On the Complexity of Robust Markov Decision Processes and Bisimulation Metrics

Marnix Suilen
University of Antwerp, Belgium

Guillermo A. Pérez
University of Antwerp, Flanders Make, Belgium

Abstract

Robust Markov decision processes (RMDPs) extend standard Markov decision processes (MDPs) to account for uncertainty in the transition probabilities. RMDPs have an uncertainty set that defines a set of possible transition functions, each of which induces a standard MDP. The natural objective in an RMDP is to optimize the discounted cumulative reward under the worst-case transition function in the uncertainty set. We study the complexity of the associated threshold problem for RMDPs with polytopic uncertainty sets in halfspace representation. Previous results focused on approximating the optimum or restricted attention to specific subclasses of RMDPs, such as interval MDPs or L_∞-RMDPs. Our contributions are threefold: (1) For (s, a)-rectangular RMDPs, we prove that robust policy evaluation is in P via robust linear programming, and that the threshold problem is in NP. As a corollary, robust policy iteration is a polynomial-time algorithm for these RMDPs when the discount factor is fixed. (2) For s-rectangular RMDPs, we show that the threshold problem is in PSPACE via the first-order theory of the reals. (3) We establish lower bounds by reducing both parity games and bisimulation metrics between MDP states to the RMDP threshold problem. A polynomial-time algorithm for the threshold problem would resolve the long-standing open question of whether parity games can be solved in polynomial time. The reduction from bisimulation metrics also yields a practical benefit: it allows us to apply robust policy iteration as a more efficient alternative to the standard fixed-point iteration, as our empirical evaluation demonstrates.

2012 ACM Subject Classification Theory of computation → Markov decision processes

Keywords and phrases Robust Markov decision processes, bisimulation metrics

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2026.48

Related Version *Full Version*: `https://arxiv.org/abs/2604.26748` [38]

Supplementary Material *Software (Source Code)*: `https://github.com/marnixrs/bisimRMDP` [37] archived at `swh:1:dir:af608e324f40fa8a40d40cd75396e1796e6c8dfa`

Funding This work was supported by the Belgian FWO-FNRS "SynthEx" (G0AH524N) project.

1 Introduction

Markov decision processes (MDPs) form the theoretical backbone for many sequential decision-making problems and are foundational in reinforcement learning (RL) [31, 40]. The standard objective in an MDP is for the decision-making agent to compute a policy that maximizes the expected cumulative discounted reward.

MDPs rely on the assumption that their transition probabilities are exactly known; a prohibitive assumption in practice, where probabilities may be inaccurate or derived from a finite amount of samples and thus subject to statistical error [4]. *Robust MDPs* (RMDPs) overcome this assumption by instead modeling an uncertainty set which contains all probabilities the transition function can attain [20, 36, 44]. The typical objective in an RMDP is to optimize the policy and value against the *worst-case* instance in the uncertainty set. RMDPs are particularly well studied from the algorithmic and semantic perspectives, and

have a wide range of applications in, *e.g.*, reinforcement learning [21, 34, 35, 39], statistical model checking [2], and control [3]. Under certain structural independence assumptions on the uncertainty set, known as *rectangularity*, the Bellman operator of standard MDPs can be extended to RMDPs to account for the worst (or best) case instance. Consequently, the classical value and policy iteration algorithms can be extended to RMDPs [20, 28].

For standard MDPs, it is well known that comparing their value against a threshold is P-hard and can be solved in polynomial time through a linear program (LP) [29]. Existing complexity results for RMDPs are less pronounced and follow different directions. In particular, previous works establish (1) polynomial time complexity based on the number of iterations the robust Bellman operator needs to achieve ϵ-approximate solutions, where the precision ϵ and the discount factor are both considered constant [28, 44]; (2) that RMDPs with polytopic uncertainty where the vertices are part of the input (*i.e.*, vertex representation) are in NP $\cap$ coNP by reduction to stochastic games [10]; and (3) RMDPs with an L_∞-based uncertainty set can be solved in (strongly) polynomial time for a fixed discount factor. Note that all these results are consistent with each other, but the complexity of exactly solving RMDPs with polytopic uncertainty in the halfspace representation, the input format typically used in practice [3, 27, 28, 39], has remained open until now.

Contributions. In this paper, we expand on and generalize previous results regarding the computational complexity of RMDPs. We focus on the decision variant of the robust optimization problem: does there exist a policy such that its value surpasses a given threshold under the worst-case instance of the uncertainty set? Our specific contributions are as follows:

1. For (s, a)-rectangular RMDPs, we prove that robust policy evaluation is in P via robust linear programming (Lemma 4.2), and that the threshold problem is in NP (Theorem 4.3). As a corollary, robust policy iteration is a polynomial-time algorithm for these RMDPs when the discount factor is fixed (Theorem 4.6). We also show how three common subclasses of RMDPs, interval MDPs, L_1-RMDPs, and L_∞-RMDPs can be translated into equivalent polytopic RMDPs of polynomial size.
2. For the larger class of s-rectangular RMDPs, where optimal policies may need randomization, we show that the threshold problem is in PSPACE via the first-order theory of the reals (Theorem 4.4).
3. We establish lower bounds by reducing both parity games and bisimulation metrics between MDP states to the threshold problem (Proposition 4.1 and Theorem 5.6). Consequently, a polynomial-time algorithm for the RMDP threshold problem would resolve the long-standing open question of whether parity games can be solved in polynomial time.

We empirically demonstrate the benefits of using robust policy iteration over the standard fixed-point iteration for computing bisimulation metrics.

Related Work

In addition to the results already mentioned, our work is consistent with and expands on the following existing results. In [30], robust linear programs are constructed to compute *optimistic* values and policies for RMDPs with convex uncertainty sets. That is, policies and their value under the *best-case* in the uncertainty set.

For Markov chains with interval uncertainty, [12] shows that computing the value can be done in polynomial time via a linear program. Similarly, bisimulation metrics between Markov chains are also known to be in P [11]. Both results are special cases of ours: that evaluation of a memoryless policy in (s, a)-rectangular RMDPs is in P. For bisimulation

metrics between MDP states, it is known that a single step of the fixed-point iteration is in P [9]. In [14], it is shown that, for a fixed coupling, the fixed-point operator for the bisimulation metric under that coupling yields an MDP value function. Consequently, there exists an MDP whose value function equals the optimal bisimulation metric. We extend their result by showing that the optimal bisimulation metric, characterized by its fixed-point equation, coincides with the robust value function of a specifically constructed RMDP, characterized by its robust Bellman equation. This generalizes their result and allows us to directly apply RMDP algorithms such as robust policy iteration.

RMDPs are closely related to stochastic games [10, 20, 36]. A similar observation has been made for MDPs (*i.e.*, one-and-a-half-player games) and bisimulation metrics between DTMC states. Consequently, MDP algorithms such as policy iteration have been proposed to compute said bisimulation metrics [41]. Our work connecting bisimulation metrics between MDP states with RMDPs can be seen as an extension of that work to two-and-a-half-player games. Finally, in [25], the problem of *minimizing* the bisimulation metric under memoryless randomized policies is shown to be complete for the existential theory of the reals.

2 Background

We write $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$ for the sets of real numbers and non-negative real numbers, respectively. For a finite set X we write $|X|$ for the number of elements in X. A discrete probability distribution over X is a function $\mu\colon X \to [0,1] \subset \mathbb{R}$ with $\sum_{x\in X} \mu(x) = 1$. The set of all distributions over X is denoted by $\mathcal{D}(X)$.

Vectors and matrices are denoted in bold, *e.g.*, $\boldsymbol{x} \in \mathbb{R}^n$, and we write $\boldsymbol{x}(i)$ for the i-th element of $\boldsymbol{x}$. The vector $\boldsymbol{e}_i \in \mathbb{R}^n$ is the element of the standard basis with $\boldsymbol{e}_i(i) = 1$ and zeros elsewhere, and $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ denotes the identity matrix. For a finite set J, we write $\boldsymbol{x} \in \mathbb{R}^J$ for the vector indexed by elements of J, implicitly assuming a total order on J. This notation extends to Cartesian products of indices, *i.e.*, $\boldsymbol{x} \in \mathbb{R}^{J\times K}$. A *convex polytope* is a closed and bounded set of points $\boldsymbol{x} \in \mathbb{R}^n$ defined by a system of linear inequalities $\boldsymbol{D}\boldsymbol{x} \leq \boldsymbol{b}$, with $\boldsymbol{D} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, which we denote by the tuple $(\boldsymbol{D}, \boldsymbol{b})$. For any bounded set X, its *convex hull* is the smallest convex polytope that contains X.

A *linear program* (LP) is an optimization problem of the form

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^\intercal \boldsymbol{x} \\ &\text{subject to} && \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \end{aligned}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{c} \in \mathbb{R}^n$ form the input, and $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of n optimization variables. *Solving the LP* means finding an assignment of $\boldsymbol{x}$ that is feasible, *i.e.*, it satisfies $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, and optimal, *i.e.* $\boldsymbol{c}^\intercal \boldsymbol{x} \leq \boldsymbol{c}^\intercal \boldsymbol{y}$ for all $\boldsymbol{y} \in \mathbb{R}^n$ that are feasible.

For computability matters, we suppose all entries of $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are rational numbers. In this case, it is known that a solution to the LP, if one exists, is also a vector of rationals. LPs can be solved in time $(m+n+L)^{\mathcal{O}(1)}$ [7, 24, 43]. In words, we can solve them in time polynomial in the number n of variables, the number m of inequalities, and the number L of bits used to represent any entry of $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ as a pair of integers written in binary.

2.1 Markov Decision Processes

▶ **Definition 2.1** (MDP). *A Markov decision process (MDP) is a tuple $(S, A, P, R, s_\iota, \gamma)$, where S and A are finite sets of states and actions; $P\colon S\times A \to \mathcal{D}(S)$, the transition function; $R\colon S\times A \to \mathbb{R}$, the reward function; $s_\iota \in S$, the initial state; $\gamma \in (0,1)$, a discount factor.*

A *policy* selects actions to resolve the non-determinism in an MDP. In general, a policy maps paths to distributions over actions: $\pi\colon (SA)^*S \to \mathcal{D}(A)$. The set of all policies is denoted by Π. A *memoryless randomized* policy is a function $\pi\colon S \to \mathcal{D}(A)$ that maps states to distributions over actions; a *memoryless deterministic* policy, a function $\pi\colon S \to A$ that maps states to actions. The set of all memoryless randomized (resp. memoryless deterministic) policies is denoted by Π^{MR} (resp. Π^{MD}).

An MDP and a policy together induce a classical (possibly infinite) Markov chain. The probability space of a Markov chain is well (and even uniquely) defined: its events are sets of infinite paths starting from the initial state s_ι [5, 26]. We can thus talk about the expectation of (Borel-measurable) aggregation functions applied to the transition rewards along runs.

Discounted-reward value. Given an MDP M and a policy $\pi \in \Pi$, its expected discounted cumulative reward value function $V_M^\pi\colon S \to \mathbb{R}$, or simply "value", is defined as

$$V_M^\pi = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right],$$

where $R(s_t, a_t)$ is the reward obtained from the state-action pair visited at step t. We are interested in maximizing this value, *i.e.* we are looking for the optimal value $V_M^* = \sup_{\pi\in\Pi} V_M^\pi$.

It is well established that for this objective in MDPs, the optimum can be achieved by a memoryless deterministic policy [31]. Consequently, the supremum is equal to the maximum over the finite set of all memoryless deterministic policies, *i.e.*

$$V_M^* = \max_{\pi\in\Pi^{\mathrm{MD}}} V_M^\pi.$$

2.1.1 Decision Problem, Encoding, and Known Complexity Results

We reformulate the problem of optimizing the expected reward as that of deciding whether a specific threshold can be surpassed or not.

▶ **Problem 2.2** (MDP discounted reward problem). *Given an MDP M, is there a policy whose value is at least some given threshold $\kappa \in \mathbb{R}$,* i.e., *$\exists \pi \in \Pi^{MD}\colon V_M^\pi(s_\iota) \geq \kappa$?*

It is well known that the discounted reward problem for MDPs reduces, in polynomial time, to solving an LP. To facilitate the encoding, the transition and reward functions of the MDP can be interpreted as matrices. Formally, for each state-action pair $(s, a) \in S \times A$, we define probability vectors $\boldsymbol{p}_{sa} \in [0, 1]^S$ with $\boldsymbol{p}_{sa}(s') = P(s, a)(s')$ for all states $s' \in S$. Similarly, for any memoryless deterministic policy $\pi \in \Pi^{\mathrm{MD}}$, we write $\boldsymbol{P}^\pi \in [0, 1]^{S\times S}$ and $\boldsymbol{R}^\pi \in \mathbb{R}^S$ to denote the transition matrix and reward vector of the induced Markov chain.

▶ **Definition 2.3** (MDP LP encoding [31]). *Let $\boldsymbol{v} \in \mathbb{R}^S$ be a vector of $|S|$ optimization variables, $\boldsymbol{c} \in \mathbb{R}^S$ a vector of $|S|$ strictly positive, but otherwise arbitrary, coefficients (*e.g.*, all ones). For the (unique) optimal solution $\boldsymbol{v}$ to the following LP, we have that $\boldsymbol{v}(\iota) = V_M^*$.*

$$\begin{array}{ll} \textit{minimize} & \boldsymbol{c}^\intercal \boldsymbol{v} \\ \textit{subject to} & \bigwedge_{s\in S} \bigwedge_{a\in A} (-\boldsymbol{e}_s + \gamma \boldsymbol{p}_{sa})^\intercal \boldsymbol{v} \leq -R(s, a) \end{array}$$

This LP is polynomial in the size of M. Moreover, if the probabilities and rewards of M, and the threshold κ are all rational numbers given as pairs of binary-encoded integers, the same holds for the entries of the matrices and vectors of the LP. (Importantly, the magnitude of the maximal constant in the LP is bounded by that of M.) Since solving LPs

is in polynomial time and $V^*_M = \max_{\pi \in \Pi^{\text{MD}}} V^\pi_M$, solving Problem 2.2 is also in polynomial time.

In practice, however, value iteration and policy iteration are more commonly used. Value iteration, on the one hand, computes the least fixed point of the Bellman equation, which results in the optimal value function V^*_M. Several variants of value iteration have been developed over time, either focusing on soundness [17, 32] or efficiency [19]. Policy iteration, on the other hand, computes an optimal policy and its value by iteratively evaluating and improving an arbitrary initial policy. The evaluation of a memoryless deterministic policy π, *i.e.*, computing V^π_M, can be done by solving the following system of linear equations:

$$V^\pi_M = (\boldsymbol{I} - \gamma \boldsymbol{P}^\pi)^{-1} \boldsymbol{R}^\pi. \tag{1}$$

The policy iteration algorithm (optimized with a so-called optimality test) is as follows.

▶ **Definition 2.4** (Policy iteration with suboptimality test). *Let $\pi \colon S \to A$ be an arbitrary policy and $A^*_s = A$ be the set of possibly optimal actions for every state s. Repeat the following:*

1. *Evaluate the current policy π by computing V^π_M by solving Equation 1.*
2. *Compute the normalized values $\Delta^\pi_{sa} = \big(R(s,a) + \gamma \sum_{s' \in S} P(s,a)(s') V^\pi_M(s')\big) - V^\pi_M(s)$.*
3. *Determine the improving actions: $A^+_s = \{a \in A^*_s \mid \Delta^\pi_{sa} > 0\}$. If $A^+_s = \emptyset$ for all $s \in S$, return π and V^π_M. Otherwise, update π such that $\forall s \in S \colon \pi(s) = \operatorname{argmax}_{a \in A^*_s} \Delta^\pi_{sa}$.*
4. *Prune suboptimal actions by computing the new set of possibly optimal actions as follows:*

$$A^*_s = \left\{ a \in A^*_s \,\middle|\, \Delta^\pi_{sa} \geq \max_{a' \in A} \{\Delta^\pi_{sa'}\} - \frac{\gamma}{1-\gamma} \left(\max_{s' \in S} \max_{a' \in A} \{\Delta^\pi_{s'a'}\} - \min_{s' \in S} \max_{a' \in A} \{\Delta^\pi_{s'a'}\} \right) \right\}.$$

Policy iteration without suboptimality tests is sound as long as the policy evaluation step is precise enough to correctly identify improving actions [18]. The variant with suboptimality tests is known to terminate in at most $T(|S|(|A|-1))$ iterations, with $T = \lfloor {}^{|S|}/_{(1-\gamma)} \log({}^{|S|^2}/_{(1-\gamma)}) \rfloor + 1$, and $\mathcal{O}(|S|^2(|S|(|A|-1)))$ operations per iteration to compute all values exactly. Interestingly, it is unknown whether policy iteration runs in polynomial time independent of the discount factor, *i.e.*, when it is part of the input [23].

3 Robust Markov Decision Processes

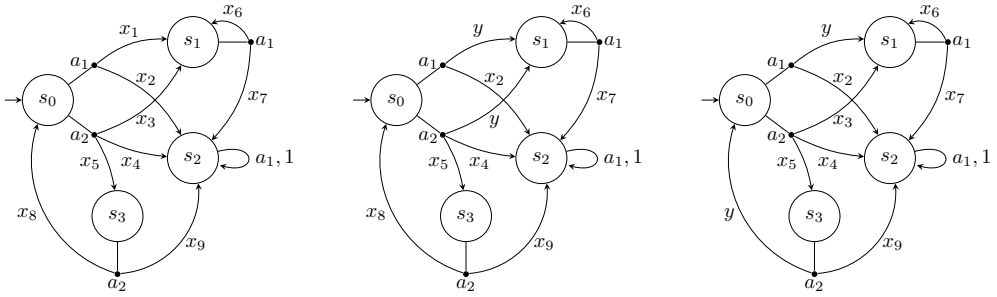
We now introduce robust Markov decision processes (RMDPs). Intuitively, an RMDP is a set of MDPs that differ in their transition probabilities, and policies for RMDPs must be optimized for the best- or worst-case MDP in the set.

▶ **Definition 3.1** (RMDPs). *A* robust MDP *(RMDP) is a tuple $(S, A, \mathcal{U}, R, s_\iota, \gamma)$, where S, A, R, s_ι and γ are as for MDPs, and $\mathcal{U} \subseteq \mathbb{R}^{S \times A \times S}_{\geq 0}$ is a set of non-negative real vectors $\boldsymbol{u} \in \mathcal{U}$ that satisfy $\sum_{s' \in S} \boldsymbol{u}(s,a,s') = 1$, for all $(s,a) \in S \times A$.*

The *uncertainty set* $\mathcal{U}$ of an RMDP induces a family of transition functions $P_{\boldsymbol{u}}(s,a)(s') = \boldsymbol{u}(s,a,s')$, one for each $\boldsymbol{u} \in \mathcal{U}$, such that each tuple $(S, A, P_{\boldsymbol{u}}, R, s_\iota, \gamma)$ is a classical MDP.

3.1 Rectangularity: Local Dependencies in Uncertainty Sets

In this work, we focus on the case where $\mathcal{U}$ is a convex polytope. We also study subcases where $\mathcal{U}$ arises from the Cartesian product of "transition-local" polytopes. This additional structure is known as *rectangularity*, with important algorithmic and complexity implications for solving the RMDP at hand.



(a) An (s,a)-rectangular RMDP. Each transition has a unique variable assigned to it. The constraints on the uncertainty set dictate the range of each variable so that they induce valid probability distributions for each state-action pair.

(b) An s-rectangular RMDP. The same variable y occurs on two different transitions. While there is a dependency between these two transitions, y occurs only on transitions from the same state s_0. This RMDP is thus s-rectangular.

(c) A non-rectangular RMDP. The variable y occurs on multiple transitions that belong to different states, *i.e.*, s_0 and s_3. Hence, the RMDP is neither (s,a)- nor s-rectangular.

Figure 1 An example RMDP under different forms of rectangularity.

In a slight abuse of notation, for two sets $X \subseteq \mathbb{R}^I_{\geq_0}$ and $Y \subseteq \mathbb{R}^J_{\geq 0}$ indexed by sets I and J with $I \cap J = \emptyset$, we assume the vectors $\boldsymbol{v}$ in the cross product $X \times Y$ are "flattened" so that $\boldsymbol{v} \in \mathbb{R}^{I \cup J}_{\geq 0}$. That is, $X \times Y$ consists of real vectors that are indexed by $I \cup J$ instead of pairs of vectors with the first component indexed by I and the other by J.

▶ **Definition 3.2** (Rectangularity). *The uncertainty set $\mathcal{U}$ of an RMDP $(S, A, \mathcal{U}, R, s_\iota, \gamma)$ is:*
(s,a)-rectangular *if it is composed of sets $\mathcal{U}_{(s,a)} \subseteq \mathbb{R}^{\{s\}\times\{a\}\times S}_{\geq 0}$,* i.e. $\mathcal{U} = \bigtimes_{(s,a)\in S\times A} \mathcal{U}_{(s,a)}$*;*
s-rectangular *if it is composed of sets $\mathcal{U}_s \subseteq \mathbb{R}^{\{s\}\times A\times S}_{\geq 0}$ such that $\mathcal{U} = \bigtimes_{s\in S} \mathcal{U}_s$.*

Any RMDP that is neither (s,a)- nor s-rectangular is called *non-rectangular*. Intuitively, (s,a)-rectangularity allows nature to choose a probability distribution at each state-action pair independently from all other state-action pairs. Consequently, nature can minimize the agent's reward by selecting probability distributions *locally*, *i.e.*, at each state-action pair independently and unrestricted from the uncertainty sets at other state-action pairs. The (s,a)-rectangularity assumption is commonly made for both theoretical and algorithmic purposes, as well as in applications such as reinforcement learning [21, 39], abstraction techniques [3], and (statistical) model checking [2]. Under s-rectangularity and, more generally, without rectangularity assumptions, nature's choices are more restricted. While this results in the agent's *robust discounted reward* (a formal definition follows) being less conservative compared to (s,a)-rectangularity, the associated problems become harder. Figure 1 depicts an example RMDP under the three forms of rectangularity.

Known subclasses of (s,a)-rectangular RMDPs, such as interval MDPs (trivially included in convex polytopes), L_∞-RMDPs (which are a special case of IMDPs), and L_1-RMDPs, can all be reduced to RMDPs with convex uncertainty sets, as detailed in the extended version [38].

3.2 Robust Discounted Reward

Given an RMDP $\mathcal{M}$ and a fixed policy $\pi \in \Pi$, its *robust value* is defined as

$$\underline{V}^\pi_{\mathcal{M}} = \inf_{\boldsymbol{u}\in\mathcal{U}} \mathbb{E}_{\pi, P_{\boldsymbol{u}}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right].$$

In words, this is the worst-case value of π over all instantiations of the uncertainty. As before, we are interested in maximizing this value, *i.e.* we are looking for the optimal robust value $\underline{V}^*_{\mathcal{M}} = \sup_{\pi \in \Pi} \underline{V}^{\pi}_{\mathcal{M}}$.

▶ **Proposition 3.3** (Sufficient Policy Classes [44]). *The following policy classes are known to be sufficient for optimality, depending on the rectangularity and uncertainty set assumed.*

(s,a)-rectangular. *Under (s,a)-rectangularity, memoryless deterministic policies suffice,* i.e. $\underline{V}^*_{\mathcal{M}} = \max_{\pi \in \Pi^{MD}} \underline{V}^{\pi}_{\mathcal{M}}$.

s-rectangular and convex. *Under s-rectangularity, if each composing uncertainty set $\mathcal{U}_s$ is convex, memoryless randomized policies suffice,* i.e. $\underline{V}^*_{\mathcal{M}} = \max_{\pi \in \Pi^{MR}} \underline{V}^{\pi}_{\mathcal{M}}$.

To study its complexity, we reformulate the optimization problem as a decision one where we take a threshold as input.

▶ **Problem 3.4** (Robust discounted reward problem). *Given an RMDP $\mathcal{M}$, is there a policy whose robust value is at least some given threshold $\kappa \in \mathbb{R}$,* i.e., $\exists \pi \in \Pi \colon \underline{V}^{\pi}_{\mathcal{M}}(s_\iota) \geq \kappa$?

Note that we are not restricting π in this general version of the problem. We will mostly look at particular instances based on rectangularity and properties of the uncertainty set, where further assumptions can be made about the policy.

3.3 Bellman Operator for (s,a)-Rectangular RMDPs

For this section, we fix an RMDP $\mathcal{M}$. Under (s,a)-rectangularity, as per Proposition 3.3, the optimal robust value of $\mathcal{M}$ is given as

$$\underline{V}^*_{\mathcal{M}} = \max_{\pi \in \Pi^{\mathrm{MD}}} \inf_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\pi, P_{\boldsymbol{u}}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right].$$

The robust value function $\underline{V}^*_{\mathcal{M}} \colon S \to \mathbb{R}$, where we are making the initial state s_ι a parameter, admits a *robust version of the standard Bellman equation.* To state this formally, we recall the robust Bellman operator $\mathfrak{B} \colon (S \to \mathbb{R}) \to (S \to \mathbb{R})$:

$$\mathfrak{B}(\underline{V})(s) = \max_{a \in A} R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \left\{ \sum_{s' \in S} P_{\boldsymbol{u}}(s,a)(s') \underline{V}(s') \right\}.$$

Intuitively, under (s,a)-rectangularity, nature's choice for a probability distribution can be inserted directly into the Bellman equation as a *local optimization problem* at the given state s and action a. The operator behaves just as nicely as that for standard MDPs.

▶ **Proposition 3.5** ([20, Theorem 3.2]). *The operator $\mathfrak{B}$ is a contraction mapping w.r.t. L_∞ and its unique fixed point is $\underline{V}^*_{\mathcal{M}}$.*

4 The Complexity of the Robust Discounted Reward Problem

In this section, we consider the computational complexity of the robust discounted reward problem for the different kinds of rectangularity constraints we introduced earlier. First, we make some observations about how hard the problem is: P-hard and at least as hard as finding the winner in parity games, a problem known to be in $\mathrm{NP} \cap \mathrm{coNP}$ and $\mathrm{UP} \cap \mathrm{coUP}$ [22] and even solvable in quasi-polynomial time [8], but not known to be in P.

▶ **Proposition 4.1.** *The robust discounted reward problem is P-hard, even under rectangularity and convex uncertainty assumptions. Moreover, determining the winner of a parity game reduces, in polynomial time, to the robust discounted reward problem.*

Proof sketch. The problem is P-hard since the discounted reward problem for classical MDPs is P-hard [29] and MDPs are a special case of (rectangular, convex) RMDPs.

For the second part of the claim, consider an arbitrary *discounted-sum game* – in contrast to MDPs, here, the successor state after playing a from s is resolved antagonistically by an adversary. We can construct an (s,a)-rectangular RMDP with uncertainty sets $\mathcal{U}_{(s,a)}$ being the simplex of the deterministic distributions over successors of s after playing a. By memoryless determinacy of discounted-sum games [45], the robust value of the RMDP can be shown to coincide with the "value" of the discounted-sum game. Parity-game hardness is then obtained by appealing to the reduction from parity games to discounted-sum games [22]. The full proof can be found in Appendix A. ◀

Concerning upper bounds, we establish that for (s,a)-rectangular RMDPs, the problem is in NP. The crux of the argument relies on proving that the robust value of a (guessed) memoryless deterministic policy can be evaluated in polynomial time. For s-rectangular RMDPs, where optimal policies may need randomization, we instead resort to an encoding into the theory of the reals, placing it in PSPACE.

To conclude, we leverage our policy evaluation results to derive a robust policy iteration procedure for (s,a)-rectangular convex-uncertainty RMDPs to obtain optimal robust policies.

4.1 The Case of (s,a)-Rectangularity and Convex Polytopic Uncertainty

Let $\mathcal{M}$ be an (s,a)-rectangular RMDP with convex uncertainty sets given as convex polytopes $\{\mathcal{U}_{(s,a)} = (\boldsymbol{D}_{sa}, \boldsymbol{b}_{sa}) \subseteq \mathbb{R}_{\geq 0}^{\{s\}\times\{a\}\times S} \mid s \in S, a \in A\}$. For complexity matters, we suppose the entries of all $\boldsymbol{D}_{sa}$ and $\boldsymbol{b}_{sa}$ are rationals represented as pairs of binary-encoded integers. We also suppose we are given a threshold $\kappa \in \mathbb{Q}$. The last part of the input we consider is a memoryless deterministic policy $\pi\colon S \to A$.

▶ **Lemma 4.2.** *Determining whether $\underline{V}^{\pi}_{\mathcal{M}}(s_\iota) \geq \kappa$ holds true can be done in polynomial time.*

We remark that this result generalizes [12, Proposition 3]. In fact, it can be proven by adapting the separation-oracle argument from [12]. Below, we give an alternative argument using results from *robust linear optimization* [15].

Proof. For all $s \in S$, let $\boldsymbol{u}_s$ be an arbitrary element of the uncertainty set $\mathcal{U}_s = \bigtimes_{a\in A} \mathcal{U}_{(s,a)}$. We introduce some notation to talk about the (uncertain) probabilities and rewards of the DTMC induced by π and the $\boldsymbol{u}_s$. Write $\boldsymbol{p}_s^{\boldsymbol{u}\pi}$ for the vector from $\mathbb{R}^S_{\geq 0}$ satisfying $\boldsymbol{p}_s^{\boldsymbol{u}\pi}(s') = \boldsymbol{u}_s(s,\pi(s),s')$, for all $s' \in S$, and set $R^\pi(s) = R(s,\pi(s))$.

We will be working with the definition of the robust value $\underline{V}^{\pi}_{\mathcal{M}}\colon S \to \mathbb{R}$ where we consider it a function of the initial state. Now, by definition of $\underline{V}^{\pi}_{\mathcal{M}}$ and linearity of the expectation operator, we have the following.

$$\underline{V}^{\pi}_{\mathcal{M}}(s) = \inf_{\boldsymbol{u}_s \in \mathcal{U}_s} R^\pi(s) + \gamma \sum_{s'\in S} \boldsymbol{p}_s^{\boldsymbol{u}\pi}(s') \underline{V}^{\pi}_{\mathcal{M}}(s'), \text{ for all } s \in S.$$

If we write $P^\pi_{\boldsymbol{u}}$ for the matrix such that $P^\pi_{\boldsymbol{u}}(s)(s') = \boldsymbol{p}_s^{\boldsymbol{u}\pi}(s')$ for all $s, s' \in S$, then we can rephrase the above in matrix form as $\underline{V}^{\pi}_{\mathcal{M}} = \inf_{\boldsymbol{u}\in\bigtimes_{s\in S}\mathcal{U}_s} R^\pi + \gamma P^\pi_{\boldsymbol{u}} \underline{V}^{\pi}_{\mathcal{M}}$. By definition of $\underline{V}^{\pi}_{\mathcal{M}}$ and V^π_M for an MDP M, we also have that $\underline{V}^{\pi}_{\mathcal{M}} = \inf_{\boldsymbol{u}\in\bigtimes_{s\in S}\mathcal{U}_s} V^\pi_{M(\boldsymbol{u})}$ where $M(\boldsymbol{u})$ is the MDP

obtained from $\mathcal{M}$ by fixing $\boldsymbol{u}$. Now, from Equation 1 we get that $V^\pi_{M(\boldsymbol{u})} = (\boldsymbol{I} - \gamma P^\pi_{\boldsymbol{u}})^{-1} R^\pi$. In particular, the matrix $\boldsymbol{I} - \gamma P^\pi_{\boldsymbol{u}}$ is guaranteed to be nonsingular, thus invertible, for all $\boldsymbol{u}$.

Since each $\mathcal{U}_s$ is a product of convex polytopes, hence also a convex polytope, we know that the infimum is achieved at some vertex. This means the infimum can be rewritten as a minimum over the (finite) set $\mathrm{vert}(\mathcal{U}_s)$ of vertices of $\mathcal{U}_s$.

$$\underline{V}^\pi_{\mathcal{M}}(s) = \min_{\boldsymbol{u}_s \in \mathrm{vert}(\mathcal{U}_s)} R^\pi(s) + \gamma \sum_{s' \in S} \boldsymbol{p}_s^{\boldsymbol{u}\pi}(s') \underline{V}^\pi_{\mathcal{M}}(s'), \text{ for all } s \in S.$$

The equation above implies the following inequality. Moreover, we know that for all $s \in S$ there is some $\mu(s) \in \mathrm{vert}(\mathcal{U}_s)$ that makes it an equality.

$$\underline{V}^\pi_{\mathcal{M}}(s) \leq R^\pi(s) + \gamma \sum_{s' \in S} \boldsymbol{p}_s^{\boldsymbol{u}\pi}(s') \underline{V}^\pi_{\mathcal{M}}(s'), \text{ for all } s \in S \text{ and all } \boldsymbol{u}_s \in \mathcal{U}_s. \tag{2}$$

Let $\boldsymbol{c} \in \mathbb{R}^S_{>0}$ and $\boldsymbol{v} \in \mathbb{R}^S$ be a vector of (optimization) variables. The unique optimal solution $\boldsymbol{v}$ to the following mathematical program is such that $\boldsymbol{v}(s) = \underline{V}^\pi_{\mathcal{M}}(s)$, for all $s \in S$.

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^\intercal \boldsymbol{v} \\ \text{subject to} & \bigwedge_{s \in S} \bigwedge_{\boldsymbol{u}_s \in (\boldsymbol{D}_{s\pi(s)}, \boldsymbol{b}_{s\pi(s)})} (\boldsymbol{e}_s - \gamma \boldsymbol{u}_s(s, \pi(s)))^\intercal \boldsymbol{v} \leq R^\pi(s) \end{array} \tag{3}$$

Indeed, Equation 2, tells us $\boldsymbol{v}(s) = \underline{V}^\pi_{\mathcal{M}}(s)$ is a feasible assignment. Moreover, any optimal solution must satisfy $(\boldsymbol{I} - \gamma P^\pi_{\boldsymbol{u}})\boldsymbol{v} \leq R^\pi$ for all $\boldsymbol{u} \in \bigtimes_{s \in S} \mathcal{U}_s$. Then, $\boldsymbol{v} \leq (\boldsymbol{I} - \gamma P^\pi_{\boldsymbol{u}})^{-1} R^\pi$ for all $\boldsymbol{u} \in \bigtimes_{s \in S} \mathcal{U}_s$, and in particular $\boldsymbol{v}(s) \leq ((\boldsymbol{I} - \gamma P^\pi_{\mu(s)})^{-1} R^\pi)(s) = \underline{V}^\pi_{\mathcal{M}}(s)$ for all $s \in S$, where the last equality follows from our choice of μ to make Equation 2 an equality. Since $\boldsymbol{c} \in \mathbb{R}^S_{>0}$, we conclude $\underline{V}^\pi_{\mathcal{M}}$ is just as good as any optimal solution $\boldsymbol{v}$.

Equation 3 is a *robust LP* with *polytopic uncertainty*, already in standard form [15]. Hence, we can directly apply dualization techniques to replace the infinite set of constraints by a small number of constraints, yielding the following standard LP [15, Table 1].

$$\begin{array}{ll} \text{minimize} & -\boldsymbol{c}^\intercal \boldsymbol{v} \\ \text{subject to} & \displaystyle\bigwedge_{s \in S} \begin{cases} \boldsymbol{e}_s^\intercal \boldsymbol{v} + (\boldsymbol{b}_{s\pi(s)})^\intercal \boldsymbol{w}_s \leq R^\pi(s) \\ (-\boldsymbol{D}_{s\pi(s)})^\intercal \boldsymbol{w}_s = \gamma \boldsymbol{v} \\ \boldsymbol{w}_s \geq 0 \end{cases} \end{array} \tag{4}$$

This LP is polynomial in the input, particularly in the convex polytopes given in their halfspace representation as tuples $(\boldsymbol{D}_{s\pi(s)}, \boldsymbol{b}_{s\pi(s)})$. Let m be the maximum number of rows of any $\boldsymbol{D}_{s\pi(s)}$ and $\boldsymbol{b}_{s\pi(s)}$, for all s. The LP has at most $(m+1)|S|$ variables, $\boldsymbol{w}_s \in \mathbb{R}^k$ with $k \leq m$ for each $s \in S$ and $\boldsymbol{v} \in \mathbb{R}^S$, and $(2m+1)|S|$ constraints. ◀

Now, we can leverage the fact that deterministic policies suffice. This allows us to have a guess-and-check procedure for the robust discounted reward problem.

▶ **Theorem 4.3.** *The robust discounted reward problem (Problem 3.4) is in NP, assuming (s,a)-rectangularity and convex polytopic uncertainty.*

Proof. By Lemma 4.2 we have that robust policy evaluation of a memoryless deterministic policy is in P. We also know that memoryless deterministic policies are sufficient for discounted reward in (s,a)-rectangular RMDPs by Proposition 3.3. Since each such policy can be represented using polynomial space, we can thus nondeterministically guess a policy and evaluate it in polynomial time. We conclude that the problem is in NP. ◀

4.2 The Case of s-Rectangularity and Convex Polytopic Uncertainty

This time we let $\mathcal{M}$ be an s-rectangular RMDP with convex uncertainty sets given as convex polytopes $\{\mathcal{U}_s = (\boldsymbol{D}_s, \boldsymbol{b}_s) \subseteq \mathbb{R}^{\{s\}\times A\times S} \mid s \in S\}$. Once more we suppose rationals given as pairs of binary-encoded integers, and we take a threshold $\kappa \in \mathbb{Q}$ as input.

▶ **Theorem 4.4.** *The robust discounted reward problem (Problem 3.4) is in PSPACE, assuming s-rectangularity and convex polytopic uncertainty.*

Proof. By Proposition 3.3, it suffices to focus on memoryless randomized policies. Let $\pi\colon S \to \mathcal{D}(A)$ be such a policy. Then, by repeating the first part of the proof of Lemma 4.2 for s-rectangular uncertainty sets, we establish that the unique optimal solution $\boldsymbol{v}$ to the following mathematical program is such that $\boldsymbol{v}(s) = \underline{V}^{\pi}_{\mathcal{M}}(s)$, for all $s \in S$.

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^{\intercal}\boldsymbol{v} \\ \text{subject to} & \bigwedge_{s\in S}\bigwedge_{\boldsymbol{u}_s\in(\boldsymbol{D}_s,\boldsymbol{b}_s)}(\boldsymbol{e}_s^{\intercal} - \gamma\pi(s)^{\intercal}\boldsymbol{u}_s(s))\boldsymbol{v} \leq R^{\pi}(s) \end{array} \tag{5}$$

Equation 5 is a bit different from Equation 3 since π is now memoryless randomized. Note that $\pi(s)$, interpreted as a vector $\mathbb{R}^A$, is right-multiplied by $\boldsymbol{u}_s(s)$ interpreted as a matrix $\mathbb{R}^{A\times S}$. Also, $R^{\pi}(s)$ is now set to $\sum_{a\in A}\pi(s)(a)R(s,a)$.

It follows that the existence of a feasible assignment of $\boldsymbol{v}$ such that $\boldsymbol{v}(s_\iota) \geq \kappa$ suffices to conclude $\underline{V}^{\pi}_{\mathcal{M}}(s_\iota) \geq \kappa$. Recall that π was chosen to be an arbitrary memoryless randomized policy. Towards encoding this in a logical statement, let $\boldsymbol{x} \in \mathbb{R}^{S\times A}$ be a vector of $|S||A|$ real-valued variables. So far we have established that $\underline{V}^{*}_{\mathcal{M}}(s_\iota) \geq \kappa$ if and only if there exists an assignment of $\boldsymbol{v}$ and $\boldsymbol{x}$ that satisfies the following.

$$\boldsymbol{v}(s_\iota) \geq \kappa \wedge \bigwedge_{s\in S}\left(\sum_{a\in A}\boldsymbol{x}(s)(a) = 1 \wedge \bigwedge_{a\in A}\boldsymbol{x}(s)(a) \geq 0\right) \text{ and}$$

$$\forall \boldsymbol{u} : \bigwedge_{s\in S}\left(\boldsymbol{D}_s\boldsymbol{u} \leq \boldsymbol{b}_s \to (\boldsymbol{e}_s^{\intercal} - \gamma\boldsymbol{x}(s)^{\intercal}\boldsymbol{u}(s))\boldsymbol{v} \leq \sum_{a\in A}R(s,a)\boldsymbol{x}(s)(a)\right).$$

This is a sentence in the first-order theory of the reals with order, multiplication, and addition. Given such a sentence of fixed quantifier alternation (in prenex form), determining its truth value is known to be in PSPACE [6, 33]. This concludes the proof of the claim, but we note that we even got a sentence from the $\exists\forall$-fragment of the theory, which is rather low in the recently explored hierarchy of complexity classes induced by the theory of the reals [33]. ◀

4.3 Robust Policy Iteration with Exact Evaluation

Standard policy iteration can be extended to robust policy iteration for (s,a)-rectangular RMDPs with convex polytopic uncertainty by replacing the policy evaluation step from Definition 2.4 with the solution of the linear program from Equation 4.

▶ **Definition 4.5** (Robust policy iteration)**.** *Let $\pi\colon S \to A$ be an arbitrary policy and $A^*_s = A$ be the set of possibly optimal actions for every state s. Repeat the following:*

1. *Evaluate the current policy π by computing its robust value $\underline{V}^{\pi}_{\mathcal{M}}$ by solving Equation 4.*
2. *Compute the normalized values*

$$\underline{\Delta}^{\pi}_{sa} = \left(R(s,a) + \gamma \inf_{\boldsymbol{u}\in\mathcal{U}_{(s,a)}}\sum_{s'\in S}P_{\boldsymbol{u}}(s,a)(s')\underline{V}^{\pi}_{\mathcal{M}}(s')\right) - \underline{V}^{\pi}_{\mathcal{M}}(s),$$

3. *Determine the improving actions:* $A_s^+ = \{a \in A_s^* \mid \underline{\Delta}^\pi_{sa} > 0\}$. *If* $A_s^+ = \emptyset$ *for all* $s \in S$, *return* π *and* $\underline{V}^\pi_{\mathcal{M}}$. *Otherwise, update* π *such that* $\forall s \in S\colon \pi(s) = \operatorname{argmax}_{a \in A_s^*} \underline{\Delta}^\pi_{sa}$.
4. *Prune suboptimal actions by computing the new set of possibly optimal actions as follows:*

$$A_s^* = \left\{ a \in A_s^* \,\middle|\, \underline{\Delta}^\pi_{sa} \geq \max_{a' \in A} \{\underline{\Delta}^\pi_{sa'}\} - \frac{\gamma}{1-\gamma} \left(\max_{s' \in S} \max_{a' \in A} \{\underline{\Delta}^\pi_{s'a'}\} - \min_{s' \in S} \max_{a' \in A} \{\underline{\Delta}^\pi_{s'a'}\} \right) \right\}.$$

▶ **Theorem 4.6.** *Robust policy iteration returns an optimal policy* $\pi \in \Pi^{MD}$ *in at most* $T(|S|(|A|-1))$ *iterations, with* $T = \lfloor {}^{|S|}/_{(1-\gamma)} \log({}^{|S|^2}/_{(1-\gamma)}) \rfloor + 1$, *and* $(m+n+L)^{\mathcal{O}(1)}$ *operations per iteration, where* m *is the number of inequalities,* n *the number of variables, and* L *the number of bits required to represent any of the rationals in the robust policy evaluation LP used in step one.*

Proof. The number of iterations $T(|S|(|A|-1))$ follows directly from the number of iterations of policy iteration for standard MDPs, see Definition 2.4. The complexity of a single iteration is dominated by solving $1 + |S||A|$ polynomially-sized LPs, one global policy evaluation LP in step one, and a local LP for each state-action pair in step two. The number of variables and inequalities of the robust policy evaluation LP in step one is polynomial in the size of the sum of all local LPs, hence the runtime per iteration can be succinctly expressed as $(m+n+L)^{\mathcal{O}(1)}$ steps in the input size of this global policy evaluation LP. ◀

Note that the algorithm runs in polynomial time if γ is fixed, just like standard policy iteration for classical MDPs. Critically, this general formulation of robust policy iteration is *not strongly polynomial*, even with γ being fixed, because we solve an LP in the first steps (that is not known to be doable in strongly polynomial time). If one can replace the LP from steps 1 and 2 by a strongly polynomial policy evaluation, like was recently done for RMDPs with bounded L_∞ uncertainty sets [1], then the whole algorithm becomes strongly polynomial – again, for a fixed γ.

The correctness of the optimality test follows from the robust Bellman operator (still) being a contraction mapping, and thus the proof for standard MDPs can be modified, see Appendix B for the full proof.

5 The Relation with Bisimulation Metrics

The preceding discussion on (robust) policy iteration is the perfect transition from RMDPs to *bisimulation metrics* to compare MDPs. In this section, we will argue that computing the *Kantorovich distance* between two given distributions with respect to a given *pseudometric* roughly corresponds to the (internal) LPs solved in the first steps of the policy iteration procedure described above. Moreover, the fixed-point definition of bisimulation metrics we focus on mirrors the iterative nature of the robust policy iteration procedure. After some necessary preliminaries to introduce all the notation and formalize the two links just claimed, we will construct an RMDP from two given MDPs so that the robust value of the RMDP corresponds to the bisimulation metric between designated states of the MDPs.

5.1 Notation for Bisimulation Metrics

We recap the standard definitions for bisimulation metrics between (states of) MDPs [13].

▶ **Definition 5.1** (Pseudometric). *A pseudometric on a set* X *is a mapping* $d\colon X \times X \to [0, \infty)$ *such that for all* $x, x', x'' \in X$*:* **1.** $x = x' \implies d(x, x') = 0$*;* **2.** $d(x, x') = d(x', x)$*; and* **3.** $d(x, x'') \leq d(x, x') + d(x', x'')$. *The set of all pseudometrics over* X *is denoted as* $\mathfrak{M}_X$.

We compare two pseudometrics using the product ordering. In symbols, for any two pseudometrics $d_1, d_2 \colon X \times X \to [0, \infty)$, we write $d_1 \leq d_2$ if $d_1(x_1, x_2) \leq d_2(x_1, x_2)$ holds true for all $x_1, x_2 \in X$. More interestingly, we use pseudometrics to quantify the distance between distributions over X.

▶ **Definition 5.2** (Kantorovich distance). *Let $d \in \mathfrak{M}_X$ be a pseudometric over a finite set X, with $n = |X|$. The Kantorovich distance between two distributions $\mu, \nu \in \mathcal{D}(X)$, written $\mathfrak{D}_K(d)(\mu, \nu)$, is the solution to the following LP:*

$$\begin{array}{ll} \textit{minimize} & \sum_{j,k=1}^{n} \boldsymbol{\lambda}(j,k) d(x_j, x_k), \\ \textit{subject to} & \left(\bigwedge_{j=1}^{n} \sum_{k=1}^{n} \boldsymbol{\lambda}(j,k) = \mu(x_j)\right) \wedge \left(\bigwedge_{k=1}^{n} \sum_{j=1}^{n} \boldsymbol{\lambda}(j,k) = \nu(x_k)\right) \wedge \boldsymbol{\lambda} \geq \mathbf{0}, \end{array}$$

where $\{\boldsymbol{\lambda}(j,k) \mid 1 \leq j,k \leq n\}$ are n^2 optimization variables. A feasible assignment $\boldsymbol{\lambda} \in \mathbb{R}^{n \times n}$ to this LP is called a coupling*, and by minimizing the objective function, we find the* minimal coupling *between distributions μ and ν w.r.t. the pseudometric d. We denote the set of couplings between μ and ν by $\Lambda_{\mu,\nu}$.*

Note that the set of couplings $\Lambda_{\mu,\nu}$ is a convex polytope and $\sum_{j,k} \boldsymbol{\lambda}(j,k) = 1$, for all $\boldsymbol{\lambda} \in \Lambda_{\mu,\nu}$.

We can now define a (family of) bisimulation metrics between the states of an MDP $M = (S, A, P, R, s_\iota, \gamma)$ using [13, Theorem 4.5].

▶ **Definition 5.3** (Fixed-point bisimulation metrics). *Let $0 < c_R, c_T$ and $c_R + c_T \leq 1$ and consider the operator $\mathfrak{F}_K \colon \mathfrak{M}_S \to \mathfrak{M}_S$ on pseudometrics over the states S.*

$$\mathfrak{F}_K(d)(s_1, s_2) = \max_{a \in A} c_R |R(s_1, a) - R(s_2, a)| + c_T \mathfrak{D}_K(d)(P(s_1, a), P(s_2, a)).$$

Then, $\mathfrak{F}_K$ has a least fixed point d^* which we call the (c_R, c_T)-fixed-point bisimulation metric.

The bisimulation metric between two MDPs M_1 and M_2 is defined as the distance between states $d(s_{\iota_1}, s_{\iota_2})$ in the disjoint union of M_1 and M_2. A standard choice for the weights is $c_T = \gamma, c_R = 1 - \gamma$ [13]. Henceforth, we refer to the $(1-\gamma, \gamma)$-fixed-point bisimulation metric simply as the *bisimulation metric*.

5.2 From Bisimulation Metric to an (s, a)-Rectangular RMDP

We show how computing the bisimulation metric between states of an MDP reduces to computing the robust discounted reward value of an (s, a)-rectangular RMDP.

▶ **Definition 5.4** (Bisimulation metric RMDP). *Let $M = (S, A, P, R, s_\iota, \gamma)$ be an MDP and consider states $s_1^*, s_2^* \in S$. Construct the following (s, a)-rectangular RMDP: $\mathcal{M} = (S \times S, A, \mathcal{U}, \mathcal{R}, (s_1^*, s_2^*), \gamma)$, where $\mathcal{U}$ and $\mathcal{R}$ are the uncertainty set and RMDP reward function defined as follows:*

- *$\mathcal{U} = \bigtimes_{((s,s'),a) \in S \times S \times A} \mathcal{U}_{((s,s'),a)}$ where $\mathcal{U}_{((s,s'),a)} = \Lambda_{P(s,a),P(s',a)}$; and*
- *$\mathcal{R}((s, s'), a) = (1 - \gamma)|R(s, a) - R(s', a)|$.*

That is, $\mathcal{U}$ is an (s, a)-rectangular uncertainty set and the local uncertainty set $\mathcal{U}_{(s,s'),a}$ is the set of couplings between the distributions $P(s, a)$ and $P(s', a)$ in the MDP transition function. The reward function is the difference between rewards in the MDP, scaled by $1 - \gamma$.

The robust value function of an RMDP is, in general, not a pseudometric. For the specific bisimulation metric RMDP constructed above, we first show that its robust value function forms a pseudometric. Then, we prove that the robust value precisely coincides with the bisimulation metric of the original MDP.

▶ **Lemma 5.5.** *The robust value function* $\underline{V}^*_{\mathcal{M}} \colon S \times S \to \mathbb{R}$ *of the RMDP* $\mathcal{M}$ *constructed in Definition 5.4 is a pseudometric over the set of states* S*.*

Proof sketch. By induction, it follows that each property holds for every iteration of the robust Bellman operator of $\mathcal{M}$. The triangle inequality follows from the fact that the Kantorovich distance is a pseudometric. The full proof can be found in Appendix C. ◀

▶ **Theorem 5.6** (Correctness of RMDP construction)**.** *The optimal robust value* $\underline{V}^*_{\mathcal{M}}$ *of the RMDP* $\mathcal{M}$ *constructed in Definition 5.4 precisely coincides with the bisimulation metric* d^* *between states of the MDP* M*,* i.e.*,* $\underline{V}^*_{\mathcal{M}} = d^*$ *and in particular* $\underline{V}^*_{\mathcal{M}}(s^*_1, s^*_2) = d^*(s^*_1, s^*_2)$*.*

Proof. Let M be an MDP for which we want to compute the bisimulation metric between two states (s^*_1, s^*_2), and let $\mathcal{M}$ be the RMDP constructed in Definition 5.4. By Proposition 3.5, the robust Bellman operator $\mathfrak{B}$ has $\underline{V}^*_{\mathcal{M}}$ as unique fixed point. We show that $\mathfrak{B}$, when used on nonnegative functions over states of $\mathcal{M}$, is equivalent to the operator $\mathfrak{F}_K$ on pseudometrics over states of the MDP M. It will thus follow that $\mathfrak{F}_K$ is a contraction mapping, it has a unique fixed point d^*, and that d^* coincides with $\underline{V}^*_{\mathcal{M}}$ as required.

To ease readability, we omit the brackets around pairs of states when used in functions. The robust Bellman operator for $\mathcal{M} = (S \times S, A, \mathcal{U}, \mathcal{R}, (s^*_1, s^*_2), \gamma)$ is defined as follows, where $\underline{V}$ is a function $\underline{V} \colon S \times S \to \mathbb{R}$.

$$\mathfrak{B}(\underline{V})(s, s') = \max_{a \in A} \mathcal{R}(s, s', a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,s',a)}} \left\{ \sum_{(t,t') \in S \times S} P_{\boldsymbol{u}}(s, s', a)(t, t') \underline{V}(t, t') \right\}.$$

We unfold the definition of $\mathcal{R}$ and rewrite the uncertainty set with the set of couplings:

$$= \max_{a \in A} (1 - \gamma) |R(s, a) - R(s', a)| + \gamma \inf_{\boldsymbol{u} \in \Lambda_{P(s,a), P(s',a)}} \left\{ \sum_{(t,t') \in S \times S} P_{\boldsymbol{u}}(s, s', a)(t, t') \underline{V}(t, t') \right\}.$$

Note that by definition of $\mathcal{U}$, we have $P_{\boldsymbol{u}}(s, s', a)(t, t') = \boldsymbol{u}(s, s', a, t, t')$, and thus we obtain $P_{\boldsymbol{u}}(s, s', a)(t, t') = \boldsymbol{\lambda}(t, t')$:

$$= \max_{a \in A} (1 - \gamma) |R(s, a) - R(s', a)| + \gamma \inf_{\boldsymbol{\lambda} \in \Lambda_{P(s,a), P(s',a)}} \left\{ \sum_{(t,t') \in S \times S} \boldsymbol{\lambda}(t, t') \underline{V}(t, t') \right\}.$$

Since the inner optimization problem is a linear program on which the infimum is attained, and minimizers are guaranteed to exist, we may replace inf by min and obtain precisely the definition of the Kantorovich distance $\mathfrak{D}_K$ with respect to $\underline{V}$:

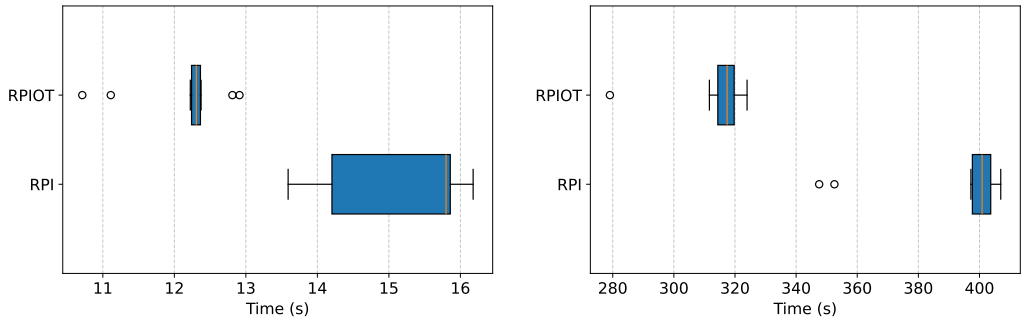
$$\begin{aligned} &= \max_{a \in A} (1 - \gamma) |R(s, a) - R(s', a)| + \gamma \min_{\boldsymbol{\lambda} \in \Lambda_{P(s,a), P(s',a)}} \left\{ \sum_{(t,t') \in S \times S} \boldsymbol{\lambda}(t, t') \underline{V}(t, t') \right\} \\ &= \max_{a \in A} (1 - \gamma) |R(s, a) - R(s', a)| + \gamma \mathfrak{D}_K(\underline{V})(P(s, a), P(s', a)) \\ &= \mathfrak{F}_K(\underline{V})(s, s'). \end{aligned}$$ ◀

5.3 Experimental Evaluation

Our theoretical results enable the use of robust policy iteration as an algorithm to compute the bisimulation metric between MDP states. We empirically evaluate this method and compare it with the standard fixed point approach [13], implemented through robust bounded

Table 1 Comparison of robust bounded value iteration with robust policy iteration with and without optimality test on three Frozen Lake instances.

Map size	Algorithm	Value	Time (s)	±SD	# Iterations	[min,max]
5×5	RBVI	1.346629	332.80		143	
5×5	RPI	1.346629	15.20	±1.09	6.7	[6, 7]
5×5	RPIOT	1.346629	12.14	±0.69	5.8	[5, 6]
10×10	RBVI	1.434019	5298.27		143	
10×10	RPI	1.434019	391.86	±22.30	7.8	[7, 8]
10×10	RPIOT	1.434019	313.83	±12.69	6.9	[6, 7]
20×20	RBVI	`Time Out`	–		–	
20×20	RPI	`Mem Out`	–		–	
20×20	RPIOT	`Mem Out`	–		–	



(a) Frozen Lake 5×5.

(b) Frozen Lake 10×10.

Figure 2 Boxplots of the computation time of RPI and RPIOT across the 10 different seeds.

value iteration (RBVI) for RMDPs [27]. RBVI not only bounds the value from below, as in the standard fixed-point approach, but also from above, and terminates when the difference between these two bounds is less than a prespecified ϵ.

Our implementation is written in Python and uses Gurobi as LP solver [16].[1] The experiments were conducted on an Intel Core Ultra 7 268V with 8 cores of 2.2 GHz base speed and 32 GB RAM. For robust policy iteration, we implement a version with the optimality test enabled (RPIOT) and one without (RPI). We initialize both variants with a random initial policy, and thus repeat the RPI experiments for 10 different seeds to account for this source of randomness, and report mean values ± standard deviation. RBVI precision is set to $\epsilon = 1e^{-6}$. We set a timeout of two hours.

As MDP environments, we use three instances of Gymnasium Frozen Lake [42], with map sizes of 5×5, 10×10, and 20×20, resulting in 25, 100, and 400 state MDPs, respectively. From the RMDP construction, it follows that to compute the bisimulation metric on an MDP with $|S|$ states, the RMDP has $|S|^2$ states; hence, our three Frozen Lake RMDP instances have 625, 10 000, and 160 000 states, respectively. The discount factor is set to $\gamma = 0.9$, and the rewards are 100 for reaching the goal state, -1 for each step, and -10 for getting stuck in a hole.

[1] The full implementation will be made publicly available upon publication.

We compute the bisimulation metric between states 0 (the initial state) and 1, the cell next to the initial state. Intuitively, the bisimulation metric between these two states indicates how much reward we can potentially gain by switching the initial state from 0 to 1. The results are given in Table 1. In Figure 2, we additionally compare the wall-clock computation time of RPI and RPIOT for all seeds. All algorithms achieve the same value to within 6 decimal places. Robust policy iteration (RPI) performs significantly faster than robust bounded value iteration (RBVI), 22 times faster on the small Frozen Lake map, and 13 times faster on the 10×10 map. Integrating the optimality test into RPI further increases its efficiency, as seen in the results for RPIOT. For all seeds, the optimality test caused the algorithm to terminate one iteration earlier, which clearly affected the computation time.

On the 20×20 map, Gurobi runs out of memory during the first policy evaluation step of RPI, while RBVI reaches the timeout of two hours before convergence. These last results signify the key difference between RBVI, which solves many small LPs per iteration, and RPI(OT), which solves one large LP per iteration. Thus, there is a direct trade-off between computational efficiency and memory consumption.

6 Conclusion

We have clarified the complexity landscape of the broad class of RMDPs with polytopic uncertainty sets. In particular, we established that these (s, a)-rectangular RMDPs are P-hard and in NP in general, and that robust policy iteration is a polynomial time algorithm for these RMDPs when the discount factor is fixed. Yet, the exact complexity of (s, a)-rectangular RMDPs remains open. A potential direction for future work would be to prove hardness for some level in the hierarchy of the theory of the reals. For s-rectangular RMDPs, we have established the complexity to be in PSPACE via the theory of the reals. Reductions from parity games and bisimulation metrics relate RMDPs to other well-known problems and allow the use of robust policy iteration to compute bisimulation metrics between MDP states, providing a computationally efficient alternative to the standard fixed-point iteration.

In this paper, we considered only the expected discounted reward as the objective. Our results on the complexity of policy evaluation naturally extend to other objectives, such as reachability, as long as the uncertainty set is *graph-preserving*, *i.e.*, all vectors in the uncertainty set must define the same graph structure, and memoryless policies suffice. Whether our complexity results can be maintained without graph preservation, or whether it makes the RMDP threshold problem harder, remains open for future work.

References

1 Ali Asadi, Krishnendu Chatterjee, Ehsan Goharshady, Mehrdad Karrabi, Alipasha Montaseri, and Carlo Pagano. Strongly polynomial time complexity of policy iteration for L_∞ robust MDPs. *CoRR*, abs/2601.23229, 2026. doi:10.48550/arXiv.2601.23229.

2 Pranav Ashok, Jan Kretínský, and Maximilian Weininger. PAC statistical model checking for Markov decision processes and stochastic games. In *CAV (1)*, volume 11561 of *Lecture Notes in Computer Science*, pages 497–519. Springer, 2019. doi:10.1007/978-3-030-25540-4_29.

3 Thom Badings, Licio Romao, Alessandro Abate, David Parker, Hasan A. Poonawala, Mariëlle Stoelinga, and Nils Jansen. Robust control for dynamical systems with non-Gaussian noise via formal abstractions. *J. Artif. Intell. Res.*, 76:341–391, 2023. doi:10.1613/JAIR.1.14253.

4 Thom Badings, Thiago D. Simão, Marnix Suilen, and Nils Jansen. Decision-making under uncertainty: beyond probabilities. *Int. J. Softw. Tools Technol. Transf.*, 25(3):375–391, 2023. doi:10.1007/S10009-023-00704-3.

5 Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

6 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry*. Springer, 2006.

7 Robert G. Bland, Donald Goldfarb, and Michael J. Todd. The ellipsoid method: A survey. *Oper. Res.*, 29(6):1039–1091, 1981.

8 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasi-polynomial time. *SIAM J. Comput.*, 51(2):17–152, 2022. doi:10.1137/17M1145288.

9 Krishnendu Chatterjee, Luca de Alfaro, Rupak Majumdar, and Vishwanath Raman. Algorithms for game metrics. In *FSTTCS*, LIPIcs, pages 107–118. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008. doi:10.4230/LIPIcs.FSTTCS.2008.1745.

10 Krishnendu Chatterjee, Ehsan Kafshdar Goharshady, Mehrdad Karrabi, Petr Novotný, and Dorde Zikelic. Solving long-run average reward robust MDPs via stochastic games. In *IJCAI*, pages 6707–6715. ijcai.org, 2024. URL: https://www.ijcai.org/proceedings/2024/741.

11 Di Chen, Franck van Breugel, and James Worrell. On the complexity of computing probabilistic bisimilarity. In *FoSSaCS*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2012. doi:10.1007/978-3-642-28729-9_29.

12 Taolue Chen, Tingting Han, and Marta Z. Kwiatkowska. On the complexity of model checking interval-valued discrete time Markov chains. *Inf. Process. Lett.*, 113(7):210–216, 2013. doi:10.1016/J.IPL.2013.01.004.

13 Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *UAI*, pages 162–169. AUAI Press, 2004. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1103&proceeding_id=20.

14 Norman Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *UAI*, pages 210–219. AUAI Press, 2014. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2456&proceeding_id=30.

15 Bram L. Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015. doi:10.1016/j.omega.2014.12.006.

16 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026. URL: https://www.gurobi.com.

17 Serge Haddad and Benjamin Monmege. Interval iteration algorithm for MDPs and IMDPs. *Theor. Comput. Sci.*, 735:111–131, 2018. doi:10.1016/J.TCS.2016.12.003.

18 Arnd Hartmanns, Sebastian Junges, Tim Quatmann, and Maximilian Weininger. The revised practitioner's guide to MDP model checking algorithms. *International Journal on Software Tools for Technology Transfer*, pages 1–31, 2026.

19 Arnd Hartmanns and Benjamin Lucien Kaminski. Optimistic value iteration. In *CAV (2)*, volume 12225 of *Lecture Notes in Computer Science*, pages 488–511. Springer, 2020. doi:10.1007/978-3-030-53291-8_26.

20 Garud N. Iyengar. Robust dynamic programming. *Math. Oper. Res.*, 30(2):257–280, 2005. doi:10.1287/MOOR.1040.0129.

21 Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *J. Mach. Learn. Res.*, 11:1563–1600, 2010. doi:10.5555/1756006.1859902.

22 Marcin Jurdzinski. Deciding the winner in parity games is in UP ∩ co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998. doi:10.1016/S0020-0190(98)00150-1.

23 Lodewijk Kallenberg. Lecture notes on Markov decision processes, 2021. URL: https://pub.math.leidenuniv.nl/~spieksmafm/colleges/LNMB-MDP/Lecture-notes-Kallenberg.pdf.

24 Leonid G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

25 Stefan Kiefer and Qiyi Tang. Minimising the probabilistic bisimilarity distance. In *CONCUR*, LIPIcs, pages 32:1–32:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CONCUR.2024.32.

26 Achim Klenke. *Probability theory: a comprehensive course*. Springer, 2008.

27 Tobias Meggendorfer, Maximilian Weininger, and Patrick Wienhöft. Solving robust Markov decision processes: Generic, reliable, efficient. In *AAAI*, pages 26631–26641. AAAI Press, 2025. doi:10.1609/AAAI.V39I25.34865.

28 Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5):780–798, 2005. doi:10.1287/OPRE.1050.0216.

29 Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Math. Oper. Res.*, 12(3):441–450, 1987. doi:10.1287/MOOR.12.3.441.

30 Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *CAV*, Lecture Notes in Computer Science, pages 527–542. Springer, 2013. doi:10.1007/978-3-642-39799-8_35.

31 Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. doi:10.1002/9780470316887.

32 Tim Quatmann and Joost-Pieter Katoen. Sound value iteration. In *CAV (1)*, volume 10981 of *Lecture Notes in Computer Science*, pages 643–661. Springer, 2018. doi:10.1007/978-3-319-96145-3_37.

33 Marcus Schaefer and Daniel Stefankovic. Beyond the existential theory of the reals. *Theory Comput. Syst.*, 68(2):195–226, 2024. doi:10.1007/S00224-023-10151-X.

34 Rolf A. N. Starre, Marco Loog, Elena Congeduti, and Frans A. Oliehoek. An analysis of model-based reinforcement learning from abstracted observations. *Trans. Mach. Learn. Res.*, 2023, 2023. URL: https://openreview.net/forum?id=YQWOzzSMPp.

35 Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for Markov decision processes. *J. Comput. Syst. Sci.*, 74(8):1309–1331, 2008. doi:10.1016/J.JCSS.2007.08.009.

36 Marnix Suilen, Thom S. Badings, Eline M. Bovy, David Parker, and Nils Jansen. Robust Markov decision processes: A place where AI and formal methods meet. In *Principles of Verif. (3)*, volume 15262 of *Lecture Notes in Computer Science*, pages 126–154. Springer, 2024. doi:10.1007/978-3-031-75778-5_7.

37 Marnix Suilen and Guillermo A. Pérez. On the Complexity of Robust Markov Decision Processes and Bisimulation Metrics (implementation). Software, swhId: swh:1:dir:af608e324f40fa8a40d40cd75396e1796e6c8dfa (visited on 2026-08-12). URL: https://github.com/marnixrs/bisimRMDP, doi:10.4230/artifacts.27643.

38 Marnix Suilen and Guillermo A. Pérez. On the complexity of robust Markov decision processes and bisimulation metrics. *CoRR*, abs/2604.26748, 2026. doi:10.48550/arXiv.2604.26748.

39 Marnix Suilen, Thiago D. Simão, David Parker, and Nils Jansen. Robust anytime learning of Markov decision processes. In *NeurIPS*, 2022.

40 Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018. URL: http://www.incompleteideas.net/book/the-book-2nd.html.

41 Qiyi Tang and Franck van Breugel. Computing probabilistic bisimilarity distances via policy iteration. In *CONCUR*, LIPIcs, pages 22:1–22:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CONCUR.2016.22.

42 Mark Towers, Ariel Kwiatkowski, Jordan K. Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *CoRR*, abs/2407.17032, 2024. doi:10.48550/arXiv.2407.17032.

43 Pravin M. Vaidya. An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations. In Alfred V. Aho, editor, *Proceedings of STOC, 1987, New York, New York, USA*, pages 29–38. ACM, 1987. doi:10.1145/28395.28399.

44 Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov decision processes. *Math. Oper. Res.*, 38(1):153–183, 2013. doi:10.1287/MOOR.1120.0566.

45 Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996. doi:10.1016/0304-3975(95)00188-3.

A Parity-game Hardness

Let π be a memoryless policy. We define the *antagonistic value* of the policy by (adversarially) choosing successors in the RMDP. For a function $\tau\colon S \times A \to S$, we say it respects the possibilities in the uncertainty set $\mathcal{U}$, and write $\tau \models \mathcal{U}$ if:

$$P_{\boldsymbol{u}}(s, a, \tau(s, a)) > 0, \text{ for all } s \in S, a \in \mathsf{Supp}(\pi(s)),$$

for some $\boldsymbol{u} \in \mathcal{U}$. If, additionally,

$$P_{\boldsymbol{u}}(s, a, \tau(s, a)) = 1, \text{ for all } s \in S, a \in \mathsf{Supp}(\pi(s)),$$

for some $\boldsymbol{u} \in \mathcal{U}$, τ is a vertex of $\mathcal{U}$ and we write $\tau \in \mathcal{U}$. Now, define:

$$\mathrm{aVal}^{\pi}_{\mathcal{M}} = \inf_{\tau \models \mathcal{U}} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t),\right],$$

where $s_{t+1} = \tau(s_t, a_t)$ (and thus the only remaining stochasticity, if any, comes from π).

▶ **Lemma A.1.** *Let $\mathcal{U}$ be such that, for all $\tau : S \times A \to S$, it contains the τ-vertex. Then, for all policies π, we have that $\mathrm{aVal}^{\pi}_{\mathcal{M}} \leq \underline{V}^{\pi}_{\mathcal{M}}$. If, moreover, $\mathcal{U}$ is the convex hull of all τ-vertices, then we have equality.*

Proof. The first part of the statement follows directly from containment of the $\boldsymbol{u}$ corresponding to each τ in $\mathcal{U}$. For the second part, we note that the τ-vertices correspond to the vertices of the convex hull. Hence, to conclude, recall that the robust discounted reward value of a given policy is achieved at a vertex of the uncertainty polytope. ◀

A *discounted-sum game* [45] (played on an automaton) is a tuple $(S, A, \Delta, R, s_\iota, \gamma)$ where S, A, R, s_ι and γ are as for MDPs and $\Delta \subseteq S \times A \times S$ is a transition relation. A *strategy* of the protagonist is a function $\sigma\colon S \to A$; while a strategy of the antagonist is a function $\tau : S \times A \to S$ such that $(s, a, \tau(s, a)) \in \Delta$ for all $(s, a) \in S \times A$. We remark that we are restricting ourselves to memoryless deterministic strategies for both players. This is known to be no loss of generality for the value problem (defined below) [45]. A pair (σ, τ) of strategies for both players defines a unique infinite path $\rho_{\sigma\tau} = s_0 a_0 s_1 \dots$ from $s_0 = s_\iota$ and its value $\mathrm{Val}(\rho_{\sigma\tau})$ is $\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$. We are usually interested in the value of the game $\mathrm{Val}^* = \max_\sigma \min_\tau \mathrm{Val}(\rho_{\sigma\tau})$. The natural decision problem asks whether this value is at least some given (rational) threshold.

▶ **Theorem A.2.** *Discounted-sum games reduce in polynomial time to the robust discounted reward problem, and the constructed RMDP is (s, a)-rectangular with convex polytopic uncertainty.*

Proof. For any instance of a discounted-sum game, add as uncertainty sets the convex hull of all τ-vertices for τ a strategy of the adversary. Equivalently, define the uncertainty set so that $\boldsymbol{u}(s, a, s') \geq 0$ if $(s, a, s') \in \Delta$, and is 0 otherwise, for all s, a, s'; and $\sum_{s'} \boldsymbol{u}(s, a, s') = 1$ for all s, a. We observe that this yields an (s, a)-rectangular uncertainty: the successor probabilities depend only on s and a. Moreover, the uncertainty sets are all convex by definition, too. To conclude, it follows from Lemma A.1 that the value of the game coincides with the robust expected discounted-sum value of the RMDP. ◀

Using the known reduction from parity games to discounted-sum games [22], we conclude that if we had a polynomial-time algorithm for the robust expected discounted-sum problem we would have a polynomial-time algorithm for parity games.

B Correctness of the Robust Optimality Test

For standard MDPs, it is known that the set of possibly optimal actions is defined as

$$A_s^* = \left\{ a \in A_s^* \,\middle|\, \Delta_{sa}^\pi \geq \max_{a \in A} \Delta_{sa}^\pi - \frac{\gamma}{1-\gamma} \left(\max_{s' \in S} \max_{a' \in A} \Delta_{s'a'}^\pi - \min_{s' \in S} \max_{a' \in A} \Delta_{s'a'}^\pi \right) \right\}.$$

In the following, we show that the set of possibly optimal actions in an RMDP is given by

$$A_s^* = \left\{ a \in A_s^* \,\middle|\, \underline{\Delta}_{sa}^\pi \geq \max_{a \in A} \underline{\Delta}_{sa}^\pi - \frac{\gamma}{1-\gamma} \left(\max_{s' \in S} \max_{a' \in A} \underline{\Delta}_{s'a'}^\pi - \min_{s' \in S} \max_{a' \in A} \underline{\Delta}_{s'a'}^\pi \right) \right\}.$$

Recall the robust Bellman operator

$$\mathfrak{B}(\underline{V})(s) = \max_{a \in A} R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} P_{\boldsymbol{u}}(s,a)(s') \underline{V}(s').$$

The robust Bellman operator is a monotone contraction mapping [20, Theorem 3.2].

▶ **Theorem B.1.** *Given an (s,a)-rectangular RMDP $\mathcal{M} = (S, A, \mathcal{U}, R, s_\iota, \gamma)$, an action $a \in A$ is suboptimal for state $s \in S$, when for any $\boldsymbol{x} \in \mathbb{R}^S$*

$$R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} P_{\boldsymbol{u}}(s,a)(s') \boldsymbol{x}(s') < \mathfrak{B}(\boldsymbol{x})(s) - \frac{\gamma}{1-\gamma} \mathsf{span}(\mathfrak{B}(\boldsymbol{x}) - \boldsymbol{x}),$$

where $\mathsf{span}(\boldsymbol{y}) = \max_{s \in S} \boldsymbol{y}(s) - \min_{s \in S} \boldsymbol{y}(s)$ *for any* $\boldsymbol{y} \in \mathbb{R}^S$.

Proof. Let $b, c \in \mathbb{R}$ with $b \leq c$, $\boldsymbol{x} \in \mathbb{R}^S$, and $\boldsymbol{v} = \underline{V}$ the robust value function as a vector. If

$$\boldsymbol{x} + b\mathbf{1} \leq \boldsymbol{v} \leq \boldsymbol{x} + c\mathbf{1} \text{ and } R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s'} P_{\boldsymbol{u}}(s,a)(s') \boldsymbol{x}(s') < \mathfrak{B}(\boldsymbol{x})(s) - \gamma(c-b),$$

then action a is suboptimal.

For any $\boldsymbol{u} \in \mathcal{U}_{(s,a)}$, let $\boldsymbol{p} \in \mathbb{R}^{S \times S}$ be the stochastic matrix defined by $\boldsymbol{p}_{\boldsymbol{u}}^a(s,s') = P_{\boldsymbol{u}}(s,a)(s')$. Then for all $\boldsymbol{u} \in \mathcal{U}_{(s,a)}$

$$R(s,a) + \gamma \sum_{s' \in S} \boldsymbol{p}_{\boldsymbol{u}}^a(s,s')(\boldsymbol{x}(s') + c) = R(s,a) + \gamma \sum_{s'} \boldsymbol{p}_{\boldsymbol{u}}^a(s,s') \boldsymbol{x}(s') + \gamma c,$$

as $\boldsymbol{p}_{\boldsymbol{u}}^a$ is stochastic. The above holds in particular for $\inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \boldsymbol{p}_u^a$, since $\mathcal{U}_{(s,a)}$ is a convex polytope and the infimum is attained at one of the vertices. Thus, we obtain

$$\begin{aligned} R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} \boldsymbol{p}_{\boldsymbol{u}}^a(s,s')(\boldsymbol{x}(s') + c) &= R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s'} \boldsymbol{p}_{\boldsymbol{u}}^a(s,s') \boldsymbol{x}(s') + \gamma c \\ &\leq \mathfrak{B}(\boldsymbol{x})(s) + \gamma b = \mathfrak{B}(\boldsymbol{x} + b\mathbf{1})(s). \end{aligned}$$

From the monotonicity of $\mathfrak{B}$, we obtain

$$\begin{aligned} \boldsymbol{v}(s) = \mathfrak{B}(\boldsymbol{v})(s) &\geq \mathfrak{B}(\boldsymbol{x} + b\mathbf{1})(s) \\ &> R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} \boldsymbol{p}_{\boldsymbol{u}}^a(s,s')(\boldsymbol{x}(s') + c) \\ &\geq R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} \boldsymbol{p}_{\boldsymbol{u}}^a(s,s') \boldsymbol{v}(s') \end{aligned}$$

Set $b = \frac{1}{1-\gamma} \min_{s \in S} (\mathfrak{B}(\boldsymbol{x}) - \boldsymbol{x})(s)$ and $c = \frac{1}{1-\gamma} \max_{s \in S} (\mathfrak{B}(\boldsymbol{x}) - \boldsymbol{x})(s)$, and we obtain

$$R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} P_{\boldsymbol{u}}(s,a)(s') \boldsymbol{x}(s') < \mathfrak{B}(\boldsymbol{x})(s) - \frac{\gamma}{1-\gamma} \mathsf{span}(\mathfrak{B}(\boldsymbol{x}) - \boldsymbol{x}),$$

as desired. ◀

From the result above, with $\boldsymbol{x} = \underline{V}^{\pi}$ the value under some policy π, we note that

$$(\mathfrak{B}(\underline{V}^{\pi}) - \underline{V}^{\pi})(s) = \max_{a \in A} R(s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,a)}} \sum_{s' \in S} P_{\boldsymbol{u}}(s,a)(s')\underline{V}^{\pi}(s') - \underline{V}^{\pi}(s) = \max_{a \in A} \underline{\Delta}^{\pi}_{sa}.$$

Thus, we can reduce the set of possibly optimal actions at state $s \in S$ to

$$\left\{ a \in A \,\middle|\, \underline{\Delta}^{\pi}_{sa} \geq \max_{a' \in A} \underline{\Delta}^{\pi}_{sa'} - \frac{\gamma}{1-\gamma}(\max_{s' \in S} \max_{a' \in A} \underline{\Delta}^{\pi}_{s'a'} - \min_{s' \in S} \max_{a' \in A} \underline{\Delta}^{\pi}_{s'a'}) \right\}.$$

C Proof Lemma 5.5

▶ **Lemma 5.5.** The robust value function $\underline{V}$ of the RMDP $\mathcal{M}$ constructed in Definition 5.4 is a pseudometric over the set of states S.

Proof. By construction of the reward function $\mathcal{R}$, all rewards are nonnegative, hence, $\underline{V} \colon S \times S \to \mathbb{R}_{\geq 0}$, and we can initialize the value function as $\underline{V}^{(0)} = \mathbf{0}$. We now show the properties of a pseudometric hold by induction on the iterations of $\underline{V}^{(n)}$. For the base case $\underline{V}^{(0)} = \mathbf{0}$, the properties immediately follow. As induction hypothesis, assume $\underline{V}^{(n)}$ is a pseudometric. We now show that applying the robust Bellman operator on $\underline{V}^{(n)}$, *i.e.*, computing $\underline{V}^{(n+1)} = \mathfrak{B}(\underline{V}^{(n)})$ again satisfies properties 1–3.

1. Assume $s = s'$, we write

$$\underline{V}^{(n+1)}(s,s') = \max_{a} \mathcal{R}(s,s',a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,s',a)}} \sum_{(t,t') \in S \times S} P_{\boldsymbol{u}}(s,s',a)(t,t')\underline{V}^{(n)}(t,t'),$$

   and observe that for all $a \in A$ we have $\mathcal{R}(s,s',a) = (1-\gamma)|R(s,a) - R(s',a)| = 0$. Additionally, since $s = s'$, we have $P(s,a) = P(s',a)$ and since $\mathcal{U}_{(s,s',a)} = \Lambda_{P(s,a),P(s',a)}$ we get that there exists a diagonal coupling of cost zero. Hence, $\underline{V}^{(n+1)}(s,s') = 0$.

2. Note that $\underline{V}^{(n)}(s,s') = \underline{V}^{(n)}(s',s)$, it follows that

$$\begin{aligned}\underline{V}^{(n+1)}(s,s') &= \max_{a \in A} \mathcal{R}(s,s',a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s,s',a)}} \sum_{(t,t') \in S \times S} P_{\boldsymbol{u}}(s,s',a)(t,t')\underline{V}^{(n)}(t,t') \\ &= \max_{a \in A} \mathcal{R}(s',s,a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{(s',s,a)}} \sum_{(t',t) \in S \times S} P_{\boldsymbol{u}}(s',s,a)(t',t)\underline{V}^{(n)}(t',t) = \underline{V}^{(n+1)}(s',s).\end{aligned}$$

3. Using that $\underline{V}^{(n)}$ is a pseudometric by assumption, and that $\mathcal{U}_{(s,s'',a)}$ is the set of couplings between $P(s,a)$ and $P(s'',a)$, we observe that the inner minimization problem defines the Kantorovich distance, which is a pseudometric and thus satisfies the triangle inequality. Thus, it follows that

$$
\begin{aligned}
\underline{V}^{(n+1)}(s,s'') &= \max_{a\in A} \mathcal{R}(s,s'',a) + \gamma \inf_{\boldsymbol{u}\in\mathcal{U}(s,s'',a)} \sum_{(t,t'')\in S\times S} P_{\boldsymbol{u}}(s,s'',a)(t,t'')\underline{V}^{(n)}(t,t'') \\
&\leq \max_{a\in A} \mathcal{R}(s,s',a) + \mathcal{R}(s',s'',a) + \gamma\Bigg(\inf_{\boldsymbol{u}\in\mathcal{U}(s,s',a)} \left\{ \sum_{(t,t')\in S\times S} P_{\boldsymbol{u}}(s,s',a)(t,t')\underline{V}^{(n)}(t,t') \right\} \\
&\quad + \inf_{\boldsymbol{u}\in\mathcal{U}_{(s',s'',a)}} \left\{ \sum_{(t',t'')\in S\times S} P_{\boldsymbol{u}}(s',s'',a)(t',t'')\underline{V}^{(n)}(t',t'') \right\} \Bigg) \\
&\leq \max_{a\in A} \mathcal{R}(s,s',a) + \gamma \inf_{\boldsymbol{u}\in\mathcal{U}(s,s',a)} \sum_{(t,t')\in S\times S} P_{\boldsymbol{u}}(s,s',a)(t,t')\underline{V}^{(n)}(t,t') \\
&\quad + \max_{a\in A} \mathcal{R}(s',s'',a) + \gamma \inf_{\boldsymbol{u}\in\mathcal{U}(s',s'',a)} \sum_{(t',t'')\in S\times S} P_{\boldsymbol{u}}(s',s'',a)(t',t'')\underline{V}^{(n)}(t',t'') \\
&= \underline{V}^{(n+1)}(s,s') + \underline{V}^{(n+1)}(s',s'').
\end{aligned}
$$

◀