Robust ϕ-Divergence MDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract

In recent years, robust Markov decision processes (MDPs) have emerged as a prominent modeling framework for dynamic decision problems affected by uncertainty. In contrast to classical MDPs, which only account for *stochasticity* by modeling the dynamics through a stochastic process with a known transition kernel, robust MDPs additionally account for *ambiguity* by optimizing in view of the most adverse transition kernel from a prescribed ambiguity set. In this paper, we develop a novel solution framework for robust MDPs with s-rectangular ambiguity sets that decomposes the problem into a sequence of robust Bellman updates and simplex projections. Exploiting the rich structure present in the simplex projections corresponding to ϕ-divergence ambiguity sets, we show that the associated s-rectangular robust MDPs can be solved substantially faster than with state-of-the-art commercial solvers as well as a recent first-order solution scheme, thus rendering them attractive alternatives to classical MDPs in practical applications.

1 Introduction

Markov decision processes (MDPs) are a flexible and popular framework for dynamic decision-making problems and reinforcement learning [33, 42]. A practical limitation of the standard MDP model is that it assumes the model parameters, such as transition probabilities and rewards, to be known exactly. In reinforcement learning and other applications, these parameters must be estimated from sampled data, which introduces estimation errors. Optimal MDP solutions, referred to as policies, are well known to be sensitive to errors and may fail catastrophically when deployed [25, 47].

Robust MDPs (RMDPs) mitigate the sensitivity of MDPs to estimation errors by computing a policy that is optimal for the worst plausible realization of the transition probabilities. This set of plausible transition probabilities is known as the *ambiguity set*. Most prior work considers ambiguity sets that are rectangular. In this work, we focus on *s-rectangular ambiguity sets*, which assume that the worst transition probabilities are chosen independently in each state [25, 47]. While several other models of rectangularity have been studied [9, 14, 22, 27], s-rectangular ambiguity sets are popular due to their generality and the existence of polynomial-time algorithms based on dynamic programming concepts. However, even those algorithms may be too slow in practice. Solving RMDPs requires the solution of a convex optimization problem in every step of value or policy iteration, which can become prohibitively slow even in moderatly sized problems with 100s of states [5, 9, 15, 20].

Motivated by the difficulty of solving RMDPs, several fast algorithms have been proposed for s-rectangular RMDPs [5, 9, 15, 20]. The preponderance of the earlier work has focused on ambiguity sets defined in terms of L_1- and L_∞-norms. These ambiguity sets are polyhedral, and they can be analyzed using linear programming techniques which offer fruitful avenues to exploit the structure inherent to those sets. However, recent statistical studies point to the superior solution quality offered by nonlinear ambiguity sets defined in terms of the Kullback-Leibler (KL) divergence, the L_2-norm and other metrics [18]. RMDPs with s-rectangular ambiguity sets defined in terms of non-polyhedral

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

ambiguity sets are currently solved using first-order methods [15] or general convex conic solvers such as MOSEK [3], which tend to be complex, closed-source and slow.

As our main contribution, we propose a new suite of fast algorithms for solving RMDPs with ϕ-divergence constrained s-rectangular ambiguity sets. ϕ-divergences, also known as f-divergences, constitute a generalization of the KL divergence that encompasses the Burg entropy as well as the L_1- and weighted L_2-norms as special cases [4, 6]. Solving ϕ-divergence RMDPs using value iteration requires the solution of seemingly unstructured min-max problems. Our main insight is that these min-max problems can be reduced to a small number of highly structured projection problems onto a probability simplex. We use this insight to develop tailored solution schemes for the projection problems corresponding to several popular ϕ-divergence ambiguity sets, which in turn give rise to efficient solution methods for the respective RMDPs. Ignoring tolerances, our algorithms achieve an overall $\mathcal{O}(S^2 \cdot A \log A)$ or $\mathcal{O}(S^2 \log S \cdot A)$ time complexity to compute the robust Bellman operator, where S and A denote the numbers of states and actions, respectively. Since the evaluation of a non-robust Bellman operator requires a runtime of $\mathcal{O}(S^2 \cdot \log A)$, our algorithms only incur an additional logarithmic overhead to account for robustness in the transition probabilities. This computational complexity compares favorably with the larger time complexity of a recent first-order solution scheme for KL divergence-constrained s-rectangular RMDPs (which we will elaborate on later in the paper) as well as a minimum complexity of $\mathcal{O}(S^{4.5} \cdot A)$ for the naïve solution with state-of-the-art interior-point algorithms. Our framework is general enough to readily accommodate for ϕ-divergences that have not been studied previously in the context of s-rectangular ambiguity sets, such as the Burg entropy and the χ^2-distance. For other ϕ-divergences, such as the L_1-norm, our framework results in the same complexity at substantially simplified proofs.

The algorithms developed in this paper can be used in combination with a variety of RMDP solution schemes. In particular, they can be used to accelerate the standard robust value iteration, policy iteration, modified policy iteration [23] and partial policy iteration [20]. They can also be combined with a first order gradient method [15] that has been introduced recently. In addition, fast algorithms for computing the Bellman operator also play a crucial role when scaling robust algorithms to value function optimization [44] and robust policy gradients [43].

The remainder of the paper proceeds as follows. Section 2 reviews relevant prior work and Section 3 describes our basic RMDP setting. Then, Section 4 shows how the robust Bellman operator for a large class of ambiguity sets can be reduced to a sequence of structured projections onto a simplex. We describe novel algorithms for efficiently computing the simplex projections for several ϕ-divergences in Section 5. Finally, Section 6 presents experimental results that compare the runtime of our algorithms with general conic solvers as well as a recent first-order optimization algorithm [15].

Notation. We denote by $\mathbf{e}$ the vector of all ones, whose context determines its dimension. We refer to the probability simplex in $\mathbb{R}^n$ by $\Delta_n = \{\boldsymbol{p} \in \mathbb{R}^n_+ \,:\, \mathbf{e}^\top \boldsymbol{p} = 1\}$. For $\boldsymbol{x} \in \mathbb{R}^n$, we let $\min\{\boldsymbol{x}\} = \min\{x_i \,:\, i = 1, \ldots, n\}$ (similar for the maximum operator), and we define $[\boldsymbol{x}]_+ \in \mathbb{R}^n_+$ component-wise as $([\boldsymbol{x}]_+)_i = \max\{x_i,\, 0\}$, $i = 1, \ldots, n$. We refer to the conjugate of a function $f : \mathbb{R}^n \to \mathbb{R}$ by $f^\star(\boldsymbol{y}) = \sup\{\boldsymbol{y}^\top \boldsymbol{x} - f(\boldsymbol{x}) \,:\, \boldsymbol{x} \in \mathbb{R}^n\}$. Random variables are indicated by a tilde.

2 Related Work

While RMDPs have been studied since the seventies [39], they have witnessed significant recent interest due to their widespread adoption in applications ranging from assortment optimization [37], medical decision-making [13, 53] and hospital operations management [17], production planning [49] and energy systems [21] to model predictive control [11], aircraft collision avoidance [24], wireless communications [48] and the robustification against approximation errors in aggregated MDPs [31].

Efficient implementations of the robust value iteration have been first proposed by [12, 22, 29] for RMDPs with (s, a)-rectangular ambiguity sets, where the worst transition probabilities are considered separately for each state and action. The authors study ambiguity sets that bound the distance of the transition probabilities to some nominal distribution in terms of finite scenarios, interval matrix bounds, ellipsoids, the relative entropy, the KL divergence and maximum a posteriori models. Subsequently, similar methods have been developed by [48] for interval matrix bounds as well as likelihood uncertainty models, by [31] for 1-norm ambiguity sets as well as by [53] for interval matrix bounds intersected with a budget constraint. All of these contributions have in common that they focus on (s, a)-rectangular ambiguity sets where the existence of optimal deterministic policies is guaranteed, and it is not clear how they could be extended to the more general class of s-rectangular ambiguity sets where all optimal policies may be randomized.

In contrast to (s, a)-rectangular ambiguity sets, s-rectangular ambiguity sets restrict the conservatism among transition probabilities corresponding to different actions in the same state, which tends to lead to a superior performance in data-driven settings. [47] solve the subproblems arising in the robust value iteration of an s-rectangular RMDP as linear or conic optimization problems using commercial off-the-shelf solvers. Despite their polynomial-time complexity, general-purpose solvers cannot exploit the structure present in these subproblems, which renders them suitable primarily for small problem instances. More efficient tailored solution methods for s-rectangular RMDPs have subsequently been developed by [5, 19, 20]. [19] develop a homotopy continuation method for RMDPs with (s, a)-rectangular and s-rectangular weighted 1-norm ambiguity sets, while [5] adapt the algorithm of [19] to unweighted ∞-norm ambiguity sets. [20] embed the algorithms of [19] in a partial policy iteration, which generalizes the robust modified policy iteration proposed by [23] for (s, a)-rectangular RMDPs to s-rectangular RMDPs.

While the present paper focuses on the robust value iteration for ease of exposition, we note that our algorithms can also be combined with the partial policy iteration of [20] to obtain further speedups. [9] establish a relationship between s-rectangular RMDPs and twice regularized MDPs, which they subsequently use to propose efficient Bellman updates for a modified policy iteration. While their approach can solve RMDPs in almost the same time as a classical non-robust MDPs, the obtained policies can be conservative as the worst-case transition probabilities are not restricted to reside in a probability simplex and, therefore, may be negative and/or add up to more or less than 1. Finally, [15] propose a first-order framework for RMDPs with s-rectangular KL and spherical ambiguity sets that interleaves primal-dual first-order updates with approximate value iteration steps. The authors show that their algorithms outperform a robust value iteration that solves the emerging subproblems using state-of-the-art commercial solvers. We compare our solution method for KL ambiguity sets with the approach proposed by [15] in terms of its theoretical complexity and numerical runtimes.

While this paper exclusively studies s-rectangular uncertainty sets, alternative generalizations of (s, a)-rectangular ambiguity sets have been proposed in the literature as well. For example, [27] consider k-rectangular ambiguity sets where the transition probabilities of different states can be coupled, [14] study factor ambiguity model ambiguity sets where the transition probabilities depend on a small number of underlying factors, and [45] construct ambiguity sets that bound marginal moments of state-action features defined over entire MDP trajectories. We also note the papers [7, 16, 51] which study the related problem of *distributionally* robust MDPs whose transition probabilities are themselves regarded as random objects that are drawn from distributions which are only partially known. The connections between RMDPs and multi-stage stochastic programs as well as distributionally robust problems are explored further by [38, 40, 41].

3 Preliminaries

Robust MDPs We study RMDPs with a finite state space $\mathcal{S} = \{1, \dots, S\}$ and a finite action space $\mathcal{A} = \{1, \dots, A\}$. We assume an infinite planning horizon, but all of our results immediately extend to a finite time horizon. Without loss of generality, we assume that every action $a \in \mathcal{A}$ is admissible in every state $s \in \mathcal{S}$. The RMDP starts in a random initial state $\tilde{s}_0$ that follows the known probability distribution $\boldsymbol{p}^0$ from the probability simplex Δ_S in $\mathbb{R}^S$. If action $a \in \mathcal{A}$ is taken in state $s \in \mathcal{S}$, then the RMDP transitions randomly to the next state according to the conditional probability distribution $\boldsymbol{p}_{sa} \in \Delta_S$. We condense the transition probabilities $\boldsymbol{p}_{sa}$ to the tensor $\boldsymbol{p} \in (\Delta_S)^{S \times A}$. The transition probabilities are only known to reside in a non-empty, compact ambiguity set $\mathcal{P} \subseteq (\Delta_S)^{S \times A}$. For a transition from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ under action $a \in \mathcal{A}$, the decision maker receives an expected reward of $r_{sas'} \in \mathbb{R}_+$. As with the transition probabilities, we condense these rewards to the tensor $\boldsymbol{r} \in \mathbb{R}_+^{S \times A \times S}$. Without loss of generality, we assume that all rewards are non-negative.

We denote by $\Pi = (\Delta_A)^S$ the set of all stationary (*i.e.*, time-independent) randomized policies. A policy $\boldsymbol{\pi} \in \Pi$ takes action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ with probability π_{sa}. The transition probabilities $\boldsymbol{p} \in \mathcal{P}$ and the policy $\boldsymbol{\pi} \in \Pi$ induce a stochastic process $\{(\tilde{s}_t, \tilde{a}_t)\}_{t=0}^{\infty}$ on the space $(\mathcal{S} \times \mathcal{A})^\infty$ of sample paths. We refer by $\mathbb{E}^{\boldsymbol{p},\boldsymbol{\pi}}$ to expectations with respect to this process. The decision maker is risk-neutral but ambiguity-averse and wishes to maximize the worst-case expected total reward under

Divergence	$d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa})$	$\phi(t)$	Complexity of $\mathfrak{J}$	State-of-the-Art
Kullback-Leibler	$\sum_{s'} p_{sas'} \log\left(\frac{p_{sas'}}{\overline{p}_{sas'}}\right)$	$t \log t - t + 1$	$\mathcal{O}(S^2 \cdot A \log A)$	$\mathcal{O}(\ell^2 \cdot S^2 \cdot A)$
Burg Entropy	$\sum_{s'} \overline{p}_{sas'} \log\left(\frac{\overline{p}_{sas'}}{p_{sas'}}\right)$	$-\log t + t - 1$	$\mathcal{O}(S^2 \cdot A \log A)$	no poly-time guarantee
Variation Distance	$\sum_{s'} \lvert p_{sas'} - \overline{p}_{sas'} \rvert$	$\lvert t - 1 \rvert$	$\mathcal{O}(S^2 \log S \cdot A)$	$\mathcal{O}(S^2 \log S \cdot A)$
χ^2-Distance	$\sum_{s'} \frac{(p_{sas'} - \overline{p}_{sas'})^2}{\overline{p}_{sas'}}$	$(t-1)^2$	$\mathcal{O}(S^2 \log S \cdot A)$	$\mathcal{O}(S^{4.5} \cdot A)$

Table 1: Summary of the ϕ-divergences studied in this paper, together with the complexity of our robust Bellman operator $\mathfrak{J}$ (applied across all states $s \in \mathcal{S}$) as well as the best known results from the literature. The complexity estimates omit constants and tolerances that are reported in Section 5 of the paper. 'ℓ', where present, refers to the number of Bellman iterations conducted so far.

a discount factor $\lambda \in (0, 1)$,

$$\max_{\boldsymbol{\pi} \in \Pi} \min_{\boldsymbol{p} \in \mathcal{P}} \mathbb{E}^{\boldsymbol{p}, \boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \lambda^t \cdot r_{\tilde{s}_t, \tilde{a}_t, \tilde{s}_{t+1}} \;\middle|\; \tilde{s}_0 \sim \boldsymbol{p}^0 \right]. \tag{1}$$

Note that the maximum and minimum in (1) are both attained by the Weierstrass theorem since Π and $\mathcal{P}$ are non-empty and compact, while the objective function is finite since $\lambda < 1$.

Rectangular Ambiguity Sets For general ambiguity sets $\mathcal{P}$, evaluating the inner minimization in (1) is NP-hard even if the policy $\boldsymbol{\pi} \in \Pi$ is fixed [47]. For these reasons, much of the research on RMDPs and their applications has focused on rectangular ambiguity sets. Among the most general rectangular ambiguity sets are the s-rectangular ambiguity sets $\mathcal{P}$ satisfying

$$\mathcal{P} = \left\{ \boldsymbol{p} \in (\Delta_S)^{S \times A} \,:\, \boldsymbol{p}_s \in \mathcal{P}_s \;\; \forall s \in \mathcal{S} \right\}, \quad \text{where} \quad \mathcal{P}_s \subseteq (\Delta_S)^A, \; s \in \mathcal{S},$$

see [25, 47, 50, 52]. In contrast to the simpler class of (s, a)-rectangular ambiguity sets, s-rectangular ambiguity sets restrict the choice of transition probabilities $\boldsymbol{p}_{s1}, \ldots, \boldsymbol{p}_{sA}$ corresponding to different actions a applied in the same state s. This limits the conservatism of the resulting RMDP (1) and typically leads to a better performance of the optimal policy [47]. Although Bellman's optimality principle extends to s-rectangular RMDPs and there is always an optimal stationary policy, all optimal policies of an s-rectangular RMDP may be randomized.

We study a new general class of s-rectangular ambiguity sets that can be expressed as

$$\mathcal{P}_s = \left\{ \boldsymbol{p}_s \in (\Delta_S)^A \,:\, \sum_{a \in \mathcal{A}} d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa}) \leq \kappa \right\}, \tag{2}$$

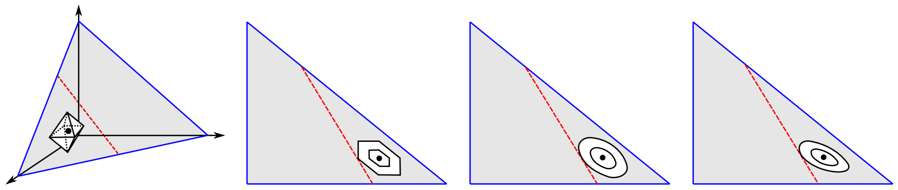
where $\kappa \in \mathbb{R}_+$ is the *uncertainty budget* and the distance functions $d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa})$, $a \in \mathcal{A}$, are *ϕ-divergences* (also known as *f-divergences*) satisfying

$$d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa}) = \sum_{s' \in \mathcal{S}} \overline{p}_{sas'} \phi\left(\frac{p_{sas'}}{\overline{p}_{sas'}}\right).$$

Here, $\phi \colon \mathbb{R}_+ \to \mathbb{R}_+$ is a convex function satisfying $\phi(1) = 0$. Intuitively, a ϕ-divergence measures the distance between two probability distributions. With an appropriate choice of ϕ, it generalizes other metrics including the KL divergence, the Burg entropy, L_1- and L_2-norms and others [4, 6]. Table 1 reports some popular ϕ-divergences that we study in this paper. Note that the variation distance coincides with the L_1-based s-rectangular ambiguity sets studied in earlier work [19, 20].

Robust Value Iteration A standard approach for computing the optimal value and the optimal policy of an RMDP (1) is the robust value iteration [22, 29, 25, 47]: Starting with an initial estimate $\boldsymbol{v}^0 \in \mathbb{R}^S$ of the state-wise optimal value to-go, we conduct robust Bellman iterations of the form $\boldsymbol{v}^{t+1} \leftarrow \mathfrak{J}(\boldsymbol{v}^t)$, $t = 0, 1, \ldots$, where the robust Bellman operator $\mathfrak{J}$ is defined component-wise as

$$[\mathfrak{J}(\boldsymbol{v})]_s \;=\; \max_{\boldsymbol{\pi}_s \in \Delta_A} \min_{\boldsymbol{p}_s \in \mathcal{P}_s} \sum_{a \in \mathcal{A}} \pi_{sa} \cdot {\boldsymbol{p}_{sa}}^\top (\boldsymbol{r}_{sa} + \lambda \boldsymbol{v}) \qquad \forall s \in \mathcal{S}. \tag{3}$$



(a) (b) (c) (d)

Figure 1: The generalized d_a-projection problem (4) in $S = 3$ dimensions (a) and two-dimensional projections for the variation distance (b), the χ^2-distance (c) and the KL divergence (d). The gray shaded areas represent the probability simplex Δ_S, the red dashed lines show the boundary of the intersection of the halfspace $\boldsymbol{b}^\top \boldsymbol{p}_{sa} \leq \beta$ with the probability simplex, and the white shapes illustrate contour lines centered at the nominal transition probabilities $\overline{\boldsymbol{p}}_{sa}$.

This yields the optimal value $\boldsymbol{p}^{0\top} \boldsymbol{v}^\star$, where the limit $\boldsymbol{v}^\star = \lim_{t \to \infty} \boldsymbol{v}^t$ is approached component-wise at a geometric rate. The optimal policy $\boldsymbol{\pi}^\star \in \Pi$, finally, is recovered state-wise via

$$\boldsymbol{\pi}^\star_s \in \arg\max_{\boldsymbol{\pi}_s \in \Delta_A} \ \min_{\boldsymbol{p}_s \in \mathcal{P}_s} \sum_{a \in \mathcal{A}} \pi_{sa} \cdot \boldsymbol{p}_{sa}{}^\top (\boldsymbol{r}_{sa} + \lambda \boldsymbol{v}^\star) \qquad \forall s \in \mathcal{S}.$$

4 Robust Bellman Updates via Simplex Projections

In this section, we show that the robust Bellman operator $\mathfrak{J}$ reduces to a generalized projection problem. This reduction is important because it underlies our fast algorithms for computing $\mathfrak{J}$.

At the core of the robust value iteration is the solution of the max-min problem (3). We now show that for ambiguity sets of the form (2), this problem can be solved efficiently whenever the following generalized d_a-projection of the nominal transition probabilities $\overline{\boldsymbol{p}}_{sa}$ can be computed efficiently:

$$\mathfrak{P}(\overline{\boldsymbol{p}}_{sa}; \boldsymbol{b}, \beta) = \left[\begin{array}{ll} \text{minimize} & d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa}) \\ \text{subject to} & \boldsymbol{b}^\top \boldsymbol{p}_{sa} \leq \beta \\ & \boldsymbol{p}_{sa} \in \Delta_S \end{array} \right]. \tag{4}$$

Here, $\boldsymbol{p}_{sa} \in \Delta_S$ are the decision variables and $\overline{\boldsymbol{p}}_{sa} \in \Delta_S$, $\boldsymbol{b} \in \mathbb{R}^S_+$ and $\beta \in \mathbb{R}_+$ are parameters. Note that problem (4) is infeasible if and only if $\min\{\boldsymbol{b}\} > \beta$. Moreover, problem (4) is trivially solved by $\overline{\boldsymbol{p}}_{sa}$ with an optimal objective value of 0 whenever $\boldsymbol{b}^\top \overline{\boldsymbol{p}}_{sa} \leq \beta$. To avoid these trivial cases, we assume throughout the paper that $\min\{\boldsymbol{b}\} \leq \beta$ and $\boldsymbol{b}^\top \overline{\boldsymbol{p}}_{sa} > \beta$. We illustrate the feasible region and optimal solution to problem (4) for different ϕ-divergences in Figure 1.

Our generalized d_a-projection (4) relates to the rich literature on projections onto simplices, which we review in the next section. In fact, our algorithms in the next section solve a variant of the simplex projection problem that is restricted by an additional inequality constraint. We therefore believe that our algorithms may find additional applications outside the RMDP literature.

In the following, we say that for a given estimate $\boldsymbol{v}^t \in \mathbb{R}^S$ of the optimal value function, the robust Bellman iteration (3) is solved to ϵ-accuracy by any $\boldsymbol{v}^{t+1} \in \mathbb{R}^S$ satisfying $\|\boldsymbol{v}^{t+1} - \mathfrak{J}(\boldsymbol{v}^t)\|_\infty \leq \epsilon$. We seek ϵ-optimal solutions because our ambiguity sets are nonlinear and hence the exact Bellman iterate $\mathfrak{J}(\boldsymbol{v}^t)$ may be irrational even if $\boldsymbol{v}^t$ is rational. To simplify the exposition, we define $\overline{R} = [1 - \lambda]^{-1} \cdot \max\{r_{sas'} \, : \, s, s' \in \mathcal{S}, \, a \in \mathcal{A}\}$ as an upper bound on all $[\mathfrak{J}(\boldsymbol{v})]_s$, $\boldsymbol{v} \leq \boldsymbol{v}^\star$ and $s \in \mathcal{S}$.

For divergence-based ambiguity sets, the projection problem (4) is generically nonlinear and can hence not be expected to be solved to exact optimality. To account for this additional complication, we say that for a given $\overline{\boldsymbol{p}}_{sa} \in \Delta_S$, $\boldsymbol{b} \in \mathbb{R}^S_+$ and $\beta \in \mathbb{R}_+$, the generalized d_a-projection $\mathfrak{P}(\overline{\boldsymbol{p}}_{sa}; \boldsymbol{b}, \beta)$ is solved to δ-accuracy by any pair $(\underline{d}, \overline{d}) \in \mathbb{R}^2$ satisfying $\mathfrak{P}(\overline{\boldsymbol{p}}_{sa}; \boldsymbol{b}, \beta) \in [\underline{d}, \overline{d}]$ and $\overline{d} - \underline{d} \leq \delta$.

Theorem 1. *Assume that the generalized d_a-projection* (4) *can be computed to any accuracy $\delta > 0$ in time $\mathcal{O}(h(\delta))$. Then the robust Bellman iteration* (3) *can be computed to any accuracy $\epsilon > 0$ in time $\mathcal{O}(AS \cdot h(\epsilon\kappa / [2A\overline{R} + A\epsilon]) \cdot \log[\overline{R}/\epsilon])$.*

Theorem 1 reduces the evaluation of the robust Bellman iterator $\mathfrak{J}$, which involves the solution of a max-min optimization problem over an s-rectangular ambiguity set that couples all actions $a \in \mathcal{A}$, to a sequence of much simpler and highly structured projection problems that are no longer coupled across different actions $a \in \mathcal{A}$. The next section describes efficient solution schemes for the projection problem (4) in the context of several ϕ-divergence ambiguity sets. The runtimes of these solution schemes are summarized in Table 1. Note that the evaluation of a non-robust Bellman operator requires a runtime of $\mathcal{O}(S^2 \cdot \log A)$, which implies that our algorithms only incur an additional logarithmic overhead to account for robustness in the transition probabilities.

5 Fast Projections on ϕ-Divergence Simplices

We next describe fast algorithms for computing generalized projections onto the probability simplex. Combined with the results from Section 4, these algorithms can be used to efficiently compute the robust Bellman operator. Note that some ϕ-divergences, such as the KL divergence and the χ^2-distance, imply that if $\overline{p}_{sas'} = 0$ for some $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$, then $p_{sas'} = 0$ for all $\boldsymbol{p}_{sa} \in \Delta_S$ with $d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa}) < \infty$, and thus we can remove indices s' with $\overline{p}_{sas'} = 0$. For other ϕ-divergences, such as the Burg entropy and the variation distance, one can readily verify that our results remain valid no matter whether $\overline{\boldsymbol{p}}_{sa} > \boldsymbol{0}$ or not, but the formulations and proofs require additional case distinctions and/or limit arguments. To simplify the exposition, we therefore assume that $\overline{\boldsymbol{p}}_{sa} > \boldsymbol{0}$.

Proposition 1. *For the distance function $d_a(\boldsymbol{p}_{sa}, \overline{\boldsymbol{p}}_{sa}) = \sum_{s' \in \mathcal{S}} \overline{p}_{sas'} \cdot \phi\left(\frac{p_{sas'}}{\overline{p}_{sas'}}\right)$, the optimal value of the projection problem* (4) *equals the optimal value of the bivariate convex problem*

$$\begin{array}{ll} \text{maximize} & -\beta\alpha + \zeta - \sum_{s' \in \mathcal{S}} \overline{p}_{sas'} \phi^\star(-\alpha b_{s'} + \zeta) \\ \text{subject to} & \alpha \in \mathbb{R}_+, \;\; \zeta \in \mathbb{R}. \end{array} \tag{5}$$

Proposition 1 reduces the S-dimensional projection problem (4) to a two-dimensional optimization problem over the dual variables α and ζ. In the following, we show that for the ϕ-divergences from Table 1, problem (5) can be further simplified to univariate convex optimization problems that can be solved efficiently via bisection, binary search or sorting.

5.1 Kullback-Leibler Divergence

We first show that for the KL divergence $\phi(t) = t \log t - t + 1$, the reduced projection problem (5) can be further simplified to a univariate convex optimization problem.

Proposition 2. *For the KL divergence $\phi(t) = t \log t - t + 1$, the optimal value of the projection problem* (4) *equals the optimal value of the univariate convex problem*

$$\begin{array}{ll} \text{maximize} & -\beta\alpha - \log\left(\sum_{s' \in \mathcal{S}} \overline{p}_{sas'} \cdot \mathrm{e}^{-\alpha b_{s'}}\right) \\ \text{subject to} & \alpha \in \mathbb{R}_+. \end{array} \tag{6}$$

We next show that the univariate optimization problem (2) admits an efficient solution via bisection.

Theorem 2. *If $\beta \geq \min\{\boldsymbol{b}\} + \omega$ for some $\omega > 0$, then the projection problem* (4) *can be solved to any δ-accuracy in time $\mathcal{O}(S \cdot \log[\max\{\boldsymbol{b}\} \cdot \log(\min\{\overline{\boldsymbol{p}}\}^{-1})/(\delta\omega)])$.*

Note that the projection problem (4) is infeasible whenever $\beta < \min\{\boldsymbol{b}\}$. The condition in the statement of Theorem 2 can thus be interpreted as a strict feasibility requirement. It is worth contrasting the result of Theorem 2 with the solution of the projection problem (4) as an exponential cone program. The latter would result in a *practical* complexity of $\mathcal{O}(S^3)$, assuming that—which is often observed in practice—the number of iterations of the employed interior-point solver does not grow with the problem dimensions. A *theoretically guaranteed* complexity, on the other hand, does not seem to be available at present as the commercial state-of-the-art solvers for exponential conic programs are not proven to terminate in polynomial time.

Corollary 1. *The robust Bellman iteration* (3) *over a KL divergence ambiguity set can be computed to any accuracy $\epsilon > 0$ in time $\mathcal{O}(S^2 \cdot A \log A \cdot \log[\overline{R}^2 \cdot \log(\min\{\overline{\boldsymbol{p}}\}^{-1})/(\epsilon^2\kappa)] \cdot \log[\overline{R}/\epsilon])$.*

[15] propose a first-order framework for RMDPs over s-rectangular KL divergence ambiguity sets whose robust Bellman update enjoys a complexity of $\mathcal{O}(\ell^2 \cdot S^2 \cdot A \cdot \log(\epsilon^{-1}))$, where ℓ is the iteration

number. A careful analysis results in an overall convergence rate for the optimal MDP policy of $\mathcal{O}(S^3 \cdot A^2 \cdot \epsilon^{-1} \log[\epsilon^{-1}])$. In contrast, the convergence rate of our robust value iteration amounts to $\mathcal{O}(S^2 \cdot A \log A \cdot \log[\overline{R}^2 \cdot \log(\min\{\overline{\boldsymbol{p}}\}^{-1})/(\epsilon^2 \kappa)] \cdot \log[\overline{R}/\epsilon] \cdot \log[\epsilon^{-1}])$. Treating the problem parameters $\overline{R}$, $\overline{\boldsymbol{p}}$ and κ as constants, our convergence rate simplifies to $\mathcal{O}(S^2 \cdot A \log A \cdot \log[\epsilon^{-2}] \cdot \log^2[\epsilon^{-1}])$, which compares favourably against the convergence rate of the first-order scheme. Our numerical results in Section 6 show that this theoretical difference appears to carry over to a favourable empirical performance on test instances as well.

We finally note the related work [1], which optimizes a linear function over the intersection of a probability simplex with a constraint on the KL divergence to a nominal distribution. While one could in principle modify that algorithm to solve our projection problem (4), the resulting algorithm would require an additional bisection and would thus be significantly slower than ours.

5.2 Burg Entropy

Similar to the KL divergence, the reduced projection problem (5) can be further simplified to a univariate convex optimization problem for the Burg entropy $\phi(t) = -\log t + t - 1$.

Proposition 3. *For the Burg entropy $\phi(t) = -\log t + t - 1$, if $\beta > \min\{\boldsymbol{b}\}$, then the optimal value of the projection problem* (4) *equals the optimal value of the univariate convex problem*

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{s' \in \mathcal{S}} \overline{p}_{sas'} \cdot \log\left(1 + \alpha \frac{b_{s'} - \beta}{\beta - \min\{\boldsymbol{b}\}}\right) \\ \text{subject to} & \alpha \leq 1 \\ & \alpha \in \mathbb{R}_+. \end{array} \tag{7}$$

Similar to the KL divergence, the univariate optimization problem (7) can be solved efficiently.

Theorem 3. *If $\beta \geq \min\{\boldsymbol{b}\} + \omega$ for some $\omega > 0$, then the projection problem* (4) *can be solved to any δ-accuracy in time $\mathcal{O}(S \cdot \log[\max\{\boldsymbol{b}\}/(\delta\omega)])$.*

As with the KL divergence, the projection problem (4) corresponding to the Burg entropy can be solved in a practical complexity of $\mathcal{O}(S^3)$ as an exponential cone program, whereas we are not aware of any state-of-the-art solvers equipped with theoretical guarantees. To our best knowledge, RMDPs with s-rectangular Burg entropy ambiguity sets have not been studied previously in the literature.

Corollary 2. *The robust Bellman iteration* (3) *over a Burg entropy ambiguity set can be computed to any accuracy $\epsilon > 0$ in time $\mathcal{O}(S^2 \cdot A \log A \cdot \log[\overline{R}^2/(\epsilon^2\kappa)] \cdot \log[\overline{R}/\epsilon])$.*

Similar to the previous subsection, we note that the related paper [1] optimizes a linear function over the intersection of a probability simplex with a bound on the Burg entropy to a nominal distribution. While that algorithm could in principle be employed to solve our projection problem (4), the resulting solution scheme would not be competitive due to the inclusion of an additional bisection.

5.3 Variation Distance

We first provide an equivalent univariate optimization problem for the reduced projection problem (5) corresponding to the variation distance $\phi(t) = |t - 1|$.

Proposition 4. *For the variation distance $\phi(t) = |t - 1|$, the optimal value of the projection problem* (4) *equals the optimal value of the univariate convex problem*

$$\begin{array}{ll} \text{maximize} & 2 + \alpha(\min\{\boldsymbol{b}\} - \beta) - \displaystyle\sum_{s' \in \mathcal{S}} \overline{p}_{sas'} \cdot [2 + \alpha \cdot (\min\{\boldsymbol{b}\} - b_{s'})]_+ \\ \text{subject to} & \alpha \in \mathbb{R}_+. \end{array} \tag{8}$$

Once more, the univariate optimization problem (8) admits an efficient solution.

Theorem 4. *The projection problem* (4) *can be solved exactly in time $\mathcal{O}(S \log S)$.*

Note that in contrast to the previous results, Theorem 4 employs a binary search and thus offers an *exact* solution to the projection problem (4). Our result of Theorem 4 matches the complexity of the homotopy continuation method proposed by [20]. The correctness and runtime of their algorithm, however, relies on lengthy ad hoc arguments, whereas Theorem 4 relies on the groundwork laid by Theorem 1 and Proposition 1. Problem (4) can also be solved as a linear program with a practical complexity of $\mathcal{O}(S^3)$ and a theoretical complexity of $\mathcal{O}(S^{3.5})$.

Corollary 3. *The robust Bellman iteration* (3) *over a variation distance ambiguity set can be computed to any accuracy* $\epsilon > 0$ *in time* $\mathcal{O}(S^2 \log S \cdot A \cdot \log[\overline{R}/\epsilon])$.

[34] study the related problem of optimizing a linear function over the intersection of a probability simplex with an unweighted 1-norm constraint, and they identify structural properties of the optimal solutions. Since the linear function and the norm constraint are in different places of the optimization problem, however, their findings are not directly applicable to our setting.

5.4 χ^2-Distance

In contrast to the previous subsections, we directly solve the bivariate problem (5) for the χ^2-distance $\phi(t) = (t-1)^2$ without first formulating an associated univariate optimization problem.

Theorem 5. *For the* χ^2*-distance* $\phi(t) = (t-1)^2$*, the optimal value of the projection problem* (4) *can be computed exactly in time* $\mathcal{O}(S \log S)$.

Theorem 5 splits the bivariate piecewise quadratic optimization problem (5) corresponding to the χ^2-distance into $S+1$ bivariate quadratic problems by sorting the components of $\boldsymbol{b}$. Each of these $S+1$ problems can be reduced to the solution of 3 univariate quadratic problems that themselves admit analytical solutions.

Corollary 4. *The robust Bellman iteration* (3) *over a* χ^2*-distance ambiguity set can be computed to any accuracy* $\epsilon > 0$ *in time* $\mathcal{O}(S^2 \log S \cdot A \cdot \log[\overline{R}/\epsilon])$.

The projection problem (4) for the χ^2-distance ambiguity set can be solved as a quadratic program with a practical complexity of $\mathcal{O}(S^3)$ as well as a theoretical complexity of $\mathcal{O}(S^{3.5})$.

The first-order framework of [15] also applies to RMDPs over s-rectangular spherical uncertainty sets. In that case, the robust Bellman update enjoys a complexity of $\mathcal{O}(\ell^2 \cdot S^2 \cdot A \cdot \log^2(\epsilon^{-1}))$, where ℓ is the iteration number. A careful analysis results in an overall convergence rate for the optimal MDP policy of $\mathcal{O}(S^3 \log S \cdot A^2 \cdot \epsilon^{-1} \log[\epsilon^{-1}])$. In contrast, the convergence rate of our robust value iteration amounts to $\mathcal{O}(S^2 \log S \cdot A \cdot \log[\overline{R}/\epsilon] \cdot \log[\epsilon^{-1}])$. Treating the parameter $\overline{R}$ as a constant, our convergence rate simplifies to $\mathcal{O}(S^2 \log S \cdot A \cdot \log^2[\epsilon^{-1}])$, which compares favourably against the convergence rate of [15]. We remark, however, that the spherical ambiguity sets of [15] differ from the χ^2-distance ambiguity sets studied here, and as such the two methods are not directly comparable. We also note that our χ^2-distance ambiguity sets enjoy a strong statistical justification [4, 6].

Computing unweighted 2-norm projections of points onto S-dimensional probability simplices has manifold applications in image processing, finance, optimization and machine learning [1, 8]. [28] proposes one of the earliest algorithms that computes this projection in time $\mathcal{O}(S^2)$ by iteratively reducing the dimension of the problem using Lagrange multipliers. The minimum complexity of $\mathcal{O}(S)$ is achieved, among others, by [26] through a linear-time median-finding algorithm and by [30] through a filtered bucket-clustering method. Note, however, that these algorithms do *not* account for the weights and the additional inequality constraint present in our generalized projection (4). The unweighted 2-norm projection of a point onto the intersection of the S-dimensional probability simplex with an axis-parallel hypercube is computed by [46] through a sorting-based method and by [2] through Newton's method, respectively. [32] optimize a linear function over the intersection of a probability simplex with an unweighted 2-norm constraint through an iterative dimension reduction scheme. [1], finally, study algorithms that optimize linear functions over the intersection of a probability simplex and a bound on the unweighted 2-norm distance to a nominal distribution.

6 Numerical Results

We compare our fast suite of algorithms with the state-of-the-art solver MOSEK 9.3 [3] (commercial) and the first-order method of [15]. To this end, we implemented our algorithms as well as the first-order scheme of [15] in C++, whereas MOSEK is called from Python [36] (BSD license) using the modelling language CVXPY 1.2 [10] (Apache license). We only account for the actual solution time of MOSEK and do not record the time required to formulate the optimization problems in Python.

S	MOSEK	fast	MOSEK/fast
20	1.00	0.01	175.35
100	7.53	0.02	317.80
400	17.87	0.09	190.95
1,000	49.23	0.24	208.20
4,000	235.43	0.94	249.18

$S = A$	MOSEK	fast	MOSEK/fast
20	12.98	1.06	12.21
100	637.78	25.25	25.28
400	24,308.16	343.37	70.79
600	47,473.61	731.17	64.93
700	63,318.00	1,084.65	58.38

Table 2: Comparison of our algorithms ('fast') vs. MOSEK for the projection problem (left) and the Bellman update (right) on KL-divergence constrained ambiguity sets. Runtimes are reported in ms.

$S = A$	f-o (3 its)	f-o (5 its)	fast	f-o/fast (3 its)	f-o/fast (5 its)
20	9.12	25.25	1.06	8.58	23.75
100	183.34	508.83	25.25	7.26	20.15
400	2,821.52	7,833.65	343.37	8.21	22.81
600	6,434.55	17,828.39	731.17	8.80	24.38
700	8,523.80	23,702.00	1,084.65	7.86	21.85

Table 3: Comparison of our algorithms ('fast') vs. the first-order method of [15] (after $\ell = 3, 5$ its.) for the Bellman update on KL-divergence constrained ambiguity sets. Runtimes are reported in ms.

S	MOSEK	fast	MOSEK/fast
20	0.57	0.00	230.64
100	1.41	0.01	202.30
400	4.55	0.02	211.57
600	10.98	0.05	208.12
700	27.33	0.24	114.94

$S = A$	MOSEK	fast	MOSEK/fast
20	4.53	0.08	57.84
100	199.74	2.99	66.80
400	4,415.54	48.35	91.32
600	12,267.68	114.32	107.31
700	18,005.51	148.09	121.59

Table 4: Comparison of our algorithms ('fast') vs. MOSEK for the projection problem (left) and the Bellman update (right) on χ^2-distance constrained ambiguity sets. Runtimes are reported in ms.

All experiments are run on a 3.6 GHz 8-Core Intel Core i9 CPU with 32 GB 2667 MHz DDR4 main memory. Our own algorithm as well as the first-order method of [15] are run in single-threaded mode, whereas MOSEK uses 16 parallel threads since CVXPY does not allow us to restrict computations to a single-threaded mode. All source codes, data sets and detailed results are available on GitHub (URL withheld to maintain anonymity during the review process).

For our experiments, we synthetically generate random RMDP instances as follows. For the projection problem, we sample each component of $\boldsymbol{b}$ uniformly at random between 0 and 1. Similarly, we sample each component of $\overline{\boldsymbol{p}}_{sa}$ uniformly at random between 0 and 1 and subsequently scale $\overline{\boldsymbol{p}}_{sa}$ so that its elements sum up to 1. The parameter β, finally, is uniformly distributed between $\min\{\boldsymbol{b}\} + 10^{-8}$ and $\overline{\boldsymbol{p}}_{sa}^\top \boldsymbol{b} - 10^{-8}$ to adhere to the assumptions of our paper. For the robust Bellman update, all vectors $\boldsymbol{b}_{sa}$ and all transition probabilities $\boldsymbol{p}_{sa}$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, are generated according to the above procedure. The parameter κ is also sampled from a uniform distribution supported on $[0, 1]$.

Tables 2–4 report average computation times over 50 randomly generated test instances. The tables reveal that for both KL divergence and χ^2-distance based ambiguity sets, our algorithms are about 2 orders of magnitude faster than MOSEK in solving the projection problem (4) and about 1-2 orders of magnitude faster than MOSEK in computing the robust Bellman update $\mathfrak{J}$, respectively. Note, however, that MOSEK benefits from a heavy parallelization (it uses 16 threads simultaneously), and a fairer comparison that either restricts MOSEK to a single thread or exploits parallelization in our algorithms (this can readily be achieved in the outer bisection, for example) would further increase the outperformance of our algorithms. The tables also show that our algorithm outperforms the first-order method of [15], and that this outperformance increases rapidly with the iteration number at which the robust Bellman update is performed: While our algorithms outperform the first-order scheme by a factor of about 8 in the third Bellman iteration, this outperformance already increases to a factor of about 20 in the fifth Bellman iteration. Since first-order methods are known to require many iterations for convergence, we conclude that our algorithm compares favorably in this experiment as well.

References

[1] L. Adam and V. Mácha. Projections onto the canonical simplex with additional linear inequalities. *Optimization Methods & Software*, Available online first, 2020.

[2] M. S. Ang, J. Ma, N. Liu, K. Huang, and Y. Wang. Fast projection onto the capped simplex with applications to sparse regression in bioinformatics. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[3] MOSEK ApS. *MOSEK Fusion API for C++ 9.3.20*, 2019. URL https://docs.mosek.com/latest/cxxfusion/index.html.

[4] G. Bayraksan and D. K. Love. Data-driven stochastic programming using phi-divergences. In D. M. Aleman and A. C. Thiele, editors, *INFORMS TutORials in Operations Research*, pages 1–19. 2015.

[5] B. Behzadian, M. Petrik, and C. P. Ho. Fast algorithms for l_∞-constrained s-rectangular robust MDPs. In *Advances in Neural Information Processing Systems*, volume 34, pages (Pre–Proceedings), 2021.

[6] A. Ben-Tal, D. den Hertog, A. de Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2): 341–357, 2013.

[7] Z. Chen, P. Yu, and W. B. Haskell. Distributionally robust optimization for sequential decision-making. *Optimization*, 68(12):2397–2426, 2019.

[8] L. Condat. Fast projection onto the simplex and the l_1 ball. *Mathematical Programming*, 158 (1–2):575–585, 2016.

[9] E. Derman, M. Geist, and S. Mannor. Twice regularized MDPs and the equivalence between robustness and regularization. In *Advances in Neural Information Processing Systems*, volume 35, pages (Pre–Proceedings), 2021.

[10] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[11] M. Diehl and J. Bjornberg. Robust dynamic programming for min-max model predictive control of constrained uncertain systems. *IEEE Transactions on Automatic Control*, 49(12):2253–2257, 2004.

[12] R. Givan, S. Leach, and T. Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122(1):71–109, 2000.

[13] J. Goh, M. Bayati, S. A. Zenios, S. Singh, and D. Moore. Data uncertainty in Markov chains: Application to cost-effectiveness analyses of medical innovations. *Operations Research*, 66(3): 697–715, 2018.

[14] V. Goyal and J. Grand-Clément. Robust Markov decision process: Beyond rectangularity. Available on arXiv, 2018.

[15] J. Grand-Clément and C. Kroer. Scalable first-order methods for robust MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12086–12094, 2021.

[16] J. Grand-Clément and C. Kroer. First-order methods for Wasserstein distributionally robust MDPs. In *Proceedings of Machine Learning Research*, volume 139, pages 2010–2019, 2021.

[17] J. Grand-Clément, C. W. Chan, V. Goyal, and G. Escobar. Robust policies for proactive ICU transfers. Working Paper, 2019.

[18] Vishal Gupta. Near-optimal Bayesian ambiguity sets for distributionally robust optimization. *Management Science*, 65(9):4242–4260, 2019.

[19] C. P. Ho, M. Petrik, and W. Wiesemann. Fast Bellman updates for robust MDPs. In *Proceedings of the 35th International Conference on Machine Learning*, pages 979–1988, 2018.

[20] C. P. Ho, M. Petrik, and W. Wiesemann. Partial policy iteration for l_1-robust Markov decision processes. *Journal of Machine Learning Research*, 22:1–46, 2021.

[21] Q. Huang, Q.-S. Jia, and X. Guan. Robust scheduling of EV charging load with uncertain wind power integration. *IEEE Transactions on Smart Grid*, 9(2):1043–1054, 2018.

[22] G. N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

[23] D. L. Kaufman and A. J. Schaefer. Robust modified policy iteration. *INFORMS Journal on Computing*, 25(3):396–410, 2013.

[24] M. J. Kochenderfer and J. P. Chryssanthacopoulos. Robust airborne collision avoidance through dynamic programming. Project Report ATC-371 for the Federal Aviation Administration, 2011.

[25] Y. Le Tallec. *Robust, Risk-Sensitive, and Data-driven Control of Markov Decision Processes*. PhD thesis, Massachusetts Institute of Technology, 2007.

[26] N. Maculan and G. G. de Paula Jr. A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$. *Operations Research Letters*, 8(4):219–222, 1989.

[27] S. Mannor, O. Mebel, and H. Xu. Robust MDPs with k-rectangular uncertainty. *Mathematics of Operations Research*, 41(4):1484–1509, 2016.

[28] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.

[29] A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[30] G. Perez, M. Barlaud, L. Fillatre, and J.-C. Régin. A filtered bucket-clustering method for projection onto the simplex and the l_1 ball. *Mathematical Programming*, 182(1–2):445–464, 2020.

[31] M. Petrik and D. Subramanian. RAAM: The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 27, pages 1979–1987, 2014.

[32] A. Philpott, V. de Matos, and L. Kapelevich. Distributionally robust SDDP. *Computational Management Science*, 15(3–4):431–454, 2018.

[33] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[34] H. Rahimian, G. Bayraksan, and T. Homem-de-Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1–2):393–420, 2019.

[35] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.

[36] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

[37] P. Rusmevichientong and H. Topaloglu. Robust assortment optimization under the multinomial logit choice model. *Operations Research*, 60(4):865–882, 2012.

[38] A. Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125(2):235–261, 2010.

[39] J. K. Satia and R. E. Lave Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.

[40] A. Shapiro. Rectangular sets of probability measures. *Operations Research*, 64(2):528–541, 2016.

[41] A. Shapiro. Distributionally robust optimal control and MDP modeling. *Operations Research Letters*, 49(3):809–814, 2021.

[42] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[43] A. Tamar, Y. Glassner, and S. Mannor. Policy gradients beyond expectations: Conditional value-at-risk. Available on arXiv, 2014.

[44] A. Tamar, S. Mannor, and H. Xu. Scaling up robust MDPs using function approximation. In *Proceedings of the 31st International Conference of Machine Learning*, 2014.

[45] A. Tirinzoni, X. Chen, M. Petrik, and B. D. Ziebart. Policy-conditioned uncertainty sets for robust Markov decision processes. In *Advances in Neural Information Processing Systems*, volume 31, pages 8953–8963, 2018.

[46] W. Wang and C. Lu. Projection onto the capped simplex. Available on arXiv, 2015.

[47] W. Wiesemann, D. Kuhn, and B. Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[48] H. Xiao, K. Yang, and X. Wang. Robust power control under channel uncertainty for cognitive radios with sensing delays. *IEEE Transactions on Wireless Communications*, 12(2):646–655, 2013.

[49] L. Xin and D. A. Goldberg. Distributionally robust inventory control when demand is a martingale. Available on arXiv, 2018.

[50] H. Xu and S. Mannor. Distributionally robust Markov decision processes. In *Advances in Neural Information Processing Systems*, volume 23, pages 2505–2513, 2010.

[51] H. Xu and S. Mannor. Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

[52] P. Yu and H. Xu. Distributionally robust counterpart in Markov decision processes. *IEEE Transactions on Automatic Control*, 61(9):2538–2543, 2016.

[53] Y. Zhang, L. N. Steimle, and B. T. Denton. Robust Markov decision processes for medical treatment decisions. Available on Optimization Online, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The abstract explains that the main contributions of the paper are *(i)* the development of a decomposition scheme that reduces the computation of a seemingly unstructured robust Bellman operator to the repeated solution of highly structured simplex projection problems and *(ii)* the fast solution of these simplex projection problems for several classes of ϕ-divergences. These claims are backed up in the introduction and the remainder of the paper.

   (b) Did you describe the limitations of your work? [Yes] We took great care to ensure that our numerical results provide an objective and unbiased assessment of our solution approach. In particular, we see that in one of the cases, the outperformance of our approach over MOSEK slightly reduces for larger problem instances.

   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Robust MDPs are well-known in the literature, and our paper develops a new suite of fast algorithms to solve these problems. As such, there are no new negative societal impacts that we can identify.

   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have carefully read those guidelines, and to our best understanding our paper fully complies with them.

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All of our results state the full set of assumptions, with exception of the blanket assumptions that are assumed to hold throughout the paper and that are clearly marked as such.

   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs are contained in the appendix.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] All code, data and instructions for our experimental results are published on GitHub. To maintain anonymity during the review process, we do not provide a link in the current version of the paper.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All details are mentioned either in the numerical results section or in the appendix.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Included in the appendix.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Included in the numerical results section.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use C++, Python and MOSEK, all of which are cited in the text.

   (b) Did you mention the license of the assets? [Yes] All licenses are mentioned.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] All code, data and instructions for our experimental results are published on GitHub. To maintain anonymity during the review process, we do not provide a link in the current version of the paper.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use synthetic data in our experiments.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use synthetic data in our experiments.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We use synthetic data in our experiments.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We use synthetic data in our experiments.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We use synthetic data in our experiments.