal{U}^s_p}(s, a_{\chi_p(s)}).$$

Proof. A special case of Property B.3. □

The above sandwich property is surprising and novel. The relation of various robust Q-value and value functions is summarized in the table 7.

Table 7. Optimal value function and Q-value

	Q-value and value relation	Remark
(P,R)	$v^*_{(P,R)}(s) = \max_a Q^*_{(P,R)}(s,a)$	Best value
$\mathcal{U}^{sa}_p$	$v^*_{\mathcal{U}^{sa}_p}(s) = \max_a[\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v^*_{\mathcal{U}^{sa}_p}) - Q^*_{\mathcal{U}^{sa}_p}(s,a)]$	Best regularized value
$\mathcal{U}^s_p$	$Q^*_{\mathcal{U}^s_p}(s, a_{\chi_p(s)+1}) < v^*_{\mathcal{U}^s_p}(s) \leq Q^*_{\mathcal{U}^s_p}(s, a_{\chi_p(s)})$	Sandwich between Q-value of best inactive action and worst active action

where (P, R) denotes the kernel and reward function for non-robust MDP.

B.2. Greedy Policy and Q-value

The greedy policy has the same form as the optimal robust policy described in Theorem 4.1. We briefly present the results in this subsection for the sake of completeness.

Theorem B.2. *(Greedy policy) The greedy policy $\pi^v_{\mathcal{U}^s_p}$ is a threshold policy, that is proportional to the advantage function, that is*

$$\pi^v_{\mathcal{U}^s_p}(a|s) \propto \big(Q^v(s,a) - (\mathcal{T}^*_{\mathcal{U}^s_p} v)(s) \big)^{p-1} \mathbf{1} \big(Q^v(s,a) \geq (\mathcal{T}^*_{\mathcal{U}^s_p} v)(s) \big) .$$

Proof. Proof in section L. □

Note that the Theorem 4.1 is a special case of the above theorem. The greedy policies for various robust MDPs are summarized in Table 8.

Table 8. Greedy policy at value function v

$\mathcal{U}$	$\pi^v_{\mathcal{U}}(a\|s) \propto$	remark
$\mathcal{U}^s_p$	$(Q^v(s,a) - (\mathcal{T}^*_{\mathcal{U}}v)(s))^{p-1}\mathbf{1}(A^v_{\mathcal{U}}(s,a) \geq 0)$	top actions proportional to $(p-1)$th power of its advantage
$\mathcal{U}^s_1$	$\frac{\mathbf{1}(A^v_{\mathcal{U}}(s,a)\geq 0)}{\sum_a \mathbf{1}(A^v_{\mathcal{U}}(s,a)\geq 0)}$	top actions with uniform probability
$\mathcal{U}^s_2$	$\frac{A^v_{\mathcal{U}}(s,a)\mathbf{1}A^v_{\mathcal{U}}(s,a)\geq 0)}{\sum_a A^v_{\mathcal{U}}(s,a)\mathbf{1}(A^v_{\mathcal{U}}(s,a)\geq 0)}$	top actions proportion to its advantage
$\mathcal{U}^s_\infty$	$\arg\max_{a\in\mathcal{A}} Q^v(s,a)$	best action
$\mathcal{U}^{\mathsf{sa}}_p$	$\arg\max_a[-\alpha_{sa} - \gamma\beta_{sa}\kappa_q(v) + Q^v(s,a)]$	best action

where $A^v_{\mathcal{U}}(s,a) = Q^v(s,a) - (\mathcal{T}^*_{\mathcal{U}}v)(s)$ and $Q^v(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$.

The above result states that the greedy policy takes actions having a non-negative advantage, so we have.

$$\chi_p(v,s) := \left|\left\{ a \mid \pi^v_{\mathcal{U}^s_p}(a|s) \geq 0 \right\}\right| = \left|\left\{ a \mid Q^v(s,a) \geq (\mathcal{T}^*_{\mathcal{U}^s_p})v(s) \right\}\right| . \tag{5}$$

The property below states that the $T^*_{\mathcal{U}}v$ is sandwiched between the lowest Q-value of active action and the highest Q-value of inactive action.

Property B.3. *(Greedy Value vs Q-value) $(\mathcal{T}^*_{\mathcal{U}^s_p}v)(s)$ is bounded by the Q-value of $\chi_p(v,s)$th and $(\chi_p(v,s)+1)$th actions, that is ,*

$$Q^v(s, a_{\chi_p(v,s)+1}) < (\mathcal{T}^*_{\mathcal{U}^s_p}v)(s) \leq Q^v(s, a_{\chi_p(v,s)}).$$

Proof. Proof in section L. □

Table 9. Greedy value function and Q-value

MDP	Relation	Remark
(P,R)	$(\mathcal{T}^*_{(P,R)}v)(s) = \max_a Q^v(s,a)$	Best value
$\mathcal{U}^s_p$	$(\mathcal{T}^*_{\mathcal{U}^{sa}_p})v(s) = \max_a[\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v) - Q^v(s,a)]$	Best regularized value
$\mathcal{U}^s_p$	$Q^v(s, a_{\chi_p(v,s)+1}) < (\mathcal{T}^*_{\mathcal{U}^s_p})v(s) \leq Q^v(s, a_{\chi_p(v,s)})$	Sandwich!

where $Q^v(s,a_1) \geq, \cdots, \geq Q^v(s,a_A)$.

C. Revisiting kernel noise assumption

Until now, we implicitly assumed that the nominal kernel has support over all states in every state and action, that is $P_0(s'|s,a) > 0$ for all $s', s \in \mathcal{S}, a \in \mathcal{A}$. Now we relax this condition, and we focus on the cases where $P_0(s'|s,a) = 0$ for some states s', which we call forbidden transitions. This enables us to capture many practical situations where in a given state, many transitions are impossible. For example, consider a grid world example where only single-step jumps (left, right, up, down) are allowed, so in this case, the probability of making a multi-step jump is impossible. So upon adding noise to the kernel, the system should not start making impossible transitions. Therefore, noise set $\mathcal{P}$ must satisfy additional constraint: For any (s,a) if $P_0(s'|s,a) = 0$ then

$$P(s'|s,a) = 0, \quad \forall P \in \mathcal{P}.$$

Incorporating this constraint without much change in the theory is one of our novel contributions, and is discussed below.

C.1. Sa-Rectangular Uncertainty

Let $F_{s,a} := \{s' \mid P_0(s'|s,a) = 0\}$ be the set states where transition is impossible at state s under action a. Further, we want to have the transition uncertainty set $\mathcal{P}$ respecting this constraint, that is

$$P(s'|s,a) = 0, \quad \forall P \in \mathcal{P}, \forall s' \in F_{s,a}. \tag{6}$$

Formally, we define, the kernel noise set as

$$\mathcal{P} = \{P \mid \|P(\cdot|s,a)\|_p = \beta_{s,a}, \quad \underbrace{\sum_{s'} P(s'|s,a) = 0,}_{\text{simplex condition}} \quad \underbrace{P(s"|s,a) = 0, \forall s" \in F_{s,a}}_{\text{forbidden state condition}}\}. \tag{7}$$

In this case, our p-variance function is redefined as

$$\begin{aligned}\kappa_p(v,s,a) &= \min_{\|P\|_p=\beta_{s,a}, \quad \sum_{s'} P(s')=0, \quad P(s")=0, \quad \forall s" \in F_{s,a}} \langle P, v\rangle && (8)\\ &= \min_{\omega\in\mathbb{R}} \|u - \omega \mathbf{1}\|_p, \qquad \text{where } u(s) = v(s)\mathbf{1}(s \notin F_{s,a}). && (9)\\ &= \kappa_p(u) && (10)\end{aligned}$$

This says we consider the value of only those states that are allowed (not forbidden) in the calculation of p-variance. For example, we have

$$\kappa_\infty(v,s,a) = \frac{\max_{s\notin F_{s,a}} v(s) - \min_{s\notin F_{s,a}} v(s)}{2}. \tag{11}$$

$$\tag{12}$$

So theorem 1 of the main paper can be re-stated as

Theorem C.1. *(Restated)* $(\texttt{Sa})$*-rectangular L_p robust Bellman operator is equivalent to reward regularized (non-robust) Bellman operator. That is, using κ_p above, we have*

$$\begin{aligned}(\mathcal{T}^\pi_{\mathcal{U}^{\mathtt{sa}}_p} v)(s) &= \sum_a \pi(a|s)[-\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v,s,a) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s')],\\ (\mathcal{T}^*_{\mathcal{U}^{\mathtt{sa}}_p} v)(s) &= \max_{a\in\mathcal{A}}[-\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v,s,a) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s')].\end{aligned}$$

C.2. S-Rectangular Uncertainty

This notion can also be applied to $\texttt{s}$-rectangular uncertainty, but with a little caution. Here, we define forbidden states in state s to be F_s (state dependent) instead of state-action dependent in $\texttt{sa}$-rectangular case. Here, we define p-variance as

$$\kappa_p(v,s) = \kappa_p(u), \qquad \text{where } u(s) = v(s)\mathbf{1}(s \notin F_s). \tag{13}$$

So Theorem 2 can be restated as

Theorem C.2. *(restated) (Policy Evaluation) $\texttt{S}$-rectangular L_p robust Bellman operator is equivalent to reward regularized (non-robust) Bellman operator, that is*

$$(\mathcal{T}^\pi_{\mathcal{U}^{s}_p} v)(s) = -\Big(\alpha_s + \gamma\beta_s\kappa_q(v,s)\Big)\|\pi(\cdot|s)\|_q + \sum_a \pi(a|s)\Big(R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')\Big)$$

where κ_p is defined above and $\|\pi(\cdot|s)\|_q$ is q-norm of the vector $\pi(\cdot|s) \in \Delta_{\mathcal{A}}$.

For all the other results (including theorem 4), we just need to replace the old p-variance function with the new p-variance function appropriately.

D. Application to UCRL

Here, we show UCRL (upper confidence reinforcement learning) although very different in motivation than robust MDPs, can benefit from our techniques developed so far.

Optimistic policies are sought that aids exploration in UCRL (Auer & Ortner, 2006; Auer et al., 2008; Asadi et al., 2019), in contrast to robust MDPs that seek pessimistic policies to avoid the risk. Referring to step 3 of the UCRL algorithm (Auer & Ortner, 2006), the objective is to find a policy

$$\arg\max_{\pi}\max_{R,P\in\mathcal{U}}\langle\mu, v^{\pi}_{P,R}\rangle, \tag{14}$$

where

$$\mathcal{U}=\{(R,P)\mid |R(s,a)-R_0(s,a)|\leq\alpha_{s,a}, |P(s'|s,a)-P_0(s'|s,a)|\leq\beta_{s,a,s'}, P\in(\Delta_{\mathcal{S}})^{\mathcal{S}\times\mathcal{A}}\}$$

for current estimated kernel P_0 and reward function R_0. We refer to Section 3.1.1 and step 4 of the UCRL 2 algorithm of (Auer et al., 2008), which seeks to find whose best performance is high, that is,

$$\max_{\pi}\max_{R,P\in\mathcal{U}}\langle\mu, v^{\pi}_{P,R}\rangle, \tag{15}$$

where

$$\mathcal{U}=\{(R,P)\mid |R(s,a)-R_0(s,a)|\leq\alpha_{s,a}, \lVert P(\cdot|s,a)-P_0(\cdot|s,a)\rVert_1\leq\beta_{s,a}, P\in(\Delta_{\mathcal{S}})^{\mathcal{S}\times\mathcal{A}}\}$$

The uncertainty radius α,β depends on the number of samples of different transitions and observations of the reward. The paper (Auer & Ortner, 2006) doesn't explain any method to solve the above problem. UCRL 2 algorithm (Auer et al., 2008), suggests solving it by linear programming that can be very slow. We show that it can be solved by our methods.

The above problem can be tackled as following

$$\max_{\pi}\max_{R,P\in\mathcal{U}^{sa}_{p}}\langle\mu, v^{\pi}_{P,R}\rangle. \tag{16}$$

We can define, optimistic Bellman operators as

$$\hat{\mathcal{T}}^{\pi}_{\mathcal{U}}v:=\max_{R,P\in\mathcal{U}}v^{\pi}_{P,R},\qquad \hat{\mathcal{T}}^{*}_{\mathcal{U}}v:=\max_{\pi}\max_{R,P\in\mathcal{U}}v^{\pi}_{P,R}. \tag{17}$$

The well definition and contraction of the above optimistic operators may follow directly from their pessimistic (robust) counterparts. We can evaluate the above optimistic operators as

$$(\hat{\mathcal{T}}^{\pi}_{\mathcal{U}^{sa}_{p}}v)(s)=\sum_{a}\pi(a|s)\Big[\,R_0(s,a)+\alpha_{s,a}+\beta_{s,a}\gamma\kappa_q(v)+\sum_{s'}P_0(s'|s,a)v(s')\,\Big], \tag{18}$$

$$(\hat{\mathcal{T}}^{*}_{\mathcal{U}^{sa}_{p}}v)(s)=\max_{a}\Big[\,R_0(s,a)+\alpha_{s,a}+\beta_{s,a}\gamma\kappa_q(v)+\sum_{s'}P_0(s'|s,a)v(s')\,\Big]\,. \tag{19}$$

The uncertainty radiuses α,β and nominal values P_0,R_0 can be found by similar analysis by (Auer & Ortner, 2006; Auer et al., 2008). We can get the Q-learning from the above results as

$$Q(s,a)\to R_0(s,a)-\alpha_{s,a}-\gamma\beta_{s,a}\kappa_q(v)+\gamma\sum_{s'}P_0(s'|s,a)\max_{a'}Q(s',a'), \tag{20}$$

where $v(s)=\max_a Q(s,a)$. From the law of large numbers, we know that uncertainty radiuses $\alpha_{s,a},\beta_{s,a}$ behaves as $O(\frac{1}{\sqrt{n}})$ asymptotically with number of iteration n. This resembles very closely to the UCB VI algorithm (Azar et al., 2017). We emphasize that similar optimistic operators can be defined and evaluated for s-rectangular uncertainty sets too.

E. Q-Learning for $(\mathtt{sa})$-rectangular MDPs

Here we generalize our result and obtain Q-learning for $\mathtt{sa}$-rectangular robust MDPs.

In view of Theorem 3.1, we can define $Q^{\pi}_{\mathcal{U}^{\mathtt{sa}}_p}$, the robust Q-values under policy π for $(\mathtt{sa})$-rectangular L_p constrained uncertainty set $\mathcal{U}^{\mathtt{sa}}_p$ as

$$Q^{\pi}_{\mathcal{U}^{\mathtt{sa}}_p}(s,a) := -\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v^{\pi}_{\mathcal{U}^{\mathtt{sa}}_p}) + R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v^{\pi}_{\mathcal{U}^{\mathtt{sa}}_p}(s'). \tag{21}$$

This implies that we have the following relation between robust Q-values and robust value function, same as its non-robust counterparts,

$$v^{\pi}_{\mathcal{U}^{\mathtt{sa}}_p}(s) = \sum_a \pi(a|s)Q^{\pi}_{\mathcal{U}^{\mathtt{sa}}_p}(s,a). \tag{22}$$

Let $Q^{*}_{\mathcal{U}^{\mathtt{sa}}_p}$ denote the optimal robust Q-values associated with optimal robust value $v^{*}_{\mathcal{U}^{\mathtt{sa}}_p}$, given as

$$Q^{*}_{\mathcal{U}^{\mathtt{sa}}_p}(s,a) := -\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v^{*}_{\mathcal{U}^{\mathtt{sa}}_p}) + R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v^{*}_{\mathcal{U}^{\mathtt{sa}}_p}(s'). \tag{23}$$

It is evident from Theorem 3.1 that optimal robust value and optimal robust Q-values satisfy the following relation, same as its non-robust counterparts,

$$v^{*}_{\mathcal{U}^{\mathtt{sa}}_p}(s') = \max_{a\in\mathcal{A}} Q^{*}_{\mathcal{U}^{\mathtt{sa}}_p}(s,a). \tag{24}$$

Combining 24 and 23, we have optimal robust Q-value recursion as follows

$$Q^{*}_{\mathcal{U}^{\mathtt{sa}}_p}(s,a) = -\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v^{*}_{\mathcal{U}^{\mathtt{sa}}_p}) + R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)\max_{a\in\mathcal{A}} Q^{*}_{\mathcal{U}^{\mathtt{sa}}_p}(s,a). \tag{25}$$

The above robust Q-value recursion enjoys similar properties as its non-robust counterparts.

Corollary E.1. *($(\mathtt{sa})$-rectangular L_p regularized Q-learning) Let*

$$Q_{n+1}(s,a) = R_0(s,a) - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v_n) + \gamma\sum_{s'} P_0(s'|s,a)\max_{a\in\mathcal{A}} Q_n(s',a),$$

where $v_n(s) = \max_{a\in\mathcal{A}} Q_n(s,a)$, then Q_n converges to $Q^{*}_{\mathcal{U}^{\mathtt{sa}}_p}$ linearly.

Observe that the above Q-learning equation is the same as non-robust MDP except for the reward penalty. Recall that $\kappa_1(v) = 0.5(\max_s v(s) - \min_s v(s))$ is difference between peak to peak values and $\kappa_2(v)$ is variance of v, that can be easily estimated. Hence, model-free algorithms for $(\mathtt{sa})$-rectangular L_p robust MDPs for $p = 1, 2$, can be derived easily from the above results. This implies that $(\mathtt{sa})$-rectangular L_1 and L_2 robust MDPs are as easy as non-robust MDPs.

F. Model Based Algorithms

In this section, we assume that we know the nominal transitional kernel and nominal reward function. Algorithm 3, algorithm 4 is model based algorithm for $(\mathtt{sa})$-rectangular and $\mathtt{s}$ rectangular L_p robust MDPs respectively. It is explained in the algorithms, how to get deal with special cases $(p = 1, 2, \infty)$ in an easy way.

Algorithm 3 Model Based Q-Learning Algorithm for sa-rectangular L_p Robust MDP

```
1: Input: $\alpha_{s,a}, \beta_{s,a}$ are uncertainty radius in reward and transition kernel respectively in state s and action $a$. Transition kernel $P$ and reward vector $R$. Take initial $Q$-values $Q_0$ randomly and $v_0(s) = \max_a Q_0(s,a)$.
2: while not converged do
3:     Do binary search in $[\min_s v_n(s), \max_s v_n(s)]$ to get $q$-mean $\omega_n$, such that
           $$\sum_s \frac{(v_n(s)-\omega_n)}{|v_n(s)-\omega_n|}|v_n(s)-\omega_n|^{\frac{1}{p-1}} = 0. \quad (26)$$
4:     Compute $q$-variance:    $\kappa_n = \|v-\omega_n\|_q$.
5:     Note: For $p = 1, 2, \infty$, we can compute $\kappa_n$ exactly in closed from, see table 2.
6:     for $s \in \mathcal{S}$ do
7:         for $a \in \mathcal{A}$ do
8:             Update Q-value as
                   $$Q_{n+1}(s,a) = R_0(s,a) - \alpha_{sa} - \gamma\beta_{sa}\kappa_n + \gamma\sum_{s'} P_0(s'|s,a)\max_a Q_n(s',a).$$
9:         end for
10:        Update value as
                   $$v_{n+1}(s) = \max_a Q_{n+1}(s,a).$$
11:    end for
                   $$n \to n+1$$
12: end while
```

Algorithm 4 Model Based Algorithm for S rectangular L_p Robust MDP

```
1: Take initial $Q$-values $Q_0$ and value function $v_0$ randomly.
2: Input: $\alpha_s, \beta_s$ are uncertainty radius in reward and transition kernel respectively in state s.
3: while not converged do
4:     Do binary search in $[\min_s v_n(s), \max_s v_n(s)]$ to get $q$-mean $\omega_n$, such that
           $$\sum_s \frac{(v_n(s) - \omega_n)}{|v_n(s) - \omega_n|}|v_n(s) - \omega_n|^{\frac{1}{p-1}} = 0. \quad (27)$$
5:     Compute $q$-variance:    $\kappa_n = \|v - \omega_n\|_q$.
6:     Note: For $p = 1, 2, \infty$, we can compute $\kappa_n$ exactly in closed from, see table 2.
7:     for $s \in \mathcal{S}$ do
8:         for $a \in \mathcal{A}$ do
9:             Update Q-value as
                   $$Q_{n+1}(s, a) = R_0(s, a) + \gamma \sum_{s'} P_0(s'|s, a)v_{n+1}(s'). \quad (28)$$
10:        end for
11:        Sort actions in decreasing order of the Q-value, that is
               $$Q_{n+1}(s, a_i) \geq Q_{n+1}(s, a_{i+1}). \quad (29)$$
12:        Value evaluation:
               $$v_{n+1}(s) = x \quad \text{such that} \quad (\alpha_s + \gamma\beta_s\kappa_n)^p = \sum_{Q_{n+1}(s,a_i) \geq x} |Q_{n+1}(s, a_i) - x|^p. \quad (30)$$
13:        Note: We can compute $v_{n+1}(s)$ exactly in closed from for $p = \infty$ and for $p = 1, 2$, we can do the same using algorithm 8,7 respectively, see table 3.
14:    end for
           $$n \to n + 1$$
15: end while
```

Algorithm 5 Model based algorithm for s-rectangular L_1 robust MDPs

```
1: Take initial value function $v_0$ randomly and start the counter $n = 0$.
2: while not converged do
3:   Calculate $q$-variance:    $\kappa_n = \frac{1}{2}\left[\ \max_s v_n(s) - \min_s v_n(s)\ \right]$
4:   for $s \in \mathcal{S}$ do
5:     for $a \in \mathcal{A}$ do
6:       Update Q-value as
           $$Q_n(s,a) = R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v_n(s'). \tag{31}$$
7:     end for
8:     Sort actions in state $s$, in decreasing order of the Q-value, that is
           $$Q_n(s,a_1) \geq Q_n(s,a_2), \cdots \geq Q_n(s,a_A). \tag{32}$$
9:     Value evaluation:
           $$v_{n+1}(s) = \max_m \frac{\sum_{i=1}^m Q_n(s,a_i) - \alpha_s - \beta_s\gamma\kappa_n}{m}. \tag{33}$$
10:    Value evaluation can also be done using algorithm 8.
11:  end for
           $$n \to n+1$$
12: end while
```

G. Generalization to General Norm

Until now, we studied uncertainty sets that are constrained by L_p norm. Here, we discuss how we can generalize our techniques to any general norm.

Specifically, we consider sa-rectangular uncertainty set $\mathcal{U} = \mathcal{U}^{\mathtt{sa}}_{\|\cdot\|}$ constrained by $\|\cdot\|$ norm, defined as

$$\mathcal{U}^{\mathtt{sa}}_{\|\cdot\|} = (P_0 + \mathcal{P}) \times (R_0 + \mathcal{R}), \qquad \text{where} \qquad (\mathcal{P}, \mathcal{R}) = (\times_{s,a}\mathcal{P}_{sa}, \times_{s,a}\mathcal{P}_{sa}),$$

$$\mathcal{R}_{(s,a)} = \{r \in \mathbb{R} \mid \|r\| \leq \alpha_{s,a}\}, \quad \text{and} \quad \mathcal{P}_{(s,a)} = \left\{p \in \mathbb{R}^{\mathcal{S}} \mid \langle p, \mathbf{1}\rangle_{\mathcal{S}} = 0, \|p\| \leq \beta_{s,a}\right\}.$$

The robust Bellman operator $\mathcal{T}^{\pi}_{\mathcal{U}}$ and optimal robust Bellman operator $\mathcal{T}^{*}_{\mathcal{U}}$ can be evaluated as

$$(\mathcal{T}^{\pi}_{\mathcal{U}}v)(s) = \sum_a \pi(a|s)\left[\ R(s,a) - \gamma\beta_{s,a}\kappa_{\|\cdot\|}(v) + \gamma\sum_{s'} P(s'|s,a)v(s')\ \right],$$

$$(\mathcal{T}^{*}_{\mathcal{U}}v)(s) = \max_a \left[\ R(s,a) - \gamma\beta_{s,a}\kappa_{\|\cdot\|}(v) + \gamma\sum_{s'} P(s'|s,a)v(s')\ \right],$$

where variance function is defined as

$$\kappa_{\|\cdot\|}(v) := \min_{\langle u,\mathbf{1}\rangle_{\mathcal{S}}=0, \|u\|\leq 1} \langle u, v^{\pi}_{\mathcal{U}}\rangle.$$

Since norms are convex functions, hence the variance function can be computed by convex optimization techniques as the objective is linear and constraints are convex.

However, a similar generalization for s-rectangular case may not be possible. The structure of L_p norm was crucially used to decompose s-rectangular into a bunch of sa-rectangular sets. We leave this to future work.

H. Experiments

Table 4 contains the relative cost (time) of robust value iteration w.r.t. non-robust MDP, for randomly generated kernel and reward function with the number of states S and the number of action A.

Each experiments are repeated 500 times. The standard deviation of all experiments was less than 10%, and the typical standard deviation is $1-2\%$.

Table 10. Relative running cost (time) for value iteration

$\mathcal{U}$	S=10 A=10	S=30 A=10	S=50 A=10	S=100 A=20	remark
non-robust	1	1	1	1	
$\mathcal{U}^{sa}_{\infty}$ by LP	1374	2282	2848	6930	scipy.optimize.linearprog
$\mathcal{U}^{sa}_{1}$ by LP	1438	6616	6622	16714	scipy.optimize.linearprog
$\mathcal{U}^{s}_{1}$ by LP	72625	629890	4904004	NA	scipy.optimize.linearprog/minimize
$\mathcal{U}^{sa}_{1}$	1.77	1.38	1.54	1.45	closed form
$\mathcal{U}^{sa}_{2}$	1.51	1.43	1.91	1.59	closed form
$\mathcal{U}^{sa}_{\infty}$	1.58	1.48	1.37	1.58	closed form
$\mathcal{U}^{s}_{1}$	1.41	1.58	1.20	1.16	closed form
$\mathcal{U}^{s}_{2}$	2.63	2.82	2.49	2.18	closed form
$\mathcal{U}^{s}_{\infty}$	1.41	3.04	2.25	1.50	closed form
$\mathcal{U}^{sa}_{5}$	5.4	4.91	4.14	4.06	binary search
$\mathcal{U}^{sa}_{10}$	5.56	5.29	4.15	3.26	binary search
$\mathcal{U}^{s}_{5}$	33.30	89.23	40.22	41.22	binary search
$\mathcal{U}^{s}_{10}$	33.59	78.17	41.07	41.10	binary search

Notations

S : number of states, A: number of actions, $\mathcal{U}_p^{sa}$ LP: Sa rectangular L_p robust MPDs by Linear Programming, $\mathcal{U}_p^{s}$ LP: S rectangular L_p robust MPDs by Linear Programming and other numerical methods, $\mathcal{U}_{p=1,2,\infty}^{sa/s}$: Sa/S rectangular $L_1/L_2/L_\infty$ robust MDPs by closed form method (see table 2, theorem 3) $\mathcal{U}_{p=5,10}^{sa/s}$: Sa/S rectangular L_5/L_{10} robust MDPs by binary search (see table 2, theorem 3 of the paper)

Observations

1. Our method for s/sa rectangular $L_1/L_2/L_\infty$ robust MDPs takes almost the same (1-3 times) the time as non-robust MDP for one iteration of value iteration. This confirms our complexity analysis (see table 4 of the paper) 2. Our binary search method for sa rectangular L_5/L_{10} robust MDPs takes around $4-6$ times more time than non-robust counterpart. This is due to extra iterations required to find p-variance function $\kappa_p(v)$ through binary search. 3. Our binary search method for s rectangular L_5/L_{10} robust MDPs takes around $30-100$ times more time than non-robust counterpart. This is due to extra iterations required to find p-variance function $\kappa_p(v)$ through binary search and Bellman operator. 4. One common feature of our method is that time complexity scales moderately as guaranteed through our complexity analysis. 5. Linear programming methods for sa-rectangular L_1/L_∞ robust MDPs take at least 1000 times more than our methods for small state-action space, and it scales up very fast. 6. Numerical methods (Linear programming for minimization over uncertainty and 'scipy.optimize.minimize' for maximization over policy) for s-rectangular L_1 robust MDPs take 4-5 order more time than our methods (and non-robust MDPs) for very small state-action space, and scales up too fast. The reason is obvious, as it has to solve two optimizations, one minimization over uncertainty and the other maximization over policy, whereas, in the sa-rectangular case, only minimization over uncertainty is required. This confirms that the s-rectangular uncertainty set is much more challenging.

Rate of convergence

The rate of convergence for all was approximately the same as $0.9 = \gamma$, as predicted by theory. It is well illustrated by the relative rate of convergence w.r.t. non-robust by the Table 11.

In the above experiments, Bellman updates for sa/s rectangular $L_1/L_2/L_\infty$ were done in closed form, and for L_5/L_{10} were done by binary search as suggested by Table 2 and Theorem 3.

Note: The above experiments' results are for a few runs, hence containing some stochasticity but the general trend is clear. In the final version, we will do an averaging of many runs to minimize the stochastic nature. Results for many different runs can be found at https://github.com/******.

Note that the above experiments were done without using too much parallelization. There is ample scope to fine-tune and improve the performance of robust MDPs. The above experiments confirm the theoretical complexity provided in Table 4 of the paper. The codes and results can be found at https://github.com/******.

Experiments parameters

Number of states S (variable), number of actions A (variable), transition kernel and reward function generated randomly, discount factor 0.9, uncertainty radiuses =0.1 (for all states and action, just for convenience), number of iterations = 100, tolerance for binary search = 0.00001

Hardware

The experiments are done on the following hardware: Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz 64 bits, memory 7862MiB Software: Experiments were done in Python, using numpy, scipy.optimize.linprog for Linear programming for policy evaluation in s/sa rectangular robust MDPs, scipy.optimize.minize, and scipy.optimize.LinearConstraints for policy improvement in s-rectangular L_1 robust MDPs.

I. Extension to Model Free Settings

Extension of Q-learning (in section E) for `sa`-rectangular MDPs to model free setting can easily done similar to (Wang & Zou, 2021), also policy gradient method can be obtained as (Wang & Zou, 2022). The only thing, we need to do, is to be

Table 11. Relative observed rate of convergence

$\mathcal{U}$	S=10 A=10	S=100 A=20	remark
non-robust	1	1	
$\mathcal{U}^{sa}_1$	0.999	0.999	closed form
$\mathcal{U}^{sa}_2$	0.999	0.999	closed form
$\mathcal{U}^{sa}_\infty$	1.000	0.998	closed form
$\mathcal{U}^{s}_1$	0.999	0.999	closed-form
$\mathcal{U}^{s}_2$	0.999	0.999	closed form
$\mathcal{U}^{s}_\infty$	1.000	0.998	closed form
$\mathcal{U}^{sa}_5$	0.999	0.995	binary search
$\mathcal{U}^{sa}_{10}$	1.000	0.999	binary search
$\mathcal{U}^{s}_5$	1.000	0.999	binary search
$\mathcal{U}^{s}_{10}$	1.000	0.995	binary search

able to compute/estimate κ_q online. It can be estimated using an ensemble (samples). Further, κ_2 can be estimated by the estimated mean and the estimated second moment. κ_∞ can be estimated by tracking maximum and minimum values.

For $\mathtt{s}$-rectangular case too, we can obtain model-free algorithms easily, by estimating κ_q online and keeping track of Q-values and value functions. The convergence analysis may be similar to (Wang & Zou, 2021), especially for $\mathtt{sa}$-rectangular case, and for the other, it would be two-time scale, which can be dealt with techniques in (Borkar, 2022). We leave this for future work. It would be interesting to obtain policy gradient methods for this case, which we believe can be obtained from the policy evaluation theorem.

J. Variance Function and its Properties

In this section, we discuss the p-variance function κ_p which is omnipresent in the main text. Here, we outline its properties, origin, and about its computation.

J.1. Closed-Form Expressions for Special Cases

Recall that κ_p is defined as follows

$$\kappa_p(v) = \min_{w}\lVert v - \omega\mathbf{1}\rVert_p = \lVert v - \omega_p(v)\rVert_p,$$

where $\omega_p(v) \in arg\min_w \|v - \omega\mathbf{1}\|_p$. Since $\|v - \omega\mathbf{1}\|_p$ is a convex function in ω and $\omega_p(v)$ is the minimizer of the above expression, hence it must satisfy the stationary condition. That is,

$$\begin{aligned} &\frac{\partial\|v-\omega\|_p}{\partial\omega}\Big|_{\omega=\omega_p(v)} = 0 \\ \implies &\sum_s sign(v(s)-\omega_p(v))|v(s)-\omega_p(v)|^{p-1} = 0, \\ \implies &\sum_s sign(v(s)-\omega_p(v))|v(s)-\omega_p(v)|^{p-1} = 0. \end{aligned} \tag{34}$$

For $p = 2$, stationary condition (34) simplifies to

$$\sum_s sign(v(s)-\omega_2(v))|v(s)-\omega_p(v)| = \sum_s [v(s)-\omega_2(v)] = 0,$$

yielding $\omega_2(v) = \frac{\sum_s v(s)}{S}$. Putting back, we get

$$\begin{aligned} \kappa_2(v) =&\|v-\omega_2\mathbf{1}\|_2 \\ =&\|v - \frac{\sum_s v(s)}{S}\mathbf{1}\|_2, \\ =&\sqrt{\sum_s (v(s) - \frac{\sum_s v(s)}{S})^2}. \end{aligned} \tag{35}$$

For $p = 1$, stationary condition (34) simplifies to

$$\sum_{s\in\mathcal{S}} sign\ \big(\ v(s) - \omega_1(v)\ \big) = 0. \tag{36}$$

Note that there may be more than one value of $\omega_1(v)$ that satisfies the above equation and each solution does an equally good job (as we will see later). So we will pick one (is median of v) according to our convenience as

$$\omega_1(v) = \frac{v(s_{\lfloor (S+1)/2\rfloor}) + v(s_{\lceil (S+1)/2\rceil})}{2} \quad \text{where} \quad v(s_i) \geq v(s_{i+1}) \quad \forall i.$$

Putting it back, we get

$$\begin{aligned} \kappa_1(v) =&\|v-\omega_1\mathbf{1}\|_1 \\ =&\|v - med(v)\mathbf{1}\|_1, \qquad \text{(putting in value of } \omega_0\text{, see table 12)} \\ =&\sum_s |v(s) - med(v)| \\ =&\sum_{i=1}^{\lfloor (S+1)/2\rfloor} (v(s) - med(v)) + \sum_{\lceil (S+1)/2\rceil}^{S} (med(v) - v(s)) \\ =&\sum_{i=1}^{\lfloor (S+1)/2\rfloor} v(s) - \sum_{\lceil (S+1)/2\rceil}^{S} v(s) \end{aligned} \tag{37}$$

where $med(v) := \frac{v(s_{\lfloor (S+1)/2\rfloor}) + v(s_{\lceil (S+1)/2\rceil})}{2}$ where $v(s_i) \geq v(s_{i+1}) \quad \forall i$ is median of v.

For the limiting case of $p = \infty$, it is clear that the expression $\|v - \omega\mathbf{1}\|_\infty$ is minimized for $\omega = \frac{\max_s v(s) + \min_s v(s)}{2}$, yielding $\omega_\infty(v) = \frac{\max_s v(s) + \min_s v(s)}{2}$ and $\kappa_\infty(v) = \frac{\max_s v(s) - \min_s v(s)}{2}$.

The results are summarized in table 2 and 12.

Table 12. p-mean, where $v(s_i) \geq v(s_{i+1}) \quad \forall i$.

x	$\omega_x(v)$	remark
p	$\sum_s sign(v(s)-\omega_p(v))\lvert v(s)-\omega_p(v)\rvert^{\frac{1}{p-1}} = 0$	Solve by binary search
1	$\frac{v(s_{\lfloor (S+1)/2 \rfloor})+v(s_{\lceil (S+1)/2 \rceil})}{2}$	Median
2	$\frac{\sum_s v(s)}{S}$	Mean
∞	$\frac{\max_s v(s)+\min_s v(s)}{2}$	Average of peaks

J.2. Origin: Variance function and Kernel Noise

This subsection is devoted to the origin of κ which is from noise in the kernel. Note that in policy evaluation step (minimization over kernel noise) has the optimization of the form:

$$\min_c \langle c, v\rangle, \qquad \lVert c\rVert_p \leq \epsilon, \qquad \sum_s c(s) = 0.$$

The lemma below connects it to the variance function.

Lemma J.1. *q-variance function κ_q is the solution of the following optimization problem (kernel noise),*

$$\kappa_q(v) = -\frac{1}{\epsilon}\min_c \langle c, v\rangle, \qquad \lVert c\rVert_p \leq \epsilon, \qquad \sum_s c(s) = 0.$$

Proof. Writing Lagrangian L, as

$$L := \sum_s c(s)v(s) + \lambda \sum_s c(s) + \mu(\sum_s \lvert c(s)\rvert^p - \epsilon^p),$$

where $\lambda \in \mathbb{R}$ is the multiplier for the constraint $\sum_s c(s) = 0$ and $\mu \geq 0$ is the multiplier for the inequality constraint $\lVert c\rVert_q \leq \epsilon$. Taking its derivative, we have

$$\frac{\partial L}{\partial c(s)} = v(s) + \lambda + \mu p \lvert c(s)\rvert^{p-1}\frac{c(s)}{\lvert c(s)\rvert} \tag{38}$$

From the KKT (stationarity) condition, the solution c^* has zero derivative, that is

$$v(s) + \lambda + \mu p \lvert c^*(s)\rvert^{p-1}\frac{c^*(s)}{\lvert c^*(s)\rvert} = 0, \qquad \forall s \in \mathcal{S}. \tag{39}$$

Using Lagrangian derivative equation (39), we have

$$\begin{aligned}
& v(s) + \lambda + \mu p \lvert c^*(s)\rvert^{p-1}\frac{c^*(s)}{\lvert c^*(s)\rvert} = 0 \\
\implies & \sum_s c^*(s)[v(s) + \lambda + \mu p \lvert c^*(s)\rvert^{p-1}\frac{c^*(s)}{\lvert c^*(s)\rvert}] = 0, \qquad \text{(multiply with } c^*(s) \text{ and summing)} \\
\implies & \sum_s c^*(s)v(s) + \lambda \sum_s c^*(s) + \mu p \sum_s \lvert c^*(s)\rvert^{p-1}\frac{(c^*(s))^2}{\lvert c^*(s)\rvert} = 0 \qquad (40) \\
\implies & \langle c^*, v\rangle + \mu p \sum_s \lvert c^*(s)\rvert^p = 0 \qquad \text{(using } \sum_s c^*(s) = 0 \text{ and } (c^*(s))^2 = \lvert c^*(s)\rvert^2 \text{)} \\
\implies & \langle c^*, v\rangle = -\mu p \epsilon^p, \qquad \text{(using } \sum_s \lvert c^*(s)\rvert^p = \epsilon^p \text{).}
\end{aligned}$$

It is easy to see that $\mu \geq 0$, as the minimum value of the objective, must not be positive (at $c = 0$, the objective value is zero). Again we use Lagrangian derivative (39) and try to get the objective value ($-\mu p\epsilon^p$) in terms of λ, as

$$\begin{aligned}
&v(s) + \lambda + \mu p|c^*(s)|^{p-1}\frac{c^*(s)}{|c^*(s)|} = 0 \\
\implies &|c^*(s)|^{p-2}c^*(s) = -\frac{v(s)+\lambda}{\mu p}, \qquad \text{(re-arranging terms)} \\
\implies &\sum_s |(|c^*(s)|^{p-2}c^*(s))|^{\frac{p}{p-1}} = \sum_s |-\frac{v(s)+\lambda}{\mu p}|^{\frac{p}{p-1}}, \qquad \text{(doing } \sum_s |\cdot|^{\frac{p}{p-1}}\text{)} \\
\implies &\|c^*\|_p^p = \sum_s |-\frac{v(s)+\lambda}{\mu p}|^{\frac{p}{p-1}} = \sum_s |\frac{v(s)+\lambda}{\mu p}|^q = \frac{\|v+\lambda\|_q^q}{|\mu p|^q} \\
\implies &|\mu p|^q\|c^*\|_p^p = \|v+\lambda\|_q^q, \qquad \text{(re-arranging terms)} \\
\implies &|\mu p|^q\epsilon^p = \|v+\lambda\|_q^q, \qquad \text{(using } \sum_s |c^*(s)|^p = \epsilon^p\text{)} \\
\implies &\epsilon(\mu p\epsilon^{p/q}) = \epsilon\|v+\lambda\|_q \qquad \text{(taking } \frac{1}{q}\text{the power then multiplying with } \epsilon) \\
\implies &\mu p\epsilon^p = \epsilon\|v+\lambda\|_q.
\end{aligned} \tag{41}$$

Again, using Lagrangian derivative (39) to solve for λ, we have

$$\begin{aligned}
&v(s) + \lambda + \mu p|c^*(s)|^{p-1}\frac{c^*(s)}{|c^*(s)|} = 0 \\
\implies &|c^*(s)|^{p-2}c^*(s) = -\frac{v(s)+\lambda}{\mu p}, \qquad \text{(re-arranging terms)} \\
\implies &|c^*(s)| = |\frac{v(s)+\lambda}{\mu p}|^{\frac{1}{p-1}}, \quad \text{(looking at absolute value)} \\
&\text{and} \quad \frac{c^*(s)}{|c^*(s)|} = -\frac{v(s)+\lambda}{|v(s)+\lambda|}, \quad \text{(looking at sign: and note } \mu, p \geq 0) \\
\implies &\sum_s \frac{c^*(s)}{|c^*(s)|}|c^*(s)| = -\sum_s \frac{v(s)+\lambda}{|v(s)+\lambda|}|\frac{v(s)+\lambda}{\mu p}|^{\frac{1}{p-1}}, \qquad \text{(putting back)} \\
\implies &\sum_s c^*(s) = -\sum_s \frac{v(s)+\lambda}{|v(s)+\lambda|}|\frac{v(s)+\lambda}{\mu p}|^{\frac{1}{p-1}}, \\
\implies &\sum_s \frac{v(s)+\lambda}{|v(s)+\lambda|}|v(s)+\lambda|^{\frac{1}{p-1}} = 0, \qquad \text{(using } \sum_i c^*(s) = 0)
\end{aligned} \tag{42}$$

Combining everything, we have

$$\begin{aligned}
&-\frac{1}{\epsilon}\min_c \langle c, v\rangle, \qquad \|c\|_p \leq \epsilon, \qquad \sum_s c(s) = 0 \\
=&\|v-\lambda\|_q, \quad \text{such that} \quad \sum_s sign(v(s)-\lambda)|v(s)-\lambda|^{\frac{1}{p-1}} = 0.
\end{aligned} \tag{43}$$

Now, observe that

$$\begin{aligned}
&\frac{\partial\|v-\lambda\|_q}{\partial\lambda} = 0 \\
\implies &\sum_s sign(v(s)-\lambda)|v(s)-\lambda|^{\frac{1}{p-1}} = 0, \\
\implies &\kappa_q(v) = \|v-\lambda\|_q, \quad \text{such that} \quad \sum_s sign(v(s)-\lambda)|v(s)-\lambda|^{\frac{1}{p-1}} = 0.
\end{aligned} \tag{44}$$

The last equality follows from the convexity of p-norm $\|\cdot\|_q$, where every local minima is global minima.

For the sanity check, we re-derived things for $p = 1$ from scratch. For $p = 1$, we have

$$\begin{aligned} &-\frac{1}{\epsilon}\min_c \langle c, v\rangle, \qquad \|c\|_1 \leq \epsilon, \qquad \sum_s c(s) = 0. \\ =&-\frac{1}{2}(\min_s v(s) - \max_s v(s)) \\ =&\kappa_1(v). \end{aligned} \tag{45}$$

It is easy to see the above result, just by inspection. □

J.3. Computation: Binary search for mean and estimation of variance

If the function $f : [-B/2, B/2] \to \mathbb{R}, B \in \mathbb{R}$ is monotonic (WLOG let it be monotonically decreasing) in a bounded domain, and it has a unique root x^* s.t. $f(x^*) = 0$. Then we can find x that is an ϵ-approximation x^* (i.e. $\|x - x^*\| \leq \epsilon$) in $O(B/\epsilon)$ iterations. Why? Let $x_0 = 0$ and

$$x_{n+1} := \begin{cases} \frac{-B+x_n}{2} & \text{if} \quad f(x_n) > 0 \\ \frac{B+x_n}{2} & \text{if} \quad f(x_n) < 0 \\ x_n & \text{if} \quad f(x_n) = 0 \end{cases}.$$

It is easy to observe that $\|x_n - x^*\| \leq B(1/2)^n$. This proves the above claim. This observation will be referred to many times.
Now, we move to the main claims of the section.

Proposition J.2. *The function*

$$h_p(\lambda) := \sum_s sign\ \big(\ v(s) - \lambda\ \big)\big|\ v(s) - \lambda\ \big|^p$$

is monotonically strictly decreasing and also has a root in the range $[\min_s v(s), \max_s v(s)]$.

Proof.

$$\begin{aligned} h_p(\lambda) &= \sum_s \frac{v(s) - \lambda}{|v(s) - \lambda|}|v(s) - \lambda|^p \\ \frac{dh_p}{d\lambda}(\lambda) &= -p\sum_s |v(s) - \lambda|^{p-1} \leq 0, \qquad \forall p \geq 0. \end{aligned} \tag{46}$$

Now, observe that $h_p(\max_s v(s)) \leq 0$ and $h_p(\min_s v(s)) \geq 0$, hence by h_p must have a root in the range $[\min_s v(s), \max_s v(s)]$ as the function is continuous. □

The above proposition ensures that a root $\omega_p(v)$ can be easily found by binary search between $[\min_s v(s), \max_s v(s)]$. Precisely, ϵ approximation of $\omega_p(v)$ can be found in $O(\log(\frac{\max_s v(s) - \min_s v(s)}{\epsilon}))$ number of iterations of binary search. And one evaluation of the function h_p requires $O(S)$ iterations. And we have finite state-action space and bounded reward hence WLOG we can assume $|\max_s v(s)|, |\min_s v(s)|$ are bounded by a constant. Hence, the complexity to approximate ω_p is $O(S\log(\frac{1}{\epsilon}))$.

Let $\hat{\omega}_p(v)$ be an ϵ-approximation of $\omega_p(v)$, that is

$$\big|\ \omega_p(v) - \hat{\omega}_p(v)\ \big| \leq \epsilon.$$

And let $\hat{\kappa}_p(v)$ be approximation of $\kappa_p(v)$ using approximated mean, that is,

$$\hat{\kappa}_p(v) := \|v - \hat{\omega}_p(v)\mathbf{1}\|_p.$$

Now we will show that ϵ error in calculation of p-mean ω_p, induces $O(\epsilon)$ error in estimation of p-variance κ_p. Precisely,

$$\begin{aligned}
\left| \kappa_p(v) - \hat{\kappa}_p(v) \right| =& \left| \left\| v - \omega_p(v)\mathbf{1} \right\|_p - \left\| v - \hat{\omega}_p(v)\mathbf{1} \right\|_p \right| \\
\leq& \left\| \omega_p(v)\mathbf{1} - \hat{\omega}_p(v)\mathbf{1} \right\|_p, \qquad \text{(reverse triangle inequality)} \\
=& \left\| \mathbf{1} \right\|_p \left| \omega_p(v) - \hat{\omega}_p(v) \right| \\
\leq& \left\| \mathbf{1} \right\|_p \epsilon \\
=& S^{\frac{1}{p}}\epsilon \leq S\epsilon.
\end{aligned} \tag{47}$$

For general p, an ϵ approximation of $\kappa_p(v)$ can be calculated in $O(S\log(\frac{S}{\epsilon})$ iterations. Why? We will estimate mean ω_p to an ϵ/S tolerance (with cost $O(S\log(\frac{S}{\epsilon})$) and then approximate the κ_p with this approximated mean (cost $O(S)$).

K. L_p Water Filling/Pouring lemma

In this section, we are going to discuss the following optimization problem,

$$\max_c -\alpha\|c\|_q + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{i=1}^{A} c_i = 1, \quad c_i \geq 0, \quad \forall i$$

where $\alpha \geq 0$, referred as L_p-water pouring problem. We are going to assume WLOG that b is sorted component-wise, that is $b_1 \geq b_2, \cdots \geq b_A$. The above problem for $p = 2$, is studied in (Anava & Levy, 2016). The approach we are going to solve the problem is as follows: a) Write Lagrangian b) Since the problem is convex, any solution of KKT condition is global maximum. c) Obtain conditions using KKT conditions.

Lemma K.1. *Let $b \in \mathbb{R}^A$ be such that its components are in decreasing order (i,e $b_i \geq b_{i+1}$), $\alpha \geq 0$ be any non-negative constant, and*

$$\zeta_p := \max_c -\alpha\|c\|_q + \langle c, b\rangle \qquad \textit{such that} \qquad \sum_{i=1}^{A} c_i = 1, \quad c_i \geq 0, \quad \forall i, \tag{48}$$

and let c^* be a solution to the above problem. Then

1. *Higher components of b, gets higher weight in c^*. In other words, c^* is also sorted component-wise in descending order, that is*
$$c_1^* \geq c_2^*, \cdots, \geq c_A^*.$$
2. *The optimal value ζ_p satisfies the following equation*
$$\alpha^p = \sum_{b_i \geq \zeta_p} (b_i - \zeta_p)^p.$$
3. *The solution c of* (48)*, is related to ζ_p as*
$$c_i = \frac{(b_i - \zeta_p)^{p-1}\mathbf{1}(b_i \geq \zeta_p)}{\sum_s (b_i - \zeta_p)^{p-1}\mathbf{1}(b_i \geq \zeta_p)}.$$
4. *Observe that the top $\chi_p := \max\{i | b_i \geq \zeta_p\}$ actions are active and rest are passive. The number of active actions can be calculated as*
$$\{k | \alpha^p \geq \sum_{i=1}^{k}(b_i - b_k)^p\} = \{1, 2, \cdots, \chi_p\}.$$
5. *Things can be re-written as*
$$c_i \propto \begin{cases} (b_i - \zeta_p)^{p-1} & \textit{if} \quad i \leq \chi_p \\ 0 & \textit{else} \end{cases} \qquad \textit{and} \qquad \alpha^p = \sum_{i=1}^{\chi_p}(b_i - \zeta_p)^p$$

6. *The function* $\sum_{b_i \geq x}(b_i - x)^p$ *is monotonically decreasing in* x*, hence the root* ζ_p *can be calculated efficiently by binary search between* $[b_1 - \alpha, b_1]$*.*

7. *Solution is sandwiched as follows*
$$b_{\chi_p+1} \leq \zeta_p \leq b_{\chi_p}$$

8. $k \leq \chi_p$ *if and only if there exists the solution of the following,*
$$\sum_{i=1}^{k}(b_i - x)^p = \alpha^p \quad \textit{and} \quad x \leq b_k.$$

9. *If action* k *is active and there is greedy increment hope then action* $k+1$ *is also active. That is*
$$k \leq \chi_p \quad \textit{and} \quad \lambda_k \leq b_{k+1} \implies k+1 \leq \chi_p,$$
where
$$\sum_{i=1}^{k}(b_i - \lambda_k)^p = \alpha^p \quad \textit{and} \quad \lambda_k \leq b_k.$$

10. *If action* k *is active, and there is no greedy hope then action* $k+1$ *is not active. That is,*
$$k \leq \chi_p \quad \textit{and} \quad \lambda_k > b_{k+1} \implies k+1 > \chi_p,$$
where
$$\sum_{i=1}^{k}(b_i - \lambda_k)^p = \alpha^p \quad \textit{and} \quad \lambda_k \leq b_k.$$
And this implies $k = \chi_p$*.*

Proof. 1. Let
$$f(c) := -\alpha\|c\|_q + \langle b, c\rangle.$$
Let c be any vector, and c' be rearrangement c in descending order. Precisely,
$$c'_k := c_{i_k}, \quad \text{where} \quad c_{i_1} \geq c_{i_2}, \cdots, \geq c_{i_A}.$$
Then it is easy to see that $f(c') \geq f(c)$. And the claim follows.

2. Writing Lagrangian of the optimization problem, and its derivative,
$$\begin{aligned} L &= -\alpha\|c\|_q + \langle c, b\rangle + \lambda(\sum_i c_i - 1) + \theta_i c_i \\ \frac{\partial L}{\partial c_i} &= -\alpha\|c\|_q^{1-q}|c_i|^{q-2}c_i + b_i + \lambda + \theta_i, \end{aligned} \tag{49}$$
$\lambda \in \mathbb{R}$ is multiplier for equality constraint $\sum_i c_i = 1$ and $\theta_1, \cdots, \theta_A \geq 0$ are multipliers for inequality constraints $c_i \geq 0, \quad \forall i \in [A]$. Using KKT (stationarity) condition, we have
$$-\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* + b_i + \lambda + \theta_i = 0 \tag{50}$$
Let $\mathcal{B} := \{i | c_i^* > 0\}$, then
$$\begin{aligned} &\sum_{i \in \mathcal{B}} c_i^*[-\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* + b_i + \lambda] = 0 \\ \implies &-\alpha\|c^*\|_q^{1-q}\|c^*\|_q^q + \langle c^*, b\rangle + \lambda = 0, \qquad (\text{using } \sum_i c_i^* = 1 \text{ and } (c_i^*)^2 = |c_i^*|^2) \\ \implies &-\alpha\|c^*\|_q + \langle c^*, b\rangle + \lambda = 0 \\ \implies &-\alpha\|c^*\|_q + \langle c^*, b\rangle = -\lambda, \qquad (\text{re-arranging}) \end{aligned} \tag{51}$$

Now again using (50), we have

$$\begin{aligned} &-\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* + b_i + \lambda + \theta_i = 0 \\ \implies &\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* = b_i + \lambda + \theta_i, \qquad \forall i, \qquad \text{(re-arranging)} \end{aligned} \tag{52}$$

Now, if $i \in \mathcal{B}$ then $\theta_i = 0$ from complimentary slackness, so we have

$$\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* = b_i + \lambda > 0, \qquad \forall i \in \mathcal{B}$$

by definition of $\mathcal{B}$. Now, if for some i, $b_i + \lambda > 0$ then $b_i + \lambda + \theta_i > 0$ as $\theta_i \geq 0$, that implies

$$\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* = b_i + \lambda + \theta_i > 0$$

$$\implies c_i^* > 0 \implies i \in \mathcal{B}.$$

So, we have,

$$i \in \mathcal{B} \iff b_i + \lambda > 0.$$

To summarize, we have

$$\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* = (b_i + \lambda)\mathbf{1}(b_i \geq -\lambda), \quad \forall i, \tag{53}$$

$$\begin{aligned} &\implies \sum_i \alpha^{\frac{q}{q-1}}\|c^*\|_q^{-q}(c_i^*)^q = \sum_i (b_i + \lambda)^{\frac{q}{q-1}}\mathbf{1}(b_i \geq -\lambda), \qquad \text{(taking } q/(q-1)\text{th power and summming)} \\ &\implies \alpha^p = \sum_{i=1}^{A}(b_i + \lambda)^p\mathbf{1}(b_i \geq -\lambda). \end{aligned} \tag{54}$$

So, we have,

$$\begin{aligned} &\zeta_p = -\lambda \quad \text{such that} \quad \alpha^p = \sum_{b_i \geq \lambda}(b_i + \lambda)^p. \\ &\implies \alpha^p = \sum_{b_i \geq \zeta_p}(b_i - \zeta_p)^p \end{aligned} \tag{55}$$

3. Furthermore, using (53), we have

$$\begin{aligned} &\alpha\|c^*\|_q^{1-q}|c_i^*|^{q-2}c_i^* = (b_i + \lambda)\mathbf{1}(b_i \geq -\lambda) = (b_i - \zeta_p)\mathbf{1}(b_i \geq \zeta_p) \quad \forall i, \\ \implies &c_i^* \propto (b_i - \zeta_p)^{\frac{1}{q-1}}\mathbf{1}(b_i \geq \zeta_p) = \frac{(b_i - \zeta_p)^{p-1}\mathbf{1}(b_i \geq \zeta_p)}{\sum_i (b_i - \zeta_p)^{p-1}\mathbf{1}(b_i \geq \zeta_p)}, \qquad (\text{using } \sum_i c_i^* = 1). \end{aligned} \tag{56}$$

4. Now, we move on to calculate the number of active actions χ_p. Observe that the function

$$f(\lambda) := \sum_{i=1}^{A}(b_i - \lambda)^p\mathbf{1}(b_i \geq \lambda) - \alpha^p \tag{57}$$

is monotonically decreasing in λ and ζ_p is a root of f. This implies

$$\begin{aligned} &f(x) \leq 0 \iff x \geq \zeta_p \\ \implies &f(b_i) \leq 0 \iff b_i \geq \zeta_p \\ \implies &\{i|b_i \geq \zeta_p\} = \{i|f(b_i) \leq 0\} \\ \implies &\chi_p = \max\{i|b_i \geq \zeta_p\} = \max\{i|f(b_i) \leq 0\}. \end{aligned} \tag{58}$$

Hence, things follow by putting back in the definition of f.

5. We have,
$$\alpha^p = \sum_{i=1}^{A}(b_i - \zeta_p)^p \mathbf{1}(b_i \geq \zeta_p), \quad \text{and} \quad \chi_p = \max\{i | b_i \geq \zeta_p\}.$$
Combining both we have
$$\alpha^p = \sum_{i=1}^{\chi_p}(b_i - \zeta_p)^p.$$
And the other part follows directly.

6. Continuity and montonocity of the function $\sum_{b_i \geq x}(b_i - x)^p$ is trivial. Now observe that $\sum_{b_i \geq b_1}(b_i - b_1)^p = 0$ and $\sum_{b_i \geq b_1 - \alpha}(b_i - (b_1 - \alpha))^p \geq \alpha^p$, so it implies that it is equal to α^p in the range $[b_1 - \alpha, b_1]$.

7. Recall that the ζ_p is the solution to the following equation
$$\alpha^p = \sum_{b_i \geq x}(b_i - x)^p.$$
And from the definition of χ_p, we have
$$\alpha^p < \sum_{i=1}^{\chi_p+1}(b_i - b_{\chi_p+1})^p = \sum_{b_i \geq b_{\chi_p+1}}(b_i - b_{\chi_p+1})^p, \quad \text{and}$$
$$\alpha^p \geq \sum_{i=1}^{\chi_p}(b_i - b_{\chi_p})^p = \sum_{b_i \geq b_{\chi_p}}(b_i - b_{\chi_p})^p.$$
So from continuity, we infer the root ζ_p must lie between $[b_{\chi_p+1}, b_\chi]$.

8. We prove the first direction, and assume we have
$$\begin{aligned} & k \leq \chi_p \\ \implies & \sum_{i=1}^{k}(b_i - b_k)^p \leq \alpha^p \qquad \text{(from definition of } \chi_p\text{)}. \end{aligned} \tag{59}$$
Observe the function $f(x) := \sum_{i=1}^{k}(b_i - x)^p$ is monotically decreasing in the range $(-\infty, b_k]$. Further, $f(b_k) \leq \alpha^p$ and $\lim_{x \to -\infty} f(x) = \infty$, so from the continuity argument there must exist a value $y \in (-\infty, b_k]$ such that $f(y) = \alpha^p$. This implies that
$$\sum_{i=1}^{k}(b_i - y)^p \leq \alpha^p, \quad \text{and} \quad y \leq b_k.$$
Hence, explicitly showed the existence of the solution. Now, we move on to the second direction and assume there exists x such that
$$\sum_{i=1}^{k}(b_i - x)^p = \alpha^p, \quad \text{and} \quad x \leq b_k.$$
$$\implies \sum_{i=1}^{k}(b_i - b_k)^p \leq \alpha^p, \qquad (\text{as } x \leq b_k \leq b_{k-1} \cdots \leq b_1)$$
$$\implies k \leq \chi_p.$$

9. We have $k \leq \chi_p$ and λ_k such that

$$\begin{aligned} \alpha^p &= \sum_{i=1}^{k}(b_i - \lambda_k)^p, \quad \text{and} \quad \lambda_k \leq b_k, \qquad \text{(from above item)} \\ &\geq \sum_{i=1}^{k}(b_i - b_{k+1})^p, \qquad (\text{as } \lambda_k \leq b_{k+1} \leq b_k) \\ &\geq \sum_{i=1}^{k+1}(b_i - b_{k+1})^p, \qquad \text{(addition of 0)}. \end{aligned} \tag{60}$$

   From the definition of χ_p, we get $k + 1 \leq \chi_p$.

10. We are given

$$\begin{aligned} &\sum_{i=1}^{k}(b_i - \lambda_k)^p = \alpha^p \\ \implies &\sum_{i=1}^{k}(b_i - b_{k+1})^p > \alpha^p, \qquad (\text{as } \lambda_k > b_{k+1}) \\ \implies &\sum_{i=1}^{k+1}(b_i - b_{k+1})^p > \alpha^p, \qquad \text{(addition of zero)} \\ \implies &k + 1 > \chi_p. \end{aligned}$$

□

K.1. Special case: $p = 1$

For $p = 1$, by definition, we have

$$\zeta_1 = \max_c -\alpha\|c\|_\infty + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{a\in\mathcal{A}} c_a = 1, \quad c \succeq 0. \tag{61}$$

And χ_1 is the optimal number of actions, that is

$$\alpha = \sum_{i=1}^{\chi_1}(b_i - \zeta_1)$$

$$\implies \zeta_1 = \frac{\sum_{i=1}^{\chi_1} b_i - \alpha}{\chi_1}.$$

Let λ_k be the such that

$$\alpha = \sum_{i=1}^{k}(b_i - \lambda_k)$$

$$\implies \lambda_k = \frac{\sum_{i=1}^{k} b_i - \alpha}{k}.$$

Proposition K.2.

$$\zeta_1 = \max_k \lambda_k$$

Proof. From lemma K.1, we have

$$\lambda_1 \leq \lambda_2 \cdots \leq \lambda_{\chi_1}.$$

Now, we have

$$\begin{aligned}\lambda_k - \lambda_{k+m} &= \frac{\sum_{i=1}^{k} b_i - \alpha}{k} - \frac{\sum_{i=1}^{k+m} b_i - \alpha}{k+m}\\ &= \frac{\sum_{i=1}^{k} b_i - \alpha}{k} - \frac{\sum_{i=1}^{k} b_i - \alpha}{k+m} - \frac{\sum_{i=1}^{m} b_{k+i}}{k+m}\\ &= \frac{m(\sum_{i=1}^{k} b_i - \alpha}{k(k+m))} - \frac{\sum_{i=1}^{m} b_{k+i}}{k+m}\\ &= \frac{m}{k+m}(\frac{\sum_{i=1}^{k} b_i - \alpha}{k} - \frac{\sum_{i=1}^{m} b_{k+i}}{m})\\ &= \frac{m}{k+m}(\lambda_k - \frac{\sum_{i=1}^{m} b_{k+i}}{m})\end{aligned} \tag{62}$$

From lemma K.1, we also know the stopping criteria for χ_1, that is

$$\lambda_{\chi_1} > b_{\chi_1+1}$$

$$\implies \lambda_{\chi_1} > b_{\chi_1+i}, \qquad i \geq 1, \qquad \text{(as } b_i \text{ are in descending order)}$$

$$\implies \lambda_{\chi_1} > \frac{\sum_{i=1}^{m} b_{\chi_1+i}}{m}, \qquad \forall m \geq 1.$$

Combining it with the (62), for all $m \geq 0$, we get

$$\begin{aligned}\lambda_{\chi_1} - \lambda_{\chi_1+m} &= \frac{m}{\chi_1+m}(\lambda_{\chi_1} - \frac{\sum_{i=1}^{m} b_{\chi_1+i}}{m})\\ &\geq 0\\ \implies \lambda_{\chi_1} &\geq \lambda_{\chi_1+m}\end{aligned} \tag{63}$$

Hence, we get the desired result,

$$\zeta_1 = \lambda_{\chi_1} = \max_k \lambda_k.$$

$\square$

K.2. Special case: $p = \infty$

For $p = \infty$, by definition, we have

$$\begin{aligned}\zeta_\infty(b) &= \max_c -\alpha\|c\|_1 + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{a\in\mathcal{A}} c_a = 1, \quad c \succeq 0.\\ &= \max_c -\alpha + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{a\in\mathcal{A}} c_a = 1, \quad c \succeq 0.\\ &= -\alpha + \max_i b_i\end{aligned} \tag{64}$$

K.3. Special case: $p = 2$

The problem is discussed in great detail in (Anava & Levy, 2016), here we outline the proof. For $p = 2$, we have

$$\zeta_2 = \max_c -\alpha\|c\|_2 + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{a\in\mathcal{A}} c_a = 1, \quad c \succeq 0. \tag{65}$$

Let λ_k be the solution of the following equation

$$\begin{aligned}
\alpha^2 &= \sum_{i=1}^{k}(b_i-\lambda)^2, \qquad \lambda \le b_k \\
&= k\lambda^2 - 2\sum_{i=1}^{k}\lambda b_i. + \sum_{i=1}^{k}(b_i)^2, \qquad \lambda \le b_k \\
\implies \lambda_k &= \frac{\sum_{i=1}^{k} b_i \pm \sqrt{(\sum_{i=1}^{k} b_i)^2 - k(\sum_{i=1}^{k}(b_i)^2-\alpha^2)}}{k}, \quad \text{and} \qquad \lambda_k \le b_k \\
&= \frac{\sum_{i=1}^{k} b_i - \sqrt{(\sum_{i=1}^{k} b_i)^2 - k(\sum_{i=1}^{k}(b_i)^2-\alpha^2)}}{k} \\
&= \frac{\sum_{i=1}^{k} b_i}{k} - \sqrt{\alpha^2 - \sum_{i=1}^{k}(b_i - \frac{\sum_{i=1}^{k} b_i}{k})^2}
\end{aligned} \tag{66}$$

From lemma K.1, we know

$$\lambda_1 \le \lambda_2 \cdots \le \lambda_{\chi_2} = \zeta_2$$

where χ_2 calculated in two ways: a)

$$\chi_2 = \max_m\{m | \sum_{i=1}^{m}(b_i - b_m)^2 \le \alpha^2\}$$

b)

$$\chi_2 = \min_m\{m | \lambda_m \le b_{m+1}\}$$

We proceed greedily until the stopping condition is met in lemma K.1. Concretely, it is illustrated in algorithm 7.

K.4. L_1 Water Pouring lemma

In this section, we re-derive the above water pouring lemma for $p = 1$ from scratch, just for sanity check. As in the above proof, there is a possibility of some breakdown, as we had to take limits $q \to \infty$. We will see that all the above results for $p = 1$ too.

Let $b \in \mathbb{R}^A$ be such that its components are in decreasing order, i,e $b_i \ge b_{i+1}$ and

$$\zeta_1 := \max_c -\alpha\|c\|_\infty + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{i=1}^{A} c_i = 1, \quad c_i \ge 0, \quad \forall i. \tag{67}$$

Lets fix any vector $c \in \mathbb{R}^A$, and let $k_1 := \lfloor \frac{1}{\max_i c_i} \rfloor$ and let

$$c_i^1 = \begin{cases} \max_i c_i & \text{if} \quad i \le k_1 \\ 1 - k_1 \max_i c_i & \text{if} \quad i = k_1 + 1 \\ 0 & \text{else} \end{cases}$$

Then we have,

$$\begin{aligned}
-\alpha\|c\|_\infty + \langle c, b\rangle &= -\alpha \max_i c_i + \sum_{i=1}^{A} c_i b_i \\
&\le -\alpha \max_i c_i + \sum_{i=1}^{A} c_i^1 b_i, \qquad \text{(recall } b_i \text{ is in decreasing order)} \\
&= -\alpha\|c^1\|_\infty + \langle c^1, b\rangle
\end{aligned} \tag{68}$$

Now, let's define $c^2 \in \mathbb{R}^A$. Let

$$k_2 = \begin{cases} k_1 + 1 & \text{if } \frac{\sum_{i=1}^{k_1} b_i - \alpha}{k_1} \leq b_{k+1} \\ k_1 & \text{else} \end{cases}$$

and let $c^2_i = \frac{\mathbf{1}(i \leq k_2)}{k_2}$. Then we have,

$$\begin{aligned} -\alpha\|c^1\|_\infty + \langle c^1, b\rangle =& -\alpha \max_i c_i + \sum_{i=1}^{A} c^1_i b_i \\ =& -\alpha \max_i c_i + \sum_{i=1}^{k_1} \max_i c_i b_i + (1 - k_1 \max_i c_i) b_{k_1+1}, \qquad \text{(by definition of } c^1\text{)} \\ =& (\frac{-\alpha + \sum_{i=1}^{k_1} b_i}{k_1}) k_1 \max_i c_i + b_{k_1+1}(1 - k_1 \max_i c_i), \qquad \text{(re-arranging)} \\ \leq& \frac{-\alpha + \sum_{i=1}^{k_2} b_i}{k_2} \\ =& -\alpha\|c^2\|_\infty + \langle c^2, b\rangle \end{aligned} \tag{69}$$

The last inequality comes from the definitions of k_2 and c^2. So we conclude that an optimal solution is uniform over some actions, that is

$$\begin{aligned} \zeta_1 =& \max_{c\in\mathcal{C}} -\alpha\|c\|_\infty + \langle c, b\rangle \\ =& \max_k \Big(\frac{-\alpha + \sum_{i=1}^{k} b_i}{k} \Big) \end{aligned} \tag{70}$$

where $\mathcal{C} := \{c^k \in \mathbb{R}^A | c^k_i = \frac{\mathbf{1}(i \leq k)}{k}\}$ is set of uniform actions. Rest all the properties follow the same as L_p water pouring lemma.

L. Robust Value Iteration (Main)

In this section, we will discuss the main results from the paper except for time complexity results. It contains the proofs of the results presented in the main body and also some other corollaries/special cases.

L.1. (sa)-rectangular robust policy evaluation and improvement

Theorem L.1. *(sa)-rectangular L_p robust Bellman operator is equivalent to reward regularized (non-robust) Bellman operator, that is*

$$\begin{aligned} (\mathcal{T}^\pi_{\mathcal{U}^{sa}_p} v)(s) =& \sum_a \pi(a|s)[-\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s')], \qquad \textit{and} \\ (\mathcal{T}^*_{\mathcal{U}^{sa}_p} v)(s) =& \max_{a\in\mathcal{A}}[-\alpha_{s,a} - \gamma\beta_{s,a}\kappa_q(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s')], \end{aligned}$$

where κ_p is defined in (2).

Proof. From definition robust Bellman operator and $\mathcal{U}_p^{\mathtt{sa}} = (R_0 + \mathcal{R}) \times (P_0 + \mathcal{P})$, we have,

$$\begin{aligned}
(\mathcal{T}^{\pi}_{\mathcal{U}_p^{\mathtt{sa}}} v)(s) &= \min_{R,P \in \mathcal{U}_p^{\mathtt{sa}}} \sum_a \pi(a|s) \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \Big] \\
&= \sum_a \pi(a|s) \Big[R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big] + \\
&\qquad \min_{p \in \mathcal{P}, r \in \mathcal{R}} \sum_a \pi(a|s) \Big[r(s,a) + \gamma \sum_{s'} p(s'|s,a) v(s') \Big], \\
&\quad \text{(from (sa)-rectangularity, we get)} \\
&= \sum_a \pi(a|s) \Big[R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big] + \\
&\qquad \sum_a \pi(a|s) \underbrace{\min_{p_{s,a} \in \mathcal{P}_{sa}, r_{s,a} \in \mathcal{R}_{s,a}} \Big[r_{s,a} + \gamma \sum_{s'} p_{s,a}(s') v(s') \Big]}_{:=\Omega_{sa}(v)}
\end{aligned} \tag{71}$$

Now we focus on regularizer function Ω, as follows

$$\begin{aligned}
\Omega_{sa}(v) &= \min_{p_{s,a} \in \mathcal{P}_{s,a}, r_{s,a} \in \mathcal{R}_{s,a}} \Big[r_{s,a} + \gamma \sum_{s'} p_{s,a}(s') v(s') \Big] \\
&= \min_{r_{s,a} \in \mathcal{R}_{s,a}} r_{s,a} + \gamma \min_{p_{s,a} \in \mathcal{P}_{sa}} \sum_{s'} p_{s,a}(s') v(s') \\
&= -\alpha_{s,a} + \gamma \min_{\|p_{sa}\|_p \le \beta_{s,a}, \sum_{s'} p_{sa}(s') = 0} \langle p_{s,a}, v \rangle, \\
&= -\alpha_{s,a} - \gamma \beta_{s,a} \kappa_q(v), \qquad \text{(from lemma J.1).}
\end{aligned} \tag{72}$$

Putting back, we have

$$(\mathcal{T}^{\pi}_{\mathcal{U}_p^{\mathtt{sa}}} v)(s) = \sum_a \pi(a|s) \Big[-\alpha_{s,a} - \gamma \beta_{s,a} \kappa_q(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big]$$

Again, reusing the above results in optimal robust operator, we have

$$\begin{aligned}
(\mathcal{T}^{*}_{\mathcal{U}_p^{\mathtt{sa}}} v)(s) &= \max_{\pi_s \in \Delta_{\mathcal{A}}} \min_{R,P \in \mathcal{U}_p^{\mathtt{sa}}} \sum_a \pi_s(a) \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \Big] \\
&= \max_{\pi_s \in \Delta_{\mathcal{A}}} \sum_a \pi_s(a) \Big[-\alpha_{s,a} - \gamma \beta_{s,a} \kappa_p(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big] \\
&= \max_{a \in \mathcal{A}} \Big[-\alpha_{s,a} - \gamma \beta_{s,a} \kappa_q(v) + R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big]
\end{aligned} \tag{73}$$

The claim is proved. $\square$

L.2. S-rectangular robust policy evaluation

Policy improvement in the s-rectangular case is more involved, hence we begin with policy evaluation.

Theorem L.2. S-*rectangular L_p robust Bellman operator is equivalent to reward regularized (non-robust) Bellman operator, that is*

$$(\mathcal{T}^{\pi}_{\mathcal{U}_p^{s}} v)(s) = -\Big(\alpha_s + \gamma \beta_s \kappa_q(v) \Big) \|\pi(\cdot|s)\|_q + \sum_a \pi(a|s) \Big(R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v(s') \Big)$$

where κ_p is defined in (2) *and $\|\pi(\cdot|s)\|_q$ is q-norm of the vector $\pi(\cdot|s) \in \Delta_{\mathcal{A}}$.*

Proof. From definition of robust Bellman operator and $\mathcal{U}^{\mathtt{s}}_p = (R_0 + \mathcal{R}) \times (P_0 + \mathcal{P})$, we have

$$\begin{aligned}
(\mathcal{T}^{\pi}_{\mathcal{U}^s_p} v)(s) &= \min_{R,P\in\mathcal{U}^s_p} \sum_a \pi(a|s) \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s') \Big] \\
&= \sum_a \pi(a|s) \underbrace{\Big[R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s') \Big]}_{\text{nominal values}} \\
&\qquad + \min_{p\in\mathcal{P}, r\in\mathcal{R}} \sum_a \pi(a|s) \Big[r(s,a) + \gamma \sum_{s'} p(s'|s,a)v(s') \Big] \\
&\quad \text{(from s-rectangularity we have)} \\
&= \sum_a \pi(a|s) \Big[R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s') \Big] \\
&\qquad + \underbrace{\min_{p_s\in\mathcal{P}_s, r_s\in\mathcal{R}_s} \sum_a \pi(a|s) \Big[r_s(a) + \gamma \sum_{s'} p_s(s'|a)v(s') \Big]}_{:=\Omega_s(\pi_s, v)}
\end{aligned} \tag{74}$$

where we denote $\pi_s(a) = \pi(a|s)$ as a shorthand. Now we calculate the regularizer function as follows

$$\begin{aligned}
\Omega_s(\pi_s, v) &:= \min_{r_s\in\mathcal{R}_s, p_s\in\mathcal{P}_s} \langle r_s + \gamma v^T p_s, \pi_s\rangle = \min_{r_s\in\mathcal{R}_s} \langle r_s, \pi_s\rangle + \gamma \min_{p_s\in\mathcal{P}_s} v^T p_s \pi_s \\
&= -\alpha_s \|\pi_s\|_q + \gamma \min_{p_s\in\mathcal{P}_s} v^T p_s \pi_s, \qquad \text{(using Holders inequality, where } \frac{1}{p} + \frac{1}{q} = 1 \text{)} \\
&= -\alpha_s \|\pi_s\|_q + \gamma \min_{p_s\in\mathcal{P}_s} \sum_a \pi_s(a) \langle p_{s,a}, v\rangle \\
&= -\alpha_s \|\pi_s\|_q + \gamma \min_{\sum_a (\beta_{s,a})^p \le (\beta_s)^p} \min_{\|p_{sa}\|_p \le \beta_{s,a}, \sum_{s'} p_{sa}(s')=0} \sum_a \pi_s(a) \langle p_{s,a}, v\rangle \\
&= -\alpha_s \|\pi_s\|_q + \gamma \min_{\sum_a (\beta_{s,a})^p \le (\beta_s)^p} \sum_a \pi_s(a) \min_{\|p_{sa}\|_p \le \beta_{s,a}, \sum_{s'} p_{sa}(s')=0} \langle p_{s,a}, v\rangle \\
&= -\alpha_s \|\pi_s\|_q + \gamma \min_{\sum_a (\beta_{sa})^p \le (\beta_s)^p} \sum_a \pi_s(a)(-\beta_{sa}\kappa_p(v)) \qquad \text{(from lemma J.1)} \\
&= -\alpha_s \|\pi_s\|_q - \gamma \kappa_q(v) \max_{\sum_a (\beta_{sa})^p \le (\beta_s)^p} \sum_a \pi_s(a) \beta_{sa} \\
&= -\alpha_s \|\pi_s\|_q - \gamma \kappa_p(v) \|\pi_s\|_q \beta_s \qquad \text{(using Holders)} \\
&= -(\alpha_s + \gamma\beta_s\kappa_q(v))\|\pi_s\|_q.
\end{aligned} \tag{75}$$

Now putting the above values in robust operator, we have

$$(\mathcal{T}^{\pi}_{\mathcal{U}^s_p} v)(s) = -\Big(\alpha_s + \gamma\beta_s\kappa_q(v) \Big) \|\pi(\cdot|s)\|_q + \sum_a \pi(a|s) \Big(R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s') \Big).$$

□

L.3. s-rectangular robust policy improvement

Reusing robust policy evaluation results in section L.2, we have

$$\begin{aligned}
(\mathcal{T}^{*}_{\mathcal{U}^s_p} v)(s) &= \max_{\pi_s\in\Delta_{\mathcal{A}}} \min_{R,P\in\mathcal{U}^{\mathtt{sa}}_p} \sum_a \pi_s(a) \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s') \Big] \\
&= \max_{\pi_s\in\Delta_{\mathcal{A}}} \Big[-(\alpha_s + \gamma\beta_s\kappa_q(v))\|\pi_s\|_q + \sum_a \pi_s(a)(R(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s')) \Big].
\end{aligned} \tag{76}$$

Observe that, we have the following form

$$(\mathcal{T}^*_{\mathcal{U}^s_p} v)(s) = \max_c -\alpha \|c\|_q + \langle c, b\rangle \qquad \text{such that} \qquad \sum_{i=1}^{A} c_i = 1, \quad c \succeq 0, \tag{77}$$

where $\alpha = \alpha_s + \gamma\beta_s\kappa_q(v)$ and $b_i = R(s,a_i) + \gamma\sum_{s'} P(s'|s,a_i)v(s')$. Now all the results below follow from the water pouring lemma (lemma K.1).

Algorithm 6 Algorithm to compute s-rectangular L_p robust optimal Bellman Operator

```
1: Input: $\sigma = \alpha_s + \gamma\beta_s\kappa_q(v)$,    $Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$.
2: Output $(\mathcal{T}^*_{\mathcal{U}^s_p} v)(s), \chi_p(v,s)$
3: Sort $Q(s,\cdot)$ and label actions such that $Q(s,a_1) \geq Q(s,a_2), \cdots$.
4: Set initial value guess $\lambda_1 = Q(s,a_1) - \sigma$ and counter $k = 1$.
5: while $k \leq A-1$ and $\lambda_k \leq Q(s,a_k)$ do
6:     Increment counter: $k = k+1$
7:     Take $\lambda_k$ to be a solution of the following
           $$\sum_{i=1}^{k} \big(\, Q(s,a_i) - x \,\big)^p = \sigma^p, \quad \text{and} \quad x \leq Q(s,a_k). \tag{78}$$
8: end while
9: Return: $\lambda_k, k$
```

Theorem L.3. *(Policy improvement) The optimal robust Bellman operator can be evaluated in the following ways.*

1. *$(\mathcal{T}^*_{\mathcal{U}^s_p} v)(s)$ is the solution of the following equation that can be found using binary search between $\big[\, \max_a Q(s,a) - \sigma, \max_a Q(s,a) \,\big]$,*

$$\sum_a \big(\, Q(s,a) - x \,\big)^p \, \mathbf{1}\big(\, Q(s,a) \geq x \,\big) = \sigma^p. \tag{79}$$

2. *$(\mathcal{T}^*_{\mathcal{U}^s_p} v)(s)$ and $\chi_p(v,s)$ can also be computed through algorithm 6.*

where $\sigma = \alpha_s + \gamma\beta_s\kappa_q(v)$, and $Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$.

Proof. The first part follows from lemma K.1, point 2. The second part follows from lemma K.1, point 9 (greedy inclusion) and point 10 (stopping condition). □

Theorem L.4. *(Go To Policy) The greedy policy π w.r.t. value function v, defined as $\mathcal{T}^*_{\mathcal{U}^s_p} v = \mathcal{T}^\pi_{\mathcal{U}^s_p} v$ is a threshold policy. It takes only those actions that have a positive advantage, with probability proportional to $(p-1)$th power of its advantage. That is*

$$\pi(a|s) \propto (A(s,a))^{p-1}\mathbf{1}(A(s,a) \geq 0),$$

where $A(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s') - (\mathcal{T}^*_{\mathcal{U}^s_p} v)(s)$.

Proof. Follows from lemma K.1, point 3. □

Property L.5. *$\chi_p(v,s)$ is the number of actions that have a positive advantage, that is*

$$\chi_p(v,s) = \Big|\Big\{\, a \mid (\mathcal{T}^*_{\mathcal{U}^s_p} v)(s) \leq R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s') \,\Big\}\Big|.$$

Proof. Follows from lemma K.1, point 4. □

Property L.6. *(Value vs Q-value) $(\mathcal{T}^*_{\mathcal{U}^s_p}v)(s)$ is bounded by the Q-value of χth and $(\chi+1)$th actions. That is*

$$Q(s,a_{\chi+1}) < (\mathcal{T}^*_{\mathcal{U}^s_p}v)(s) \le Q(s,a_\chi), \qquad \textit{where} \quad \chi = \chi_p(v,s),$$

$Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$, and $Q(s,a_1) \ge Q(s,a_2), \cdots Q(s,a_A)$.

Proof. Follows from lemma K.1, point 7. □

Corollary L.7. *For $p=1$, the optimal policy π_1 w.r.t. value function v and uncertainty set $\mathcal{U}^s_1$, can be computed directly using $\chi_1(s)$ without calculating advantage function. That is*

$$\pi_1(a^s_i|s) = \frac{\mathbf{1}(i \le \chi_1(s))}{\chi_1(s)}.$$

Proof. Follows from Theorem L.4 by putting $p=1$. Note that it can be directly obtained using L_1 water pouring lemma (see section K.4) □

Corollary L.8. *(For $p=\infty$) The optimal policy π w.r.t. value function v and uncertainty set $\mathcal{U}^s_\infty$ (precisely $\mathcal{T}^*_{\mathcal{U}^s_\infty}v = \mathcal{T}^\pi_{\mathcal{U}^s_\infty}v$), is to play the best response, that is*

$$\pi(a|s) = \frac{\mathbf{1}(a \in \textit{arg}\max_a Q(s,a))}{\mid \textit{arg}\max_a Q(s,a) \mid}.$$

In case of tie-in the best response, it is optimal to play any of the best responses with any probability.

Proof. Follows from Theorem L.4 by taking limit $p\to\infty$. □

Corollary L.9. *For $p=\infty$, $\mathcal{T}^*_{\mathcal{U}^s_p}v$, the robust optimal Bellman operator evaluation can be obtained in closed form. That is*

$$(\mathcal{T}^*_{\mathcal{U}^s_\infty}v)(s) = \max_a Q(s,a) - \sigma,$$

where $\sigma = \alpha_s + \gamma\beta_s\kappa_1(v), Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$.

Proof. Let π be such that

$$\mathcal{T}^*_{\mathcal{U}^s_\infty}v = \mathcal{T}^\pi_{\mathcal{U}^s_\infty}v.$$

This implies

$$\begin{aligned}(\mathcal{T}^*_{\mathcal{U}^\mathtt{s}_p}v)(s) &= \min_{R,P\in\mathcal{U}^\mathtt{sa}_p}\sum_a \pi(a|s)\Big[\,R(s,a) + \gamma\sum_{s'}P(s'|s,a)v(s')\,\Big] \\ &= -(\alpha_s + \gamma\beta_s\kappa_p(v))\|\pi(\cdot|s)\|_q + \sum_a \pi(a|s)(R(s,a) + \gamma\sum_{s'}P(s'|s,a)v(s')).\end{aligned} \tag{80}$$

From corollary L.8, we know that π is a deterministic best response policy. Putting this we get the desired result.
There is another way of proving this, using Theorem 3.3 by taking limit $p\to\infty$ carefully as

$$\lim_{p\to\infty}\sum_a \Big(Q(s,a) - \mathcal{T}^*_{\mathcal{U}^\mathtt{s}_p}v)(s)\Big)^p \mathbf{1}\Big(Q(s,a) \ge \mathcal{T}^*_{\mathcal{U}^\mathtt{s}_p}v)(s)\Big))^{\frac{1}{p}} = \sigma, \tag{81}$$

where $\sigma = \alpha_s + \gamma\beta_s\kappa_1(v)$. □

Corollary L.10. *For $p=1$, the robust optimal Bellman operator $\mathcal{T}^*_{\mathcal{U}^s_p}$, can be computed in closed form. That is*

$$(\mathcal{T}^*_{\mathcal{U}^s_p}v)(s) = \max_k \frac{\sum_{i=1}^k Q(s,a_i) - \sigma}{k},$$

where $\sigma = \alpha_s + \gamma\beta_s\kappa_\infty(v), Q(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v(s')$, and $Q(s,a_1) \ge Q(s,a_2), \ge \cdots \ge Q(s,a_A)$.

Proof. Follows from section K.1. □

Corollary L.11. *The* s *rectangular L_p robust Bellman operator can be evaluated for $p = 1, 2$ by algorithm 8 and algorithm 7 respectively.*

Proof. It follows from the algorithm 6, where we solve the linear equation and quadratic equation for $p = 1, 2$ respectively. For $p = 2$, it can be found in (Anava & Levy, 2016). $\square$

Algorithm 7 Algorithm to compute S-rectangular L_2 robust optimal Bellman Operator

```
1: Input: $\sigma = \alpha_s + \gamma\beta_s\kappa_2(v)$,    $Q(s,a) = R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s')$.
2: Output $(\mathcal{T}^*_{\mathcal{U}^s_2} v)(s), \chi_2(v,s)$
3: Sort $Q(s,\cdot)$ and label actions such that $Q(s,a_1) \geq Q(s,a_2), \cdots$.
4: Set initial value guess $\lambda_1 = Q(s,a_1) - \sigma$ and counter $k = 1$.
5: while $k \leq A-1$ and $\lambda_k \leq Q(s,a_k)$ do
6:     Increment counter: $k = k+1$
7:     Update value estimate:
           $$\lambda_k = \frac{1}{k}\Big[\sum_{i=1}^{k} Q(s,a_i) - \sqrt{k\sigma^2 + (\sum_{i=1}^{k} Q(s,a_i))^2 - k\sum_{i=1}^{k}(Q(s,a_i))^2}\Big]$$
8: end while
9: Return: $\lambda_k, k$
```

Algorithm 8 Algorithm to compute S-rectangular L_1 robust optimal Bellman Operator

```
1: Input: $\sigma = \alpha_s + \gamma\beta_s\kappa_\infty(v)$,    $Q(s,a) = R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a)v(s')$.
2: Output $(\mathcal{T}^*_{\mathcal{U}^s_1} v)(s), \chi_1(v,s)$
3: Sort $Q(s,\cdot)$ and label actions such that $Q(s,a_1) \geq Q(s,a_2), \cdots$.
4: Set initial value guess $\lambda_1 = Q(s,a_1) - \sigma$ and counter $k = 1$.
5: while $k \leq A-1$ and $\lambda_k \leq Q(s,a_k)$ do
6:     Increment counter: $k = k+1$
7:     Update value estimate:
           $$\lambda_k = \frac{1}{k}\Big[\sum_{i=1}^{k} Q(s,a_i) - \sigma\Big]$$
8: end while
9: Return: $\lambda_k, k$
```

M. Time Complexity

In this section, we will discuss the time complexity of various robust MDPs and compare it with the time complexity of non-robust MDPs. We assume that we have the knowledge of the nominal transition kernel and nominal reward function for robust MDPs, and in the case of non-robust MDPs, we assume the knowledge of the transition kernel and reward function. We divide the discussion into various parts depending on their similarity.

M.1. Exact Value Iteration: Best Response

In this section, we will discuss non-robust MDPs, (sa)-rectangular $L_1/L_2/L_\infty$ robust MDPs and s-rectangular L_∞ robust MDPs. They all have a common theme for value iteration as follows, for the value function v, their Bellman operator ($\mathcal{T}$) evaluation is done as

$$(\mathcal{T}v)(s) = \underbrace{\max_a}_{\text{action cost}} \Big[R(s,a) + \alpha_{s,a} \underbrace{\kappa(v)}_{\text{reward penalty/cost}} \underbrace{+\gamma \sum_{s'} P(s'|s,a)v(s')}_{\text{sweep}} \Big]. \tag{82}$$

'Sweep' requires $O(S)$ iterations and 'action cost' requires $O(A)$ iterations. Note that the reward penalty $\kappa(v)$ doesn't depend on state and action. It is calculated only once for value iteration for all states. The above value update has to be done for each states, so one full update requires

$$O\Big(\ S(\text{action cost})(\text{sweep cost}\ ) + \text{reward cost}\ \Big) = O\Big(\ S^2A + \text{reward cost}\ \Big)$$

Since the value iteration is a contraction map, to get ϵ-close to the optimal value, it requires $O(\log(\frac{1}{\epsilon}))$ full value update, so the complexity is

$$O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S^2A + \text{reward cost}\ \big)\Big)\ .$$

1. **Non-robust MDPs**: The cost of 'reward is zero as there is no regularizer to compute. The total complexity is

$$O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S^2A + 0\ \big)\Big) = O\Big(\ \log(\frac{1}{\epsilon})S^2A\ \Big)\ .$$

2. (sa)**-rectangular $L_1/L_2/L_\infty$ and s-rectangular L_∞ robust MDPs**: We need to calculate the reward penalty $(\kappa_1(v)/\kappa_2(v)/\kappa_\infty)$ that takes $O(S)$ iterations. As calculation of mean, variance, and median, all are linear time compute. Hence the complexity is

$$O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S^2A + S\ \big)\Big) = O\Big(\ \log(\frac{1}{\epsilon})S^2A\ \Big)\ .$$

M.2. Exact Value iteration: Top k response

In this section, we discuss the time complexity of s-rectangular L_1/L_2 robust MDPs as in algorithm 4. We need to calculate the reward penalty ($\kappa_\infty(v)/\kappa_2(v)$ in (26)) that takes $O(S)$ iterations. Then for each state we do: sorting of Q-values in (29), value evaluation in (30), update Q-value in (28) that takes $O(A\log(A)), O(A), O(SA)$ iterations respectively. Hence the complexity is

$$= \text{total iteration(reward cost (26) + S(sorting (29) + value evaluation (30) +Q-value(28))}$$

$$= \log(\frac{1}{\epsilon})(S + S(A\log(A) + A + SA)$$

$$O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S^2A + SA\log(A)\ \big)\Big)\ .$$

For general p, we need little caution as $k_p(v)$ can't be calculated exactly but approximately by binary search. It is the subject of discussion for the next sections.

M.3. Inexact Value Iteration: (sa)-rectangular L_p robust MDPs ($\mathcal{U}_p^{\mathtt{sa}}$)

In this section, we will study the time complexity for robust value iteration for (sa)-rectangular L_p robust MDPs for general p. Recall, that value iteration takes the best-penalized action, which is easy to compute. But reward penalization depends on p-variance measure $\kappa_p(v)$, which we will estimate by $\hat{\kappa}_p(v)$ through binary search. We have inexact value iterations as

$$v_{n+1}(s) := \max_{a\in\mathcal{A}}[\alpha_{sa} - \gamma\beta_{sa}\hat{\kappa}_q(v_n) + R_0(s,a) + \gamma\sum_{s'}P_0(s'|s,a)v_n(s')]$$

where $\hat{\kappa}_q(v_n)$ is a ϵ_1 approximation of $\kappa_q(v_n)$, that is $|\hat{\kappa}_q(v_n) - \kappa_q(v_n)| \le \epsilon_1$. Then it is easy to see that we have bounded error in robust value iteration, that is

$$\|v_{n+1} - \mathcal{T}^*_{\mathcal{U}_p^{\mathtt{sa}}}v_n\|_\infty \le \gamma\beta_{max}\epsilon_1$$

where $\beta_{max} := \max_{s,a}\beta_{s,a}$

Proposition M.1. *Let $\mathcal{T}^*_{\mathcal{U}}$ be a γ contraction map, and v^* be its fixed point. And let $\{v_n, n \geq 0\}$ be approximate value iteration, that is*

$$\|v_{n+1} - \mathcal{T}^*_{\mathcal{U}} v_n\|_\infty \leq \epsilon$$

then

$$\lim_{n\to\infty} \|v_n - v^*\|_\infty \leq \frac{\epsilon}{1-\gamma}$$

moreover, it converges to the $\frac{\epsilon}{1-\gamma}$ radius ball linearly, that is

$$\|v_n - v^*\|_\infty - \frac{\epsilon}{1-\gamma} \leq c\gamma^n$$

where $c = \frac{1}{1-\gamma}\epsilon + \|v_0 - v^*\|_\infty$.

Proof.

$$\begin{aligned}
\|v_{n+1} - v^*\|_\infty =& \|v_{n+1} - \mathcal{T}^*_{\mathcal{U}} v^*\|_\infty \\
=& \|v_{n+1} - \mathcal{T}^*_{\mathcal{U}} v_n + \mathcal{T}^*_{\mathcal{U}} v_n - \mathcal{T}^*_{\mathcal{U}} v^*\|_\infty \\
\leq& \|v_{n+1} - \mathcal{T}^*_{\mathcal{U}} v_n\|_\infty + \|\mathcal{T}^*_{\mathcal{U}} v_n - \mathcal{T}^*_{\mathcal{U}} v^*\|_\infty \\
\leq& \|v_{n+1} - \mathcal{T}^*_{\mathcal{U}} v_n\|_\infty + \gamma\|v_n - v^*\|_\infty, \qquad \text{(contraction)} \\
\leq& \epsilon + \gamma\|v_n - v^*\|_\infty, \qquad \text{(approximate value iteration)} \\
\implies \|v_n - v^*\|_\infty =& \sum_{k=0}^{n-1} \gamma^k \epsilon + \gamma^n \|v_0 - v^*\|_\infty, \qquad \text{(unrolling above recursion)} \\
=& \frac{1-\gamma^n}{1-\gamma}\epsilon + \gamma^n\|v_0 - v^*\|_\infty \\
=& \gamma^n[\frac{1}{1-\gamma}\epsilon + \|v_0 - v^*\|_\infty] + \frac{\epsilon}{1-\gamma}
\end{aligned} \tag{83}$$

Taking limit $n \to \infty$ both sides, we get

$$\lim_{n\to\infty} \|v_n - v^*\|_\infty \leq \frac{\epsilon}{1-\gamma}.$$

□

Lemma M.2. *For $\mathcal{U}^{\mathtt{sa}}_p$, the total iteration cost is $\log(\frac{1}{\epsilon})S^2A + (\log(\frac{1}{\epsilon}))^2$ to get ϵ close to the optimal robust value function.*

Proof. We calculate $\kappa_q(v)$ with $\epsilon_1 = \frac{(1-\gamma)\epsilon}{3}$ tolerance that takes $O(S\log(\frac{S}{\epsilon_1}))$ using binary search (see section J.3). Now, we do approximate value iteration for $n = \log(\frac{3\|v_0 - v^*\|_\infty}{\epsilon})$. Using the above lemma, we have

$$\begin{aligned}
\|v_n - v^*_{\mathcal{U}^{\mathtt{sa}}_p}\|_\infty =& \gamma^n[\frac{1}{1-\gamma}\epsilon_1 + \|v_0 - v^*_{\mathcal{U}^{\mathtt{sa}}_p}\|_\infty] + \frac{\epsilon_1}{1-\gamma} \\
\leq& \gamma^n[\frac{\epsilon}{3} + \|v_0 - v^*_{\mathcal{U}^{\mathtt{sa}}_p}\|_\infty] + \frac{\epsilon}{3} \\
\leq& \gamma^n\frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \leq \epsilon.
\end{aligned} \tag{84}$$

In summary, we have action cost $O(A)$, reward cost $O(S\log(\frac{S}{\epsilon}))$, sweep cost $O(S)$ and total number of iterations $O(\log(\frac{1}{\epsilon}))$. So the complexity is

$$\text{(number of iterations)}\big(\text{S(actions cost) (sweep cost) + reward cost}\big)$$

$$= \log(\frac{1}{\epsilon})\ \big(\ S^2A + S\log(\frac{S}{\epsilon})\ \big) = \log(\frac{1}{\epsilon})(S^2A + S\log(\frac{1}{\epsilon}) + S\log(S))$$

$$= \log(\frac{1}{\epsilon})S^2A + S(\log(\frac{1}{\epsilon}))^2$$

□

M.4. Inexact Value Iteration: s-rectangular L_p robust MDPs ($\mathcal{U}^{\mathtt{s}}_p$)

In this section, we study the time complexity for robust value iteration for $\mathtt{s}$-rectangular L_p robust MDPs for general p (algorithm 3). Recall, that value iteration takes regularized actions and penalized rewards. And reward penalization depends on q-variance measure $\kappa_q(v)$, that we will estimate by $\hat{\kappa}_q(v)$ through binary search, then again we will calculate $\mathcal{T}^*_{\mathcal{U}^{\mathtt{sa}}_p}$ by binary search with approximated $\kappa_q(v)$. Here, we have two error sources ((26), (30)) as a contrast to ($\mathtt{sa}$)-rectangular cases, where there was only one error source from the estimation of κ_q.

First, we account for the error caused by the first source (κ_q). Here we do value iteration with approximated q-variance $\hat{\kappa}_q$, and exact action regularizer. We have

$$v_{n+1}(s) := \lambda \quad \text{s.t.} \quad \alpha_s + \gamma\beta_s\hat{\kappa}_q(v) = (\sum_{Q(s,a)\geq\lambda}(Q(s,a)-\lambda)^p)^{\frac{1}{p}}$$

where $Q(s,a) = R_0(s,a) + \gamma\sum_{s'}P_0(s'|s,a)v_n(s')$, and $|\hat{\kappa}_q(v_n) - \kappa_q(v_n)| \leq \epsilon_1$. Then from the next result (proposition M.3), we get

$$\|v_{n+1} - \mathcal{T}^*_{\mathcal{U}^{\mathtt{sa}}_p}v_n\|_\infty \leq \gamma\beta_{max}\epsilon_1$$

where $\beta_{max} := \max_{s,a}\beta_{s,a}$

Proposition M.3. *Let $\hat{\kappa}$ be an an ϵ-approximation of κ, that is $|\hat{\kappa} - \kappa| \leq \epsilon$, and let $b \in \mathbb{R}^A$ be sorted component wise, that is, $b_1 \geq, \cdots, \geq b_A$. Let λ be the solution to the following equation with exact parameter κ,*

$$\alpha + \gamma\beta\kappa = (\sum_{b_i\geq\lambda}|b_i-\lambda|^p)^{\frac{1}{p}}$$

and let $\hat{\lambda}$ be the solution of the following equation with approximated parameter $\hat{\kappa}$,

$$\alpha + \gamma\beta\hat{\kappa} = (\sum_{b_i\geq\hat{\lambda}}|b_i-\hat{\lambda}|^p)^{\frac{1}{p}},$$

then $\hat{\lambda}$ is an $O(\epsilon)$-approximation of λ, that is

$$|\lambda - \hat{\lambda}| \leq \gamma\beta\epsilon.$$

Proof. Let the function $f : [b_A, b_1] \to \mathbb{R}$ be defined as

$$f(x) := (\sum_{b_i\geq x}|b_i - x|^p)^{\frac{1}{p}}.$$

We will show that the derivative of f is bounded, implying its inverse is bounded, and hence Lipschitz, that will prove the claim. Let proceed

$$\begin{aligned}\frac{df(x)}{dx} &= -(\sum_{b_i\geq x}|b_i-x|^p)^{\frac{1}{p}-1}\sum_{b_i\geq x}|b_i-x|^{p-1}\\ &= -\frac{\sum_{b_i\geq x}|b_i-x|^{p-1}}{(\sum_{b_i\geq x}|b_i-x|^p)^{\frac{p-1}{p}}}\\ &= -\Big[\frac{(\sum_{b_i\geq x}|b_i-x|^{p-1})^{\frac{1}{p-1}}}{(\sum_{b_i\geq x}|b_i-x|^p)^{\frac{1}{p}}}\Big]^{p-1}\\ &\leq -1.\end{aligned} \tag{85}$$

The inequality follows from the following relation between L_p norm,

$$\|x\|_a \geq \|x\|_b, \qquad \forall 0 \leq a \leq b.$$

It is easy to see that the function f is strictly monotone in the range $b_A, b_1]$, so its inverse is well defined in the same range. Then derivative of the inverse of the function f is bounded as

$$0 \geq \frac{d}{dx}f^{-}(x) \geq -1.$$

Now, observe that $\lambda = f^{-}(\alpha + \gamma\beta\kappa)$ and $\hat{\lambda} = f^{-}(\alpha + \gamma\beta\hat{\kappa})$, then by Lipschitzcity, we have

$$|\lambda - \hat{\lambda}| = |f^{-}(\alpha + \gamma\beta\kappa) - f^{-}(\alpha + \gamma\beta\hat{\kappa})| \leq \gamma\beta| - \kappa - \hat{\kappa})| \leq \gamma\beta\epsilon.$$

□

Lemma M.4. *For $\mathcal{U}^{s}_{p}$, the total iteration cost is $O\Big(\log(\frac{1}{\epsilon})\,\big(S^2A + SA\log(\frac{A}{\epsilon})\,\big)\Big)$ to get ϵ close to the optimal robust value function.*

Proof. We calculate $\kappa_q(v)$ in (26) with $\epsilon_1 = \frac{(1-\gamma)\epsilon}{6}$ tolerance that takes $O(S\log(\frac{S}{\epsilon_1}))$ iterations using binary search (see section J.3). Then for every state, we sort the Q values (as in (29)) that cost $O(A\log(A))$ iterations. In each state, to update value, we do again binary search with approximate $\kappa_q(v)$ up to $\epsilon_2 := \frac{(1-\gamma)\epsilon}{6}$ tolerance, that takes $O(\log(\frac{1}{\epsilon_2}))$ search iterations and each iteration cost $O(A)$, altogether it costs $O(A\log(\frac{1}{\epsilon_2}))$ iterations. Sorting of actions and binary search adds up to $O(A\log(\frac{A}{\epsilon}))$ iterations (action cost). So we have (doubly) approximated value iteration as follows,

$$|v_{n+1}(s) - \hat{\lambda}| \leq \epsilon_1 \tag{86}$$

where

$$(\alpha_s + \gamma\beta_s\hat{\kappa}_q(v_n))^p = \sum_{Q_n(s,a)\geq\hat{\lambda}} (Q_n(s,a) - \hat{\lambda})^p$$

and

$$Q_n(s,a) = R_0(s,a) + \gamma\sum_{s'} P_0(s'|s,a)v_n(s'), \qquad |\hat{\kappa}_q(v_n) - \kappa_q(v_n)| \leq \epsilon_1.$$

And we do this approximate value iteration for $n = \log(\frac{3\|v_0 - v^*\|_\infty}{\epsilon})$. Now, we do error analysis. By accumulating error, we have

$$\begin{aligned}
|v_{n+1}(s) - (\mathcal{T}^*_{\mathcal{U}^s_p}v_n)(s)| \leq& |v_{n+1}(s) - \hat{\lambda}| + |\hat{\lambda} - (\mathcal{T}^*_{\mathcal{U}^s_p}v_n)(s)| \\
\leq& \epsilon_1 + |\hat{\lambda} - (\mathcal{T}^*_{\mathcal{U}^s_p}v_n)(s)|, \qquad \text{(by definition)} \\
\leq& \epsilon_1 + \gamma\beta_{\max}\epsilon_1, \qquad \text{(from proposition M.3)} \\
\leq& 2\epsilon_1.
\end{aligned} \tag{87}$$

where $\beta_{max} := \max_s \beta_s, \gamma \leq 1$.

Now, we do approximate value iteration, and from proposition M.1, we get

$$\|v_n - v^*_{\mathcal{U}^s_p}\| \leq \frac{2\epsilon_1}{1-\gamma} + \gamma^n[\frac{1}{1-\gamma}2\epsilon_1 + \|v_0 - v^*_{\mathcal{U}^s_p}\|_\infty] \tag{88}$$

Now, putting the value of n, we have

$$\begin{aligned}
\|v_n - v^*_{\mathcal{U}^s_p}\|_\infty =& \gamma^n[\frac{2\epsilon_1}{1-\gamma} + \|v_0 - v^*_{\mathcal{U}^s_p}\|_\infty] + \frac{2\epsilon_1}{1-\gamma} \\
\leq& \gamma^n[\frac{\epsilon}{3} + \|v_0 - v^*_{\mathcal{U}^s_p}\|_\infty] + \frac{\epsilon}{3} \\
\leq& \gamma^n\frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \leq \epsilon.
\end{aligned} \tag{89}$$

To summarize, we do $O(\log(\frac{1}{\epsilon}))$ full value iterations. Cost of evaluating reward penalty is $O(S\log(\frac{S}{\epsilon}))$. For each state: evaluation of Q-value from value function requires $O(SA)$ iterations, sorting the actions according Q-values requires $O(A\log(A))$ iterations, and binary search for evaluation of value requires $O(A\log(1/\epsilon)$. So the complexity is

$$O(\text{(total iterations)(reward cost + S(Q-value + sorting + binary search for value)))}$$

$$= O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S\log(\frac{S}{\epsilon}) + S(SA + A\log(A) + A\log(\frac{1}{\epsilon}))\ \big)\Big)$$

$$= O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S\log(\frac{1}{\epsilon}) + S\log(S) + S^2A + SA\log(A) + SA\log(\frac{1}{\epsilon})\ \big)\Big)$$

$$= O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S^2A + SA\log(A) + SA\log(\frac{1}{\epsilon})\ \big)\Big)$$

$$= O\Big(\ \log(\frac{1}{\epsilon})\ \big(\ S^2A + SA\log(\frac{A}{\epsilon})\ \big)\Big)$$

□