Quantitative Analysis of ω-Regular Robust MDPs

Ali Asadi[1], Krishnendu Chatterjee[1], Ehsan Kafshdar Goharshady[1], Mehrdad Karrabi[1], Alipasha Montaseri[1], Ali Shafiee[1]

[1]Institute of Science and Technology Austria
ali.asadi@ist.ac.at, krichnendu.chatterjee@ist.ac.at, ehsan.goharshady@ist.ac.at, mehrdad.karrabi@ist.ac.at, alipasha.montaseri@ist.ac.at, ali.shafiee@ist.ac.at

Abstract

Robust Markov Decision Processes (RMDPs) generalize classical MDPs by allowing uncertainty in transition probabilities and optimizing against their worst-case realization. We consider (s, a)-rectangular RMDPs with *linearly defined* uncertainty sets and study parity objectives, which are a canonical representation of ω-regular objectives. An uncertainty set is linearly defined if it is described by linear inequalities over the transition distribution together with auxiliary variables, which captures the standard L_1 and L_∞ balls as well as general polytopic uncertainty sets. The quantitative value is the supremum, over all agent policies, of the satisfaction probability guaranteed against the adversarial environment. Previous work studied the qualitative analysis, namely the almost-sure (resp. positive) problem that asks whether a single agent policy guarantees satisfaction with probability one (resp. positive probability) against every environment policy. In this work, we solve the exact quantitative problem. Our contributions are threefold. First, we show that both the agent and the environment admit pure memoryless optimal policies. Second, we give a polynomial-time algorithm for quantitative parity on linearly defined robust Markov chains and use it as a subroutine in a policy-iteration algorithm for RMDPs. The algorithm combines quantitative one-step improvements with qualitative almost-sure improvements. Finally, we report experiments comparing our approach with the explicit reduction to stochastic games.

1 Introduction

Robust MDPs. Markov Decision Processes (MDPs) are a fundamental framework for reasoning, decision making, and stochastic planning (Puterman 1994). In a classical MDP, an agent chooses actions and the next state is determined stochastically according to a known transition function. However, transition probabilities are often estimated from data, and the exact transition function is therefore not known. Robust Markov Decision Processes (RMDPs) address this issue by associating every state-action pair with an uncertainty set of possible transition distributions (Iyengar 2005; Nilim and El Ghaoui 2005; Wiesemann, Kuhn, and Rustem 2013). After observing the action of the agent, an adversarial environment chooses a distribution from the corresponding uncertainty set, and the goal of the agent is to optimize against all such choices. This setting is known as (s, a)-rectangular uncertainty. In this work, we consider *linearly defined* uncertainty sets, i.e. sets described by rational linear inequalities over the transition distribution together with auxiliary variables that are existentially quantified. The auxiliary variables keep the description succinct: the standard L_1 and L_∞ balls around a nominal transition distribution are linearly defined with linearly many inequalities, whereas an L_1 ball has exponentially many facets in the transition probabilities alone. Since every uncertainty set lies in the probability simplex, linearly defined sets are precisely the convex polytopes, so this model subsumes general polytopic uncertainty sets while admitting far more succinct descriptions.

Parity and ω-Regular Objectives. Most of the RMDP literature considers quantitative objectives such as discounted sum or long-run average reward. These objectives provide performance guarantees, but they do not express the logical correctness requirements that arise in safety-critical systems, e.g., an autonomous system is required to avoid unsafe configurations, eventually complete a task, or repeatedly provide a service. Such reachability, safety, and liveness requirements are naturally expressed by ω-regular objectives. Parity objectives are a canonical representation of this class. An ω-regular objective can be represented by a finite deterministic parity automaton. Combining this automaton with an RMDP yields a product RMDP with a parity objective (Baier and Katoen 2008). Hence, RMDPs with parity objectives provide a flexible and broad framework for the analysis of probabilistic systems with logical objectives.

Quantitative Analysis. Given an RMDP and a logical objective, the value of a state is the maximal satisfaction probability guaranteed by the agent against every environment choice. Qualitative analysis considers two questions. The almost-sure (resp. positive) problem asks whether there exists an agent policy that guarantees satisfaction with probability one (resp. greater than zero) against every environment choice. However, these qualitative analyses do not distinguish among policies when no almost-sure guarantee exists. Hence, quantitative analysis is important for finding optimal policies and computing the values. The work of Asadi et al. (2026b) gives algorithms for the positive and almost-sure problems: polynomial time for reachability and safety objectives, and quasi-polynomial time and polynomial space for all parity objectives. However, the quantitative analysis remains open.

Contributions. We solve the quantitative parity problem for RMDPs with linearly defined uncertainty sets. Our contributions are threefold.

1. We first consider robust Markov chains (RMCs), i.e., RMDPs in which the agent has a single action. For reachability objectives, we formulate a polynomial-size linear program. We then combine this formulation with an analysis of recurrent components to solve parity objectives, which yields a polynomial-time algorithm for the quantitative parity problem.
2. We then present a policy-iteration algorithm for RMDPs. We establish that both the agent and the environment admit stationary memoryless optimal policies. Each iteration evaluates the RMC induced by the current agent policy and applies either a quantitative one-step improvement or a qualitative almost-sure improvement. We show that the algorithm terminates with the exact value vector and optimal agent and environment policies.
3. Finally, we implement our algorithms for reachability and parity objectives under L_1 and L_∞ uncertainty. We compare our approach with the explicit reduction to a turn-based stochastic game. Our approach scales substantially better on the Garnet and Inventory benchmarks, whose branching factor grows with the instance size, whereas the explicit reduction is faster on the Frozen Lake benchmark, which has a small branching factor.

Related Work. Our policy-iteration algorithm is, in spirit, close to the strategy-improvement method for stochastic parity and Rabin games (Chatterjee and Henzinger 2006a,b), but in our robust setting the environment ranges over a continuum of distributions in an uncertainty polytope rather than over the finitely many vertices of a stochastic game. Our improvement therefore works directly with the linearly defined uncertainty sets (through the value classes, tight actions, and tight faces of Section 4) and never constructs the exponentially larger game. The closest prior work on logical objectives for RMDPs is Asadi et al. (2026b), which solves the positive and almost-sure problems using uncertainty-set oracles. Our qualitative improvement step invokes their almost-sure parity procedure as a sub-routine, whereas the exact quantitative problem we solve here remained open. An extended discussion of related work is deferred to Appendix A.

2 Preliminaries and Model Description

Notation. For any finite set A we use $\Delta(A)$ and 2^A to denote the set of all distributions over A and the power-set of A respectively. We use $[n]$ as shorthand for $\{0, \dots, n\}$.

MDPs and MCs. *A Markov decision process (MDP)* is a tuple $M = (\mathcal{S}, \mathcal{A}, \delta)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions and $\delta \colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ specifies the transition probabilities. A *Markov chain (MC)* is an MDP whose action set is a singleton, i.e. $|\mathcal{A}| = 1$.

The semantics of MDPs are defined by *policies*. An *agent policy* is $\sigma \colon (\mathcal{S} \times \mathcal{A})^* \times \mathcal{S} \to \Delta(\mathcal{A})$ that maps a history of states and actions to a distribution over actions. Given such a policy a token is placed on an initial state $s_0 \in \mathcal{S}$ and at the i-th step, the agent draws an action $a_i \sim \sigma(s_0, a_0, \dots, s_{i-1}, a_{i-1}, s_i)$ and the next state s_{i+1} is sampled from $\delta(s_i, a_i)$. Fixing an initial state s_0, the cylinder construction of (Baier and Katoen 2008) generates a probability distribution $\Pr^\sigma_{s_0}$ over infinite paths in the MDP. The semantics of MCs are defined analogously, but since there is no choice of actions, the next state is sampled directly from the transition probabilities.

RMDPs and RMCs. Robust Markov Decision Processes (RMDPs) extend MDPs by allowing uncertainty in the transition probabiliaties. Formally, an RMDP is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ where $\mathcal{S}$ and $\mathcal{A}$ are as before, but $\mathcal{P} \colon \mathcal{S} \times \mathcal{A} \to 2^{\Delta(\mathcal{S})}$ maps each state-action pair to a set of possible transition distributions, called the uncertainty set. A Robust Markov Chain (RMC) is an RMDP where $|\mathcal{A}| = 1$, hence an RMC $\mathcal{R}$ is defined by a pair $(\mathcal{S}, \mathcal{P})$.

The semantics of RMDPs are defined by a pair of policies: an agent policy σ as before, and an environment policy $\tau \colon (\mathcal{S} \times \mathcal{A})^* \times (\mathcal{S} \times \mathcal{A}) \to \Delta(\mathcal{S})$ that maps a history of states and actions to a distribution over next states where for every history $h \in (\mathcal{S} \times \mathcal{A})^*$ and $(s_i, a_i) \in \mathcal{S} \times \mathcal{A}$, it holds that $\tau(h, s_i, a_i) \in \mathcal{P}(s_i, a_i)$. As with MDPs, fixing an initial state s_0 and a policy profile (σ, τ) in an RMDP $\mathcal{M}$ induces a probability distribution $\Pr^{\sigma,\tau}$ over infinite paths in $\mathcal{M}$. The semantics of RMCs are defined analogously.

Policies. An agent policy σ is *deterministic* if it maps every history to a single action and *memoryless* if it only depends on the current state, i.e. $\sigma(h, s) = \sigma(h', s)$ for all histories h, h' and state $s \in \mathcal{S}$. A *positional* policy is both deterministic and memoryless. Conventionally, we assume positional policies are $\mathcal{S} \to \mathcal{A}$ functions. We denote the set of all agent policies by Σ and the set of all environment policies by Γ. Fixing a positional policy σ in an RMDP $\mathcal{M}$, induces an RMC $\mathcal{M}^\sigma$ where the only available action at each state s is $\sigma(s)$ and the uncertainty set is defined as $\mathcal{P}^\sigma(s) = \mathcal{P}(s, \sigma(s))$.

Remark 1. *This work is concerned with (s, a)-rectangular RMDPs whose semantics are defined above: the environment observes the action taken by the agent and then chooses a distribution from the corresponding uncertainty set.*

Objectives and Values. An objective φ is a measurable set of infinite paths in $\mathcal{M}$. Given an agent policy σ and an environment policy τ, we denote by $\Pr^{\sigma,\tau}_{\mathcal{M},s}[\varphi]$ the probability that a path starting from state s satisfies the objective φ. The *value* of an objective φ in an RMDP $\mathcal{M}$ from s is

$$\boldsymbol{v}^{\mathcal{M}}_s(\varphi) = \sup_{\sigma \in \Sigma} \inf_{\tau \in \Gamma} \Pr^{\sigma,\tau}_s[\varphi].$$

Values are defined analogously for RMCs.

We focus on *parity* objectives. A parity objective $\mathsf{Parity}(C)$ is defined with respect to a coloring function $C \colon \mathcal{S} \to [d]$. Particularly, $\mathsf{Parity}(C)$ is defined as the set of all infinite paths π such that the maximum color that appears infinitely often in π is even, i.e.

$$\mathsf{Parity}(C) = \{\pi \in \mathcal{S}^\omega \mid \max\{C(s) \mid s \in \mathsf{Inf}(\pi)\} \text{ is even}\},$$

where $\mathsf{Inf}(\pi)$ denotes the set of states that occur infinitely often in π. It is a classical result that every ω-regular objective, including reachability, safety and liveness, can be reduced to a parity objective (Baier and Katoen 2008).

Model Assumption. A *linearly defined* uncertainty set $\mathcal{P}(s,a)$ is the projection of a polyhedron onto the probability coordinates, i.e. there exist a matrix A and a vector b, of dimensions compatible with k auxiliary variables, where:

$$\mathcal{P}(s,a) = \Big\{ x \in \Delta(\mathcal{S}) \;\Big|\; \exists y \in \mathbb{R}^k \colon A \begin{bmatrix} x \\ y \end{bmatrix} \leq b \Big\}.$$

Note that the auxiliary variables y are existentially quantified, so a linearly defined set need not be cut out by linear inequalities over x alone. This is strictly more expressive in terms of succinctness: an L_1 ball around a nominal distribution is linearly defined using only $O(|\mathcal{S}|)$ inequalities and auxiliary variables, whereas over x alone it has exponentially many facets (see Appendix B). Finally, since $\mathcal{P}(s,a) \subseteq \Delta(\mathcal{S})$ is bounded, every linearly defined uncertainty set is a convex polytope. Throughout this paper, we assume that every uncertainty set is linearly defined. This assumption is satisfied by the standard L_1 and L_∞ balls (See Appendix B), as well as by the more general polytopic uncertainty sets.

Problem Statement. The main problem statement in this paper is summarized as follows:

> Given a linearly defined RMDP $\mathcal{M}$ and a parity objective $\mathsf{Parity}(C)$, the *quantitative parity problem* asks to compute the value $\boldsymbol{v}_s^{\mathcal{M}}(\mathsf{Parity}(C))$ for every state $s \in \mathcal{S}$.

In Section 3, we present an algorithm for solving the quantitative parity problem in RMCs that runs in polynomial time. We will than use the RMC algorithm as a sub-procedure to solve the quantitative parity problem in RMDPs in Section 4.

3 Quantitative RMC Analysis Sub-Procedure

We present a polynomial-time algorithm for solving the quantitative safety and parity objectives in linearly defined RMCs $\mathcal{R} = (\mathcal{S}, \mathcal{P})$ where the environment's goal is to minimize the probability that the objective is satisfied. This is then used in the algorithm for solving RMDPs as a sub-procedure. In what follows, we first consider the case of safety objectives (Section 3.1) and present an LP-based algorithm for solving them. We then use the method for solving safety objectives and present a solution for the quantitative parity problem (Section 3.2).

3.1 Safety

Given a set $\mathcal{T} \subseteq \mathcal{S}$, the *safety* objective $\mathsf{Safe}(\mathcal{T})$ is the set of infinite paths that never leave $\mathcal{T}$. The environment is adversarial and *minimizes* the probability of remaining safe, so the value at a state s is $\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T})) = \inf_\tau \Pr_s^\tau[\mathsf{Safe}(\mathcal{T})]$. We compute this value for every s with a single linear program, exploiting the duality of safety and reachability together with the flow-like characterization of reachability probabilities.

Pre-Processing. We first dispose of the states whose safety value is 0 or 1. We use the polynomial-time method of (Asadi et al. 2026b) to compute the sets $\mathcal{S}^{=0}$ and $\mathcal{S}^{=1}$ of states with safety value 0 and 1, respectively. Our pre-processing step makes all states in $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ absorbing. This leaves the states $\mathcal{S}^? = \mathcal{S} \setminus (\mathcal{S}^{=0} \cup \mathcal{S}^{=1})$, whose values are to be computed.

Flow-like Characterization. Fix a source state $s_0 \in \mathcal{S}^?$ and picture injecting one unit of probability mass there. As the play unfolds, this mass is routed through the state space along the transition probabilities until it is absorbed in $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$. The adversarial environment moves this mass so as to *minimize* the amount absorbed in $\mathcal{S}^{=1}$ and *maximize* the amount absorbed in $\mathcal{S}^{=0}$. Recording, for each state s, the expected number of visits x_s, and for each edge (s,s') the expected number of traversals $f_{s,s'}$, these quantities behave exactly like a conserved flow. They satisfy the following linear properties, which we later assemble into the LP:

- **Non-negativity.** No state is visited and no edge is traversed a negative number of times:

$$\mathsf{NNeg} \;\equiv \bigwedge_{s \in \mathcal{S}^?,\, s' \in \mathcal{S}} \big(x_s \geq 0 \;\wedge\; f_{s,s'} \geq 0 \big).$$

- **Conservation.** The expected number of visits to a state equals the mass injected there plus the total flow entering it, which is Kirchhoff's conservation law:

$$\mathsf{Cons} \;\equiv \bigwedge_{s \in \mathcal{S}^?} \Big(x_s \;=\; \mathbf{1}[\, s = s_0 \,] \;+\; \sum_{s' \in \mathcal{S}^?} f_{s',s} \Big).$$

- **Splitting.** Each visit to a state is followed by exactly one outgoing edge, so the visits to a state are split among its outgoing flow:

$$\mathsf{Split} \;\equiv \bigwedge_{s \in \mathcal{S}^?} \Big(\sum_{s' \in \mathcal{S}} f_{s,s'} \;=\; x_s \Big).$$

- **Admissibility.** The proportions in which the flow leaves a state form a distribution $\boldsymbol{f}_s / x_s$, where $\boldsymbol{f}_s = (f_{s,s'})_{s' \in \mathcal{S}}$, that the environment is allowed to pick, i.e. $\boldsymbol{f}_s / x_s \in \mathcal{P}(s)$. This membership is, at face value, nonlinear: it divides the flow variables by the visit variable. However, by the Assumption $\mathcal{P}(s) = \{p : A_s\, p \leq b_s\}$, so the constraint reads $A_s(\boldsymbol{f}_s / x_s) \leq b_s$; multiplying both sides by $x_s \geq 0$ preserves the inequality and clears the denominator, yielding the equivalent linear constraint

$$\mathsf{Adm} \;\equiv \bigwedge_{s \in \mathcal{S}^?} \big(A_s\, \boldsymbol{f}_s \;\leq\; x_s\, b_s \big).$$

  This is jointly linear in the flow variables $\boldsymbol{f}_s$ and the visit variable x_s, since A_s and b_s are constants.

Since $\mathcal{S}^{=1}$ is absorbing, the probability of reaching $\mathcal{S}^{=1}$ from s_0 is simply the total flow that crosses into $\mathcal{S}^{=1}$. Hence, the environment minimises $\sum_{\substack{s \in \mathcal{S}^? \\ w \in \mathcal{S}^{=1}}} f_{s,w}$, which is therefore the objective of the LP. The following lemma (proved in Appendix C) formalises the above discussion and establishes the correctness of the LP:

Lemma 2. *Given a linearly defined RMC $\mathcal{R}$, initial state $s_0 \in \mathcal{S}^?$ and with safe set $\mathcal{T} \subseteq \mathcal{S}$, let $\boldsymbol{v}^*$ be the solution to the following linear program:*

$$\min_{x,\, f} \quad \sum_{s \in \mathcal{S}^?,\, w \in \mathcal{S}^{=1}} f_{s,w} \qquad (\star)$$
$$\textit{s.t.} \quad \mathsf{NNeg} \wedge \mathsf{Cons} \wedge \mathsf{Split} \wedge \mathsf{Adm},$$

then $\boldsymbol{v}_{s_0}^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T})) = \boldsymbol{v}^*$.

Building on this, we obtain that the safety values of an RMC, together with optimal environment policies, can be computed in polynomial time.

Theorem 3. *Given a linearly defined RMC $\mathcal{R}$ with safety objective $\mathsf{Safe}(\mathcal{T})$, for each fixed s, the value $\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T}))$, together with a positional environment policy that attains it, can be computed in time polynomial in the size of $\mathcal{R}$.*

3.2 Parity

In this section, we solve the quantitative parity problem by reducing it to a safety computation. Recall that the environment is adversarial and *minimizes* the probability of satisfying $\mathsf{Parity}(C)$. We first show a polynomial-time algorithm that computes the set W_{odd} of states lying in an end-component whose dominant color is odd implying that the environment can violate $\mathsf{Parity}(C)$ with probability 1. We then show that the parity value at any state equals the value of $\mathsf{Safe}(\mathcal{S} \setminus W_{\mathrm{odd}})$, i.e., the environment's goal is to minimize the probability of not reaching W_{odd}. This value is then computed by the safety variant of the LP of Section 3.1.

Almost-Sure Violation of Parity. We compute the set W_{odd} of states lying in an end-component from which the environment can violate $\mathsf{Parity}(C)$ with probability 1. We note that the algorithm of (Asadi et al. 2026b) can be used to compute this qualitative information too, however, their method requires super-polynomial time in the worst case, while our goal is to provide a polynomial-time algorithm.

We first define the notion of *maximal end-components* (MECs) in RMCs, which are the analogue of MECs in MDPs:

Definition 4. *For a set $U \subseteq \mathcal{S}$ and a state $s \in U$, let $\mathcal{P}_U(s) = \{\, p \in \mathcal{P}(s) \;:\; \mathsf{supp}(p) \subseteq U \,\}$ be the set of distributions at s with which the environment can keep the play inside U. A set $U \subseteq \mathcal{S}$ is an* end-component *(EC) of an RMC $\mathcal{R}$ if the following conditions hold:*
1. For every state $s \in U$, we have $\mathcal{P}_U(s) \neq \emptyset$.
2. The graph induced by $\mathcal{P}_U$ on U is strongly connected.
Furthermore, U is a maximal EC (MEC) if there is no end-component $U' \subseteq \mathcal{S}$ such that $U \subset U'$.

Intuitively, whenever the play enters a (maximal) EC, the environment can ensure that the play remains in the end-component forever. The environment's goal would then be to reach an end-component in which the parity objective can be violated almost-surely.

The MecDecomp procedure in Algorithm 4 of Appendix D takes an RMC $\mathcal{R}$ as input and returns its MECs in polynomial time. The algorithm is based on the standard algorithm for computing MECs in MDPs (Chatterjee and Henzinger 2014), and utilizes the polytopic structure of the uncertainty sets.

Algorithm 1 illustrates the OddEC procedure which utilizes MecDecomp to compute the set W_{odd} as the union of those ECs where the environment can almost-surely violate $\mathsf{Parity}(C)$ in them. Since the play can be confined to a MEC and visits all of its states infinitely often, the dominant color the environment can enforce in a MEC is its highest color m_i: if m_i is odd, the environment stays in the MEC forever and makes the dominant color odd, so the whole MEC violates the objective and is added to W_{odd}; otherwise the environment must avoid the even color m_i becoming dominant, so we discard the states of color m_i and recurse on the rest.

Algorithm 1: $\mathsf{OddEC}(B)$

Input: A subset $B \subseteq \mathcal{S}$; the top-level call uses $B = \mathcal{S}$.
Output: Union of ECs in B whose dominant color is odd.

$M_1, \ldots, M_k \leftarrow \mathsf{MecDecomp}(B)$
$W \leftarrow \emptyset$
for $i = 1, \ldots, k$ **do**
  $m_i \leftarrow \max_{s \in M_i} C(s)$
  if m_i is odd **then** $W \leftarrow W \cup M_i$
  else $W \leftarrow W \cup \mathsf{OddEC}(M_i \setminus C^{-1}(m_i))$
end for
return W

The following lemma (proved in Appendix E) establishes the correctness and complexity of the OddEC procedure.

Lemma 5. OddEC *computes W_{odd} in polynomial time.*

Quantitative Satisfaction of Parity. Having computed W_{odd}, we reduce the quantitative parity problem to the safety computation of Section 3.1. From any state in W_{odd} the environment violates $\mathsf{Parity}(C)$ almost-surely, and, conversely, under every environment policy, almost every play settles in an EC. So, under every environment policy, the probability of violating $\mathsf{Parity}(C)$ is at most the probability of reaching W_{odd}, and the environment attains it by playing optimally for reachability of W_{odd} and then enforcing an odd dominant color. The following lemma establishes this reduction, and is proved in Appendix E.

Lemma 6. *Given a linearly defined RMC $\mathcal{R} = (\mathcal{S}, \mathcal{P})$ it holds that $\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Parity}(C)) = \boldsymbol{v}_s^{\mathcal{R}}\big(\mathsf{Safe}(\mathcal{S} \setminus W_{\mathrm{odd}})\big)$.*

The right-hand side is exactly the value computed by the LP $(\star)$ of Section 3.1, instantiated with unsafe set $U = W_{\mathrm{odd}}$. This yields the parity values, and optimal positional environment policies, in polynomial time:

Theorem 7. *Given a linearly defined RMC $\mathcal{R}$ with parity objective $\mathsf{Parity}(C)$, for all $s \in \mathcal{S}$ the value $\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Parity}(C))$, together with a positional environment policy that attains it, can be computed in time polynomial in the size of $\mathcal{R}$.*

4 Quantitative Analysis of ω-Regular RMDPs

Algorithm 2 presents our policy-iteration algorithm for the quantitative parity problem on linearly defined RMDPs. The algorithm ranges over *positional* agent policies and, in each iteration, tries to improve the current policy by calling the Improve procedure of Algorithm 3. Throughout this section we fix an RMDP $\mathcal{M}$ and a parity objective $\mathsf{Parity}(C)$. In what follows we describe the algorithm and its ingredients, and state its correctness and complexity guarantees. The proofs are deferred to Appendix G.

Positional Determinacy. Before diving into the algorithm, we establish that for parity objectives, positional optimal policies exist for both the agent and the environment. Fixing a positional agent policy σ induces the RMC $\mathcal{M}^{\sigma}$ whose

only action at each state s is $\sigma(s)$ and whose uncertainty set is $\mathcal{P}^\sigma(s) = \mathcal{P}(s, \sigma(s))$; we write $\boldsymbol{v}^\sigma$ for its value vector, $\boldsymbol{v}^\sigma_s = \inf_\tau \Pr^{\sigma,\tau}_s[\varphi]$. Our analysis rests on the following determinacy result, which is where polytopic properties of the uncertainty sets are used.

Proposition 8. *Let $\mathcal{M}$ be a linearly defined RMDP with parity objective $\varphi = \mathsf{Parity}(C)$. There exists a positional agent policy σ^* and a positional environment policy τ^* where $\inf_\tau \Pr^{\sigma^*,\tau}_s[\varphi] = \sup_\sigma \Pr^{\sigma,\tau^*}_s[\varphi] = \boldsymbol{v}^{\mathcal{M}}_s(\varphi)$ for every s.*

The proof uses the reduction introduced in (Chatterjee et al. 2024) to reduce $\mathcal{M}$ to a finite turn-based stochastic parity game in which the environment's choice is replaced by an adversarial choice of a corner of the uncertainty set. The reduction is value-preserving and maps positional strategies of the game back to positional policies of $\mathcal{M}$, so positional determinacy of stochastic games implies the proposition. The number of corners, and hence the size of the game, may however be exponential, so this route requires exponential space; our policy iteration algorithm below instead runs in polynomial space. Combined with the polynomial-time RMC solver of Section 3, Proposition 8 also settles the complexity of the decision problem.

Corollary 9. *Given a linearly defined RMDP $\mathcal{M}$, a state $s \in \mathcal{S}$ and a rational $\lambda \in [0, 1]$, the problem of deciding whether $\boldsymbol{v}^{\mathcal{M}}_s(\varphi) \geq \lambda$ is (i) in NP $\cap$ coNP and (ii) at least as hard as the analogous problem in turn-based stochastic parity games.*

Hardness implies that a polynomial-time algorithm here would also solve turn-based stochastic parity games, a long-standing open problem.

Policy Iteration. The policy iteration algorithm PI (Algorithm 2) starts from an arbitrary positional agent policy and iteratively calls the Improve procedure (Algorithm 3) to improve it. Once the policy can not be improved further, the algorithm returns it together with its value vector. To this end, the main technical ingredient of the algorithm lies in the Improve procedure, explained next.

Policy Improvement. The Improve procedure (Algorithm 3) takes as input a positional agent policy σ, computes its value vector $\boldsymbol{v}^\sigma$ (using the method of Section 3), and then tries to improve σ in one of the following two ways:

- *Quantitative* improvement: A quantitative improvement is possible when for some state s an action a exists such that its worst-case one-step value $\nu(s, a) = \min_{p \in \mathcal{P}(s,a)} p \cdot \boldsymbol{v}^\sigma$ strictly exceeds the current value $\boldsymbol{v}^\sigma_s$. In this case, the agent switches to such an action a for every state s where an improvement is possible and returns the new policy (Lines 2–9 of Algorithm 3). The following lemma shows that the resulting policy is strictly better than σ:
  Lemma 10. *Let $\sigma' = \mathsf{Improve}(\mathcal{M}, \sigma)$ be obtained by a quantitative improvement. Then $\boldsymbol{v}^{\sigma'}_s \geq \boldsymbol{v}^\sigma_s$ for all $s \in \mathcal{S}$, and $\boldsymbol{v}^{\sigma'}_s > \boldsymbol{v}^\sigma_s$ for some $s \in \mathcal{S}$.*
- *Qualitative* improvement: If a quantitative improvement is not available, the algorithm checks whether a qualitative improvement is possible. To this end, the algorithm iterates over the value classes of $\boldsymbol{v}^\sigma$:

Algorithm 2: $\mathsf{PI}(\mathcal{M})$

Input: Linearly Defined RMDP $\mathcal{M}$, coloring C
Output: optimal agent policy σ^* and value vector $\boldsymbol{v}^{\mathcal{M}}(\varphi)$

1: pick an arbitrary positional policy σ
2: **repeat**
3: $\quad \sigma_{\text{old}} \leftarrow \sigma$
4: $\quad \sigma \leftarrow \mathsf{Improve}(\mathcal{M}, \sigma_{\text{old}})$
5: **until** $\sigma = \sigma_{\text{old}}$
6: **return** σ and $\boldsymbol{v}^\sigma$

Value Classes, Tight Actions, Tight Faces. For $r \in [0, 1]$, the *value class* $\mathsf{VC}^\sigma(r) = \{s \in \mathcal{S} \mid \boldsymbol{v}^\sigma_s = r\}$ is the set of states of value r under σ. There are at most $|\mathcal{S}|$ non-empty value classes. An action $a \in \mathcal{A}$ is *tight* at s (with respect to $\boldsymbol{v}^\sigma$) if $\nu(s, a) = \boldsymbol{v}^\sigma_s$; we write $\mathcal{A}^{\boldsymbol{v}^\sigma}(s)$ for the set of tight actions at s, and call $\mathcal{P}^{\boldsymbol{v}^\sigma}(s, a)$ the *tight face* at (s, a) when a is tight at s. If a is tight at s then every $p \in \mathcal{P}(s, a)$ satisfies $p \cdot \boldsymbol{v}^\sigma \geq \boldsymbol{v}^\sigma_s$; if a is not tight at s and no action improves at s, then $\min_p p \cdot \boldsymbol{v}^\sigma < \boldsymbol{v}^\sigma_s$.
For every value class $U_r = \mathsf{VC}^\sigma(r)$, the algorithm will then construct an RMDP $\mathcal{M}_r$ together with a coloring C_r defined as follows: (i) the states of $\mathcal{M}_r$ are those of U_r together with a fresh terminal state $\perp$, (ii) the agent's actions are its tight actions, (iii) the uncertainty sets of $\mathcal{M}_r$ are the tight faces of $\mathcal{M}$ inside U_r and the singleton $\{\delta_\perp\}$ (dirac distribution over $\perp$) outside U_r, and (iv) the coloring C_r is the same as C inside U_r and assigns a fresh dominating odd color to $\perp$.
The algorithm then calls the almost-sure parity procedure of (Asadi et al. 2026b) on $(\mathcal{M}_r, \mathsf{Parity}(C_r))$ to compute the set $W_r \subseteq U_r$ as the set of state where the agent can guarantee $\mathsf{Parity}(C_r)$'s satisfaction with probability 1 together with its corresponding optimal policy σ_r (lines 10–15). If $W_r \neq \emptyset$, then the agent can improve its policy by switching to σ_r on W_r. The following lemma shows that the resulting policy is strictly better than σ:
Lemma 11. *Let $\sigma' = \mathsf{Improve}(\mathcal{M}, \sigma)$ be obtained by a qualitative improvement, i.e. a quantitative improvement is not possible and $W := \bigcup_{r<1} W_r \neq \emptyset$. Then $\boldsymbol{v}^{\sigma'}_s \geq \boldsymbol{v}^\sigma_s$ for all $s \in \mathcal{S}$, and $\boldsymbol{v}^{\sigma'}_s > \boldsymbol{v}^\sigma_s$ for all $s \in W$.*
Intuitively, Lemma 11 follows by the fact that if the environment tries to keep the run inside U_r, then the agent can guarantee $\mathsf{Parity}(C_r)$ with probability 1, and if the environment tries to escape U_r, then the agent can guarantee a value strictly larger than r by the definition of tight faces. The full proof is given in Appendix G.

Correctness and Complexity. It follows directly from Lemma 10 and Lemma 11 that every call to Improve either returns a strictly better policy or the same policy, in which case a fixpoint is reached. Hence, the policy is indeed improved in every iteration of PI. Since there are only finitely many positional policies, the algorithm must terminate. The following theorem establishes the correctness and complexity of the algorithm:

Theorem 12. *Given a linearly defined RMDP $\mathcal{M}$ and a parity objective φ:*

Algorithm 3: $\mathsf{Improve}(\mathcal{M}, \sigma)$

Input: RMDP $\mathcal{M}$, coloring C, agent policy σ
Output: policy σ' with $\sigma' = \sigma$ or $\boldsymbol{v}^{\sigma'} \gtrdot \boldsymbol{v}^{\sigma}$

$\boldsymbol{v}^{\sigma} \leftarrow \textsc{SolveRMC}(\mathcal{M}^{\sigma}, \mathsf{Parity}(C))$ ▷ Section 3
for each $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
  $\nu(s, a) \leftarrow \min_{p \in \mathcal{P}(s,a)} p \cdot \boldsymbol{v}^{\sigma}$ ▷ one linear program
end for
$I \leftarrow \{\, s \in \mathcal{S} \mid \max_{a \in \mathcal{A}} \nu(s, a) > \boldsymbol{v}^{\sigma}_s \,\}$
if $I \neq \emptyset$ **then**
  $\sigma'(s) \leftarrow \begin{cases} \arg\max_{a \in \mathcal{A}} \nu(s, a) & \text{if } s \in I, \\ \sigma(s) & \text{otherwise.} \end{cases}$
  return σ'
end if
$W \leftarrow \emptyset$
for each value $r < 1$ of $\boldsymbol{v}^{\sigma}$ with $U_r = \mathsf{VC}^{\sigma}(r) \neq \emptyset$ **do**
  Construct $\mathcal{M}_r$ and coloring C_r from ν
  $(W_r, \sigma_r) \leftarrow \textsc{AlmostSureParity}(\mathcal{M}_r, \mathsf{Parity}(C_r))$
  $W \leftarrow W \cup W_r$
end for
if $W \neq \emptyset$ **then**
  $\sigma'(s) \leftarrow \begin{cases} \sigma_r(s) & \text{if } s \in W_r \text{ for some } r, \\ \sigma(s) & \text{otherwise.} \end{cases}$
  return σ'
end if
return σ

- $\mathsf{PI}(\mathcal{M})$ *terminates after at most* $|\mathcal{A}|^{|\mathcal{S}|}$ *calls to* $\mathsf{Improve}$.
- $\mathsf{PI}(\mathcal{M})$ *returns an optimal positional agent policy* σ^* *together with the value vector* $\boldsymbol{v}^{\mathcal{M}}(\varphi)$.
- $\mathsf{PI}(\mathcal{M})$ *runs in polynomial space.*

The polynomial space requirement is specifically important because the reduction in the proof of Proposition 8 explodes the state-space exponentially.

5 Experimental Results

We implemented our proposed method and conducted a series of experiments to evaluate its performance. The experiments were designed to assess the effectiveness and applicability of our approach for quantitative analysis of both general parity objectives and the special class of reachability objectives. We compare our method against the technique that reduces the RMDP into a stochastic game (as in Appendix F) and solves it using the policy iteration algorithm of (Chatterjee and Henzinger 2006a).

The experiments are designed to answer the following research questions:

- **RQ1:** How does our method perform in terms of computational efficiency compared to the baseline approach?
- **RQ2:** How does our method scale with increasing problem size and complexity?
- **RQ3:** What are the limitations of our method, and in what scenarios does it outperform the baseline approach?

Experimental Setup. We implemented our approach and the baseline method in Python, and used them to compute the quantitative value of the benchmarks explained next. All experiments were conducted on a machine with an Intel Core Ultra 5 225U processor and 16GB of RAM. We used a timeout of 120 seconds for each experiment.

Benchmarks. We conducted experiments on three classes of benchmarks, where the branching factor m denotes the number of possible successors of a state under a fixed action. Garnet and Inventory Management support both objectives, while Frozen Lake supports only reachability. In all cases, the uncertainty sets are L_∞ or L_1 balls around the nominal distributions. The benchmarks are summarized as follows (full details are given in Appendix H):

- **Garnet** (adapted from (Archibald, McKinnon, and Thomas 1995)) is a family of randomly generated MDPs. The state-space consists of N states partitioned into k subsets, $G_0, \ldots, G_k$. Every state has three available actions, each leading to $m = \lceil \sqrt{N} \rceil$ possible successors. Taking an action from a state in G_0 will either lead to a state in G_0 or to a state in one of the other subsets, while taking an action from a state in G_i for $i > 0$ will lead to a state in G_i. The transition probabilities are drawn uniformly at random from the probability simplex. As a reachability objective, each one of the subsets is considered to have a target and an trap state; the goal is to reach the target state of some subset. As a parity objective, each state is assigned a random color and the goal is to satisfy the respective parity condition.
- **Inventory Management** (adapted from (Iyengar 2005)) models a warehouse with capacity N whose state is the stock level. At each step the agent orders new units against an uncertain demand. Specifically, the agent will choose to buy $a \in \{0, \ldots, a_{\max}\}$ new units and the demand d is drawn uniformly from $\{0, \ldots, d_{\max}\}$. The state will then transition from s to $s + a - d$; if the demand exceeds the available stock ($d > s + a$), or the delivery pushes the stock over the capacity ($s + a - d > N$), the run instead ends in one of two absorbing failure states. We fix $d_{\max} = \lceil \sqrt{N} \rceil$ and $a_{\max} = \lceil 0.75\, d_{\max} \rceil$. Once the stock reaches a threshold T, a larger order option becomes available. The reachability objective is to reach a stock level above a threshold, while the parity objective is to keep the stock level above a threshold infinitely often.
- **Frozen Lake** (Brockman et al. 2016) is the classic grid world where the agent walks on a $k \times k$ grid from the top-left cell toward the bottom-right. Each move proceeds in the intended direction or in one of the two perpendicular ones with probability $\frac{1}{3}$ each, and a random 20% of the cells are holes, which are absorbing failures.

Results. Figure 1 reports the solve times of both algorithms against the state-space size, on a log–log scale; every configuration is grown until each algorithm, in turn, exceeds the 120-second timeout, marked by a $\times$. The returned values of both approaches were checked to match on all benchmarks. The picture cleanly separates two regimes, governed by the branching factor m.

On **Garnet** and **Inventory Management**, where m grows with the instance size, our algorithm scales substantially better than the baseline, across both norms and both objectives (panels a, b, d, e). The baseline's cost is driven by the num-

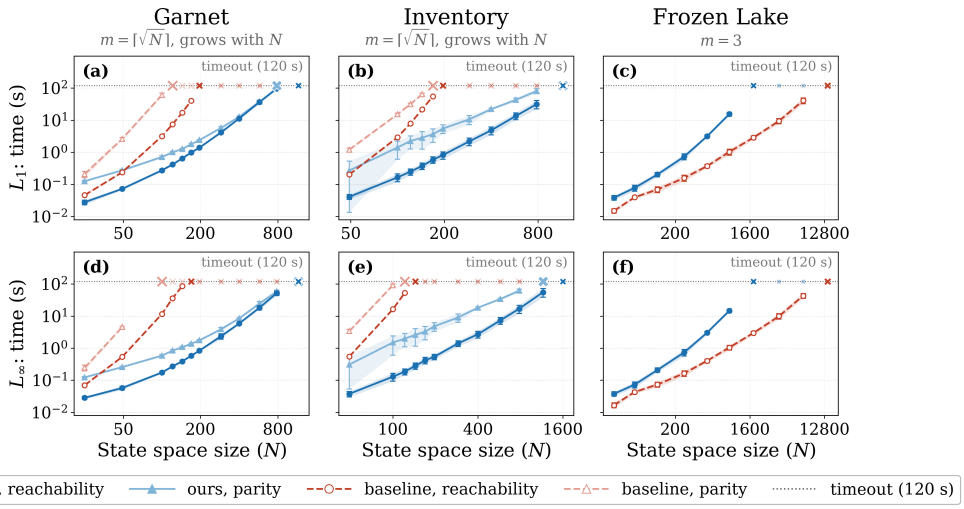

Figure 1: Solve time as a function of the number of states N, with one benchmark per column and one uncertainty norm per row: L_1 (a–c) and L_∞ (d–f). Blue solid lines show our algorithm and red dashed lines the baseline; within each color, dark circles mark the reachability objective and light triangles the parity objective (Frozen Lake is reachability-only). Each point is the mean over 10 random seeds, with error bars and shaded bands giving one standard deviation. The dotted horizontal line is the 120s timeout, and a $\times$ marks a timed-out.

ber of extreme points of the uncertainty set, which grows rapidly with m: it already exceeds the timeout at a few hundred states, whereas our algorithm keeps solving an order of magnitude larger instances (up to $n \approx 1200$ on Inventory). At the largest sizes where the baseline still completes, our algorithm is faster by roughly one to two orders of magnitude, and this gap widens as the instances grow. Consistent with the extra work of the parity decomposition, parity (light curves) is somewhat more expensive than reachability (dark curves) for both algorithms, so the baseline times out slightly earlier on parity than on reachability.

Frozen Lake is the opposite regime (panels c, f). Here the branching factor is fixed at $m \leq 3$, so each uncertainty set has only a handful of extreme points and the reduced stochastic game stays close in size to the original RMDP. The baseline therefore has the advantage: it is roughly an order of magnitude faster than our algorithm at matched sizes and scales to far larger grids (beyond $n \approx 10^4$), while the overhead of our linear-programming inner solver dominates.

These results answer our research questions directly. Our method is the more efficient choice (RQ1) and scales to markedly larger instances (RQ2) precisely when the branching factor m is large, since its per-iteration cost is polynomial in the size of the RMDP rather than in the number of extreme points of the uncertainty sets. Its limitation (RQ3) shows up when m is small: when the uncertainty sets have few extreme points, the reduction to a stochastic game produces a compact game that the baseline solves faster.

Finally, we note that the number of iterations both in our approach and the baseline is observed to be smaller than the theoretical worst-case bound, which is exponential in the number of states. This is consistent with the known behavior of policy iteration on stochastic games, where the number of iterations is often observed to be polynomial in practice, despite the worst-case exponential bound. This suggests that the practical performance of our algorithms may be significantly better than the theoretical worst-case analysis would suggest (See Appendix H for details).

6 Conclusion and Future Works

In this paper, we study the quantitative parity problem for (s, a)-rectangular RMDPs with linearly defined uncertainty sets. We give a polynomial-time algorithm for robust Markov chains, casting robust safety as a linear program over occupation measures and reducing parity to safety through a maximal-end-component analysis, and use it as a subroutine in a policy-iteration algorithm for RMDPs that combines quantitative one-step improvements with qualitative almost-sure improvements, runs in polynomial space, and computes the exact values together with optimal positional policies for both the agent and the environment. The decision problem for linearly defined RMDPs is in $NP \cap coNP$ and as hard as turn-based stochastic parity games. An interesting future question is whether the exact value is computable in polynomial time. Other promising directions are to extend beyond linearly defined sets to nonlinear convex uncertainty, such as ellipsoidal or divergence-based balls, to relax (s, a)-rectangularity, and to study the learning setting in which the uncertainty sets are estimated from samples.

References

Archibald, T. W.; McKinnon, K. I.; and Thomas, L. C. 1995. On the generation of markov decision processes. *Journal of the Operational Research Society*, 46(3): 354–361.

Asadi, A.; Chatterjee, K.; Goharshady, E. K.; Karrabi, M.; Montaseri, A.; and Pagano, C. 2026a. Strongly Polynomial Time Complexity of Policy Iteration for L_∞ Robust MDPs.

In *Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research.

Asadi, A.; Chatterjee, K.; Goharshady, E. K.; Karrabi, M.; and Shafiee, A. 2026b. Qualitative Analysis of ω-Regular Objectives on Robust MDPs. In *AAAI*, 36137–36145. AAAI Press.

Baier, C.; and Katoen, J. 2008. *Principles of model checking*. MIT Press.

Behzadian, B.; Petrik, M.; and Ho, C. P. 2021. Fast Algorithms for L_∞-constrained S-rectangular Robust MDPs. *Advances in Neural Information Processing Systems*, 34: 25982–25992.

Bozkurt, A. K.; Wang, Y.; Zavlanos, M. M.; and Pajic, M. 2020. Control Synthesis from Linear Temporal Logic Specifications using Model-Free Reinforcement Learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 10349–10355. IEEE.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. ArXiv:1606.01540, arXiv:1606.01540.

Chatterjee, K.; Goharshady, E. K.; Karrabi, M.; Novotný, P.; and Žikelić, Đ. 2024. Solving long-run average reward robust MDPs via stochastic games. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 6707–6715.

Chatterjee, K.; and Henzinger, M. 2014. Efficient and Dynamic Algorithms for Alternating Büchi Games and Maximal End-Component Decomposition. *Journal of the ACM*, 61(3): 15:1–15:40.

Chatterjee, K.; and Henzinger, T. A. 2006a. Strategy Improvement and Randomized Subexponential Algorithms for Stochastic Parity Games. In *STACS 2006 – 23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, 512–523. Springer.

Chatterjee, K.; and Henzinger, T. A. 2006b. Strategy Improvement for Stochastic Rabin and Streett Games. In *CONCUR 2006 – Concurrency Theory*, volume 4137 of *Lecture Notes in Computer Science*, 375–389. Springer.

Chatterjee, K.; and Henzinger, T. A. 2012. A survey of stochastic ω-regular games. *Journal of Computer and System Sciences*, 78(2): 394–413.

Givan, R.; Leach, S. M.; and Dean, T. L. 2000. Bounded-parameter Markov decision processes. *Artif. Intell.*, 122(1-2): 71–109.

Hahn, E. M.; Perez, M.; Schewe, S.; Somenzi, F.; Trivedi, A.; and Wojtczak, D. 2019. Omega-Regular Objectives in Model-Free Reinforcement Learning. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 11427 of *Lecture Notes in Computer Science*, 395–412. Springer.

Ho, C. P.; Petrik, M.; and Wiesemann, W. 2021. Partial policy iteration for l1-robust markov decision processes. *Journal of Machine Learning Research*, 22(275): 1–46.

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

Kaufman, D. L.; and Schaefer, A. J. 2013. Robust Modified Policy Iteration. *INFORMS J. Comput.*, 25(3): 396–410.

Moos, J.; Hansel, K.; Abdulsamad, H.; Stark, S.; Clever, D.; and Peters, J. 2022. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction*, 4(1): 276–315.

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

Parys, P. 2019. Parity Games: Zielonka's Algorithm in Quasi-Polynomial Time. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 10:1–10:13. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust Adversarial Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, 2817–2826. PMLR.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley.

Suilen, M.; Badings, T.; Bovy, E. M.; Parker, D.; and Jansen, N. 2024. Robust Markov Decision Processes: A Place Where AI and Formal Methods Meet. *arXiv preprint arXiv:2411.11451*.

Tewari, A.; and Bartlett, P. L. 2007. Bounded Parameter Markov Decision Processes with Average Reward Criterion. In *Learning Theory, 20th Annual Conference on Learning Theory (COLT)*, volume 4539 of *Lecture Notes in Computer Science*, 263–277. Springer.

Wang, Y.; Velasquez, A.; Atia, G. K.; Prater-Bennette, A.; and Zou, S. 2023. Robust Average-Reward Markov Decision Processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15215–15223.

Wang, Y.; and Zou, S. 2021. Online Robust Reinforcement Learning with Model Uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34.

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

Wolff, E. M.; Topcu, U.; and Murray, R. M. 2012. Robust control of uncertain Markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 3372–3379. IEEE.

Zielonka, W. 1998. Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees. *Theoretical Computer Science*, 200(1-2): 135–183.

A Related Work

Robust MDPs. RMDPs were introduced for sequential decision making under uncertain transition probabilities, with the foundational works of (Iyengar 2005; Nilim and El Ghaoui 2005) establishing robust dynamic programming for rectangular uncertainty. We refer to Suilen et al. (2024) for a recent survey. Subsequent works studied general robust policy-iteration methods (Wiesemann, Kuhn, and Rustem 2013; Kaufman and Schaefer 2013), as well as faster algorithms for specific uncertainty classes such as L_1 and L_∞ (Ho, Petrik, and Wiesemann 2021; Behzadian, Petrik, and Ho 2021). Recent works study exact computation and the complexity of policy iteration for discounted RMDPs (Asadi et al. 2026a), while others consider the average-reward and bounded-parameter settings (Wang et al. 2023; Chatterjee et al. 2024; Tewari and Bartlett 2007; Givan, Leach, and Dean 2000). All of these results consider quantitative objectives such as discounted-sum, average reward, and finite horizon, whereas our focus is logical objectives.

Parity Objectives and Strategy Improvement. Parity objectives and their quantitative analysis have been studied extensively for MDPs and two-player stochastic games. We refer to Baier and Katoen (2008); Chatterjee and Henzinger (2012) for an overview, and to Zielonka (1998); Parys (2019) for algorithms solving parity games. Our policy-iteration algorithm is, in spirit, close to the strategy-improvement paradigm for stochastic parity and Rabin games (Chatterjee and Henzinger 2006a,b), which likewise iterates over positional strategies and alternates value improvements with qualitative sub-computations. The essential difference is that in our robust setting the environment ranges over a continuum of distributions in an uncertainty polytope rather than over the finitely many vertices of a stochastic game. Our improvement therefore works directly with the linearly defined uncertainty sets, through the value classes, tight actions, and tight faces of Section 4, and never constructs the exponentially larger game.

Logical Objectives on Uncertain MDPs. The closest prior work considers temporal-logic control of uncertain MDPs under a constant-support assumption and approximates satisfaction probabilities using value iteration (Wolff, Topcu, and Murray 2012). Asadi et al. (2026b) remove structural assumptions and solve the positive and almost-sure reachability and parity problems using uncertainty-set oracles. Our qualitative improvement step invokes their almost-sure parity procedure as a sub-routine, whereas the exact quantitative problem we solve here remained open.

Reinforcement Learning. RMDPs are the model underlying robust reinforcement learning, where the goal is to learn policies that hedge against misspecification of the transition model. Adversarial and model-uncertainty formulations optimize the worst-case return over an uncertainty set (Pinto et al. 2017; Wang and Zou 2021), and we refer to Moos et al. (2022) for a survey. This line targets quantitative reward objectives, whereas we compute exact values of ω-regular objectives. Orthogonally, a body of work learns policies for ω-regular and temporal-logic objectives in the non-robust setting, typically by reducing the objective to a reachability reward on a product with an automaton (Hahn et al. 2019; Bozkurt et al. 2020). Our setting combines the two axes: exact quantitative analysis of ω-regular objectives against adversarial transition uncertainty.

B Linear Descriptions of L_∞ and L_1 Uncertainty Sets

Recall the Assumption of Section 3: the uncertainty set of each state s is a polytope $\mathcal{P}(s) = \{p \in \Delta(\mathcal{S}) : A_s\, p \le b_s\}$. Here we show that the two most common uncertainty models, the L_∞ and L_1 balls around a nominal distribution, fit this form (the L_1 case after introducing auxiliary variables). Fix a state s with nominal distribution $\hat{p}_s \in \Delta(\mathcal{S})$ and radius $\delta_s \ge 0$; write $\hat{p}_{s,s'}$ for the nominal probability of the successor s'.

L_∞ ball. The set

$$\mathcal{P}(s) = \left\{\, p \in \Delta(\mathcal{S}) : \|p - \hat{p}_s\|_\infty \le \delta_s \,\right\}$$

is cut out, with no auxiliary variables, by the box constraints

$$-\delta_s \;\le\; p_{s'} - \hat{p}_{s,s'} \;\le\; \delta_s \qquad \text{for every } s' \in \mathcal{S},$$

i.e. $p_{s'} \le \hat{p}_{s,s'} + \delta_s$ and $-p_{s'} \le \delta_s - \hat{p}_{s,s'}$. Stacking these $2|\mathcal{S}|$ inequalities gives $A_s = \left[\begin{smallmatrix} I \\ -I \end{smallmatrix}\right]$ and $b_s = \left[\begin{smallmatrix} \hat{p}_s + \delta_s \mathbf{1} \\ -\hat{p}_s + \delta_s \mathbf{1} \end{smallmatrix}\right]$, so $\mathcal{P}(s) = \{p \in \Delta(\mathcal{S}) : A_s\, p \le b_s\}$ exactly as in the Assumption.

L_1 ball. The set

$$\mathcal{P}(s) = \left\{\, p \in \Delta(\mathcal{S}) : \|p - \hat{p}_s\|_1 \le \delta_s \,\right\} = \left\{\, p \in \Delta(\mathcal{S}) : \textstyle\sum_{s' \in \mathcal{S}} |p_{s'} - \hat{p}_{s,s'}| \le \delta_s \,\right\}$$

has, in general, exponentially many facets, but it becomes linear after introducing one auxiliary variable $y_{s'}$ per successor to model $|p_{s'} - \hat{p}_{s,s'}|$:

$$y_{s'} \ge p_{s'} - \hat{p}_{s,s'}, \qquad y_{s'} \ge \hat{p}_{s,s'} - p_{s'}, \qquad (s' \in \mathcal{S}), \qquad \sum_{s' \in \mathcal{S}} y_{s'} \le \delta_s.$$

At an optimum each $y_{s'}$ equals $|p_{s'} - \hat{p}_{s,s'}|$, so the projection onto the p-coordinates of the polytope defined by these $2|\mathcal{S}| + 1$ inequalities (over the $2|\mathcal{S}|$ variables (p, y)) is exactly the L_1 ball. Writing the constraints as $A_s \left[\begin{smallmatrix} p \\ y \end{smallmatrix}\right] \le b_s$ presents $\mathcal{P}(s)$ as the projection of a polytope, using only a linear number of auxiliary variables and inequalities.

Extending the LP to auxiliary variables. The L_1 description uses auxiliary variables, so $\mathcal{P}(s)$ is a *projected* polytope rather than the plain form $\{p : A_s p \le b_s\}$. The program $(\star)$ and its correctness extend to this case without change. Indeed, the linearisation of Lemma 16 rewrites the admissibility constraint $p \in \mathcal{P}(s)$, with $p = \boldsymbol{f}_s / x_s$, by multiplying through by $x_s \ge 0$; applying the *same* scaling to the auxiliary variables (replacing y by $y' = x_s\, y$) turns $A_s \left[\begin{smallmatrix} p \\ y \end{smallmatrix}\right] \le b_s$ into the linear constraint $A_s \begin{bmatrix} \boldsymbol{f}_s \\ y' \end{bmatrix} \le x_s\, b_s$ over the flow and (scaled) auxiliary variables. Thus, for L_∞, L_1, and general (possibly projected) polytopes alike, the value $\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T}))$ is computed by a single polynomial-size linear program, and Lemma 2 applies uniformly.

C Proofs of Lemma 2 and Theorem 3

Throughout this section we fix a linearly defined RMC $\mathcal{R} = (\mathcal{S}, \mathcal{P})$ with, for every $s \in \mathcal{S}$, $\mathcal{P}(s) = \{p \in \Delta(\mathcal{S}) : A_s\, p \le b_s\}$, a safe set $\mathcal{T} \subseteq \mathcal{S}$, and an initial state s_0. A *positional* environment policy is a map $\tau \colon \mathcal{S} \to \Delta(\mathcal{S})$ with $\tau(s) \in \mathcal{P}(s)$ for all s; we write $p^\tau_{s,s'} = \tau(s)(s')$ for its one-step probabilities. Since an RMC has a single action, the safety value from Section 3.1 is

$$\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T})) = \inf_\tau \Pr^\tau_s[\mathsf{Safe}(\mathcal{T})],$$

the infimum ranging over all environment policies. The preprocessing computes the sets

$$\mathcal{S}^{=1} = \{\, s \in \mathcal{S} : \boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T})) = 1 \,\}, \quad \mathcal{S}^{=0} = \{\, s \in \mathcal{S} : \boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T})) = 0 \,\}, \quad \mathcal{S}^{?} = \mathcal{S} \setminus (\mathcal{S}^{=0} \cup \mathcal{S}^{=1}),$$

and makes the states of $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ absorbing sinks. Every unsafe state (outside $\mathcal{T}$) has safety value 0, so $\mathcal{S} \setminus \mathcal{T} \subseteq \mathcal{S}^{=0}$ and hence $\mathcal{S}^{?} \subseteq \mathcal{T}$. We prove that for $s_0 \in \mathcal{S}^{?}$ the optimum of $(\star)$ equals $\boldsymbol{v}_{s_0}^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T}))$; for $s_0 \in \mathcal{S}^{=1}$ (resp. $\mathcal{S}^{=0}$) the value is 1 (resp. 0) by definition.

A *reachability* objective $\mathsf{Reach}(T)$ specifies the set of all runs that reach a state in T eventually, i.e.

$$\mathsf{Reach}(T) = \{\pi \in \mathcal{S}^\omega | \exists i : \pi_i \in T\}$$

C.1 Occupation Measures

Definition 13 (Expected visits and flows)**.** *Fix a positional environment policy τ. For $s, s' \in \mathcal{S}$, the* expected number of visits *to s and the* expected flow *along (s, s'), both from s_0, are*

$$x^\tau_s = \sum_{t=0}^{\infty} \Pr^\tau_{s_0}[s_t = s], \qquad f^\tau_{s,s'} = \sum_{t=0}^{\infty} \Pr^\tau_{s_0}[s_t = s,\ s_{t+1} = s'].$$

By the Markov property of a positional τ, $\Pr^\tau_{s_0}[s_t = s,\, s_{t+1} = s'] = p^\tau_{s,s'}\, \Pr^\tau_{s_0}[s_t = s]$; summing over t gives the factorisation

$$f^\tau_{s,s'} = x^\tau_s\, p^\tau_{s,s'}, \qquad \text{hence} \qquad \sum_{s' \in \mathcal{S}} f^\tau_{s,s'} = x^\tau_s \quad \text{whenever } x^\tau_s < \infty. \tag{1}$$

Whenever $x^\tau_s < \infty$ for all $s \in \mathcal{S}^{?}$, conditioning on the state at time $t-1$ and summing gives Cons, and Split is the second identity in (1); together with NNeg (immediate) the pair (x^τ, f^τ) then satisfies the constraints of $(\star)$ by Lemma 16. Moreover, since $\mathcal{S}^{=1}$ is absorbing, a play crosses from $\mathcal{S}^{?}$ into $\mathcal{S}^{=1}$ at most once and exactly when it reaches $\mathcal{S}^{=1}$, so the objective of $(\star)$ evaluated at this occupation is

$$\sum_{s \in \mathcal{S}^{?},\, w \in \mathcal{S}^{=1}} f^\tau_{s,w} = \Pr^\tau_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})]. \tag{2}$$

C.2 Correctness of the Preprocessing

Lemma 14 (Qualitative preprocessing)**.** *The sets $\mathcal{S}^{=1}$ and $\mathcal{S}^{=0}$ are computable in polynomial time, and making states in $\mathcal{S}^{=0}$ unsafe absorbing and states in $\mathcal{S}^{=1}$ safe absorbing does not change the safety value of any state in $\mathcal{S}^{?}$.*

Proof. The qualitative sets $\mathcal{S}^{=1}$ and $\mathcal{S}^{=0}$ are computable in polynomial time by (Asadi et al. 2026b). From a state of $\mathcal{S}^{=1}$ the play is safe under *every* policy, and from a state of $\mathcal{S}^{=0}$ some policy is unsafe almost surely; making these states absorbing therefore only relabels play that has already committed to value 1 or 0, and does not change any safety value. □

Lemma 15. *Let $\tau^\star$ be a positional environment policy attaining $\boldsymbol{v}_{s_0}^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T}))$. Then from $s_0 \in \mathcal{S}^{?}$ the play reaches $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ almost surely; consequently $x^{\tau^\star}_s < \infty$ for every $s \in \mathcal{S}^{?}$, and under $\tau^\star$ a play is safe if and only if it reaches $\mathcal{S}^{=1}$, so that $\Pr^{\tau^\star}_{s_0}[\mathsf{Safe}(\mathcal{T})] = \Pr^{\tau^\star}_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})]$.*

Proof. Suppose the induced Markov chain has a BSCC B reachable from s_0 with $B \subseteq \mathcal{S}^?$. Since $\mathcal{S}^? \subseteq \mathcal{T}$ and B is a BSCC, the play from any $u \in B$ stays in $B \subseteq \mathcal{T}$ forever and is therefore safe with probability 1. Fix such a u. As $u \in \mathcal{S}^?$, we have $\boldsymbol{v}^{\mathcal{R}}_u(\mathsf{Safe}(\mathcal{T})) < 1$, so some policy τ' is unsafe from u with positive probability. Consider the policy that follows $\tau^\star$ until u is first visited and then switches to τ'. Since $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ is absorbing, u is reached from s_0 with some probability $q > 0$ along a path inside $\mathcal{S}^? \subseteq \mathcal{T}$, and under $\tau^\star$ these are exactly the safe plays that visit u; replacing their safe continuation by τ' hence lowers the safety probability from s_0 strictly below $\Pr^{\tau^\star}_{s_0}[\mathsf{Safe}(\mathcal{T})]$, contradicting the optimality of $\tau^\star$ at s_0. Hence every BSCC reachable from s_0 meets $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$, so the play reaches $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ almost surely, the hitting time τ has finite expectation, and $x^{\tau^\star}_s \le \mathbb{E}^{\tau^\star}_{s_0}[\tau] < \infty$ for every $s \in \mathcal{S}^?$. Finally, once absorbed in $\mathcal{S}^{=1}$ the play is safe, whereas from $\mathcal{S}^{=0}$ the optimal $\tau^\star$ is unsafe almost surely; hence, under $\tau^\star$, being safe coincides with reaching $\mathcal{S}^{=1}$, giving $\Pr^{\tau^\star}_{s_0}[\mathsf{Safe}(\mathcal{T})] = \Pr^{\tau^\star}_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})]$. □

C.3 Linearization of the Uncertainty

Lemma 16. *Fix $s \in \mathcal{S}^?$, a scalar $x_s \ge 0$, and a vector $\boldsymbol{f}_s = (f_{s,s'})_{s' \in \mathcal{S}}$. The constraints $\boldsymbol{f}_s \ge 0$, $\sum_{s' \in \mathcal{S}} f_{s,s'} = x_s$, and $A_s \boldsymbol{f}_s \le x_s b_s$ hold if and only if there exists $p \in \mathcal{P}(s)$ with $\boldsymbol{f}_s = x_s p$.*

Proof. ($\Leftarrow$) Given $p \in \mathcal{P}(s)$ with $\boldsymbol{f}_s = x_s p$: non-negativity and $\sum_{s'} f_{s,s'} = x_s \sum_{s'} p_{s'} = x_s$ follow from $p \in \Delta(\mathcal{S})$, and multiplying $A_s p \le b_s$ by $x_s \ge 0$ gives $A_s \boldsymbol{f}_s \le x_s b_s$.

($\Rightarrow$) If $x_s > 0$, set $p = \boldsymbol{f}_s / x_s$. Then $p \ge 0$, $\sum_{s'} p_{s'} = 1$, so $p \in \Delta(\mathcal{S})$; dividing $A_s \boldsymbol{f}_s \le x_s b_s$ by x_s gives $A_s p \le b_s$, hence $p \in \mathcal{P}(s)$ and $\boldsymbol{f}_s = x_s p$. If $x_s = 0$, then $\boldsymbol{f}_s \ge 0$ and $\sum_{s'} f_{s,s'} = 0$ force $\boldsymbol{f}_s = 0 = x_s p$ for any $p \in \mathcal{P}(s)$, and $\mathcal{P}(s) \neq \emptyset$. □

In particular, $\mathsf{Split} \wedge \mathsf{NNeg} \wedge \mathsf{Adm}$ at a state $s \in \mathcal{S}^?$ holds exactly when the outgoing flow is $\boldsymbol{f}_s = x_s p$ for some admissible distribution $p \in \mathcal{P}(s)$.

C.4 Proof of Lemma 2

Proof of Lemma 2. Fix $s_0 \in \mathcal{S}^?$. We show that the optimum of $(\star)$ equals $\boldsymbol{v}^{\mathcal{R}}_{s_0}(\mathsf{Safe}(\mathcal{T}))$, via two inequalities.

The optimum is at most the value. Let $\tau^\star$ be an optimal positional policy; by Lemma 15 it has finite occupation. By (1) and Lemma 16, the occupation $(x^{\tau^\star}, f^{\tau^\star})$ satisfies $\mathsf{NNeg} \wedge \mathsf{Cons} \wedge \mathsf{Split} \wedge \mathsf{Adm}$, so it is feasible for $(\star)$. By (2) and Lemma 15, its objective is $\Pr^{\tau^\star}_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})] = \Pr^{\tau^\star}_{s_0}[\mathsf{Safe}(\mathcal{T})] = \boldsymbol{v}^{\mathcal{R}}_{s_0}(\mathsf{Safe}(\mathcal{T}))$. Since $(\star)$ minimises, its optimum is at most the value.

The value is at most the optimum. Let $(x^\star, f^\star)$ be any feasible solution of $(\star)$. Define a positional environment policy τ° by

$$\tau^\circ(s) = \begin{cases} \boldsymbol{f}^\star_s / x^\star_s & \text{if } s \in \mathcal{S}^? \text{ and } x^\star_s > 0, \\ p^\dagger \in \mathcal{P}(s) \text{ (arbitrary)} & \text{otherwise.} \end{cases}$$

By Lemma 16, $\tau^\circ(s) \in \mathcal{P}(s)$, so τ° is a valid positional policy, and $x^\star_s \, p^{\tau^\circ}_{s,s'} = f^\star_{s,s'}$ for all $s \in \mathcal{S}^?$ (both sides vanish when $x^\star_s = 0$, as then $\boldsymbol{f}^\star_s = 0$).

We show $x^{\tau^\circ}_s \le x^\star_s$ for every $s \in \mathcal{S}^?$ by proving $x^{\tau^\circ,N}_s \le x^\star_s$ for the truncated occupation $x^{\tau^\circ,N}_s = \sum_{t=0}^{N} \Pr^{\tau^\circ}_{s_0}[s_t = s]$, by induction on N. For $N = 0$, $x^{\tau^\circ,0}_s = \mathbf{1}[s = s_0] \le x^\star_s$ by Cons and $f^\star \ge 0$; and the inductive step uses $x^\star_{s'} p^{\tau^\circ}_{s',s} = f^\star_{s',s}$ and Cons,

$$x^{\tau^\circ,N}_s = \mathbf{1}[s = s_0] + \sum_{s' \in \mathcal{S}^?} x^{\tau^\circ,N-1}_{s'} \, p^{\tau^\circ}_{s',s} \le \mathbf{1}[s = s_0] + \sum_{s' \in \mathcal{S}^?} f^\star_{s',s} = x^\star_s.$$

Letting $N \to \infty$ gives $x^{\tau^\circ}_s \le x^\star_s < \infty$; in particular τ° has finite occupation, so it reaches $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ almost surely from s_0. Using (2), $x^{\tau^\circ}_s \le x^\star_s$, and $x^\star_s p^{\tau^\circ}_{s,w} = f^\star_{s,w}$,

$$\Pr^{\tau^\circ}_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})] = \sum_{\substack{s \in \mathcal{S}^? \\ w \in \mathcal{S}^{=1}}} x^{\tau^\circ}_s \, p^{\tau^\circ}_{s,w} \le \sum_{\substack{s \in \mathcal{S}^? \\ w \in \mathcal{S}^{=1}}} x^\star_s \, p^{\tau^\circ}_{s,w} = \sum_{\substack{s \in \mathcal{S}^? \\ w \in \mathcal{S}^{=1}}} f^\star_{s,w}. \tag{3}$$

It remains to bound the safety value by $\Pr^{\tau^\circ}_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})]$. Consider the policy $\widehat{\tau}$ that follows τ° until the play first reaches $\mathcal{S}^{=0} \cup \mathcal{S}^{=1}$ (which happens almost surely), then, from a state of $\mathcal{S}^{=1}$, plays to stay safe (possible, since safety value 1 means safe under every policy), and from a state of $\mathcal{S}^{=0}$, plays an unsafe-almost-surely continuation (possible, since safety value 0). Then a $\widehat{\tau}$-play is safe exactly when it is absorbed in $\mathcal{S}^{=1}$, so

$$\boldsymbol{v}^{\mathcal{R}}_{s_0}(\mathsf{Safe}(\mathcal{T})) \;\le\; \Pr^{\widehat{\tau}}_{s_0}[\mathsf{Safe}(\mathcal{T})] \;=\; \Pr^{\tau^\circ}_{s_0}[\mathsf{Reach}(\mathcal{S}^{=1})] \;\le\; \sum_{\substack{s \in \mathcal{S}^? \\ w \in \mathcal{S}^{=1}}} f^\star_{s,w},$$

using (3). As this holds for every feasible solution, the value is at most the optimum of $(\star)$.

Combining the two inequalities, the optimum of $(\star)$ equals $\boldsymbol{v}^{\mathcal{R}}_{s_0}(\mathsf{Safe}(\mathcal{T}))$. □

C.5 Proof of Theorem 3

Proof of Theorem 3. The sets $\mathcal{S}^{=1}, \mathcal{S}^{=0}$ are computable in polynomial time (Asadi et al. 2026b). The program $(\star)$ has $O(|\mathcal{S}|^2)$ variables and, by the Assumption, a number of constraints polynomial in the encoding of $\mathcal{R}$, and linear programs of polynomial size are solvable in polynomial time. States in $\mathcal{S}^{=1}$ and $\mathcal{S}^{=0}$ have known safety values 1 and 0; for the remaining states, solving $(\star)$ once per source $s_0 \in \mathcal{S}^?$ yields all values $\boldsymbol{v}_s^{\mathcal{R}}(\mathsf{Safe}(\mathcal{T}))$ and, corresponding optimal positional policies, all in polynomial time. □

D MecDecomp Procedure for Computing MEC Decompositions of RMCs

Throughout, fix an RMC $\mathcal{R} = (\mathcal{S}, \mathcal{P})$ with linearly defined uncertainty. We recall the notion of maximal end-component from Definition 4 and give the MecDecomp procedure that computes them; its correctness is established in Theorem 18.

Internal transitions. For a set $B \subseteq \mathcal{S}$ and a state $s \in B$, let

$$\mathcal{P}_B(s) \;=\; \{\, p \in \mathcal{P}(s) \;:\; \mathsf{supp}(p) \subseteq B \,\}$$

be the set of admissible distributions at s whose support stays inside B, i.e. the one-step choices with which the environment can keep the play within B. We say B is *closed* if $\mathcal{P}_B(s) \neq \emptyset$ for every $s \in B$; equivalently, from every state of B the environment has at least one distribution that does not leave B. The *internal transition graph* of B is the directed graph $G_B = (B, E_B)$ whose edges record the transitions the environment can take while remaining in B,

$$E_B \;=\; \{\, (s, s') \in B \times B \;:\; \exists p \in \mathcal{P}_B(s),\; p_{s'} > 0 \,\}.$$

This is exactly the notation of Definition 4: the end-components of $\mathcal{R}$ are the non-empty closed sets B whose internal transition graph G_B is strongly connected, and a *maximal end-component* (MEC) is an end-component that is not strictly contained in another.

Algorithm. Algorithm 4 computes the MEC decomposition of $\mathcal{R}$. Following the standard MDP procedure, it repeatedly *closes* a candidate set B by discarding every state from which the environment cannot avoid leaving B, and then *splits* the closed set into the strongly connected components of G_B, recursing on each proper component. Both operations reduce to polynomially many linear-programming queries against the description of $\mathcal{P}$.

Algorithm 4: MecDecomp(B): MEC decomposition of an RMC

Input: A subset $B \subseteq \mathcal{S}$ of an RMC $\mathcal{R}$ with linearly defined uncertainty; the top-level call uses $B = \mathcal{S}$.
Output: The set of MECs contained in B.

repeat
  Leak $\leftarrow \{\, s \in B \;:\; \mathcal{P}_B(s) = \emptyset \,\}$
  $B \leftarrow B \setminus$ Leak
until Leak $= \emptyset$
if $B = \emptyset$ **then**
  return $\emptyset$
end if
Compute the edge set E_B and the SCCs $C_1, \ldots, C_k$ of G_B
if $k = 1$ **then**
  return $\{B\}$ {B is a MEC}
end if
Result $\leftarrow \emptyset$
for $i = 1, \ldots, k$ **do**
  Result $\leftarrow$ Result $\cup$ MecDecomp(C_i)
end for
return Result

LP primitives. The two operations on $\mathcal{P}$ are realised as linear programs:

- *Closure test* $\mathcal{P}_B(s) \neq \emptyset$. Add the equalities $p_{s'} = 0$ for every $s' \notin B$ to the linear description $A_s\, p \leq b_s$ of $\mathcal{P}(s)$ and test feasibility over the simplex; the set $\mathcal{P}_B(s)$ is non-empty iff this program is feasible.
- *Edges of* G_B. For each ordered pair $(s, s') \in B \times B$, maximise $p_{s'}$ subject to $p \in \mathcal{P}_B(s)$; the edge (s, s') belongs to E_B iff the optimum is strictly positive.

The strongly connected components of G_B are then computed by Tarjan's algorithm in time linear in $|B| + |E_B|$.

Lemma 17. *The maximal end-components of $\mathcal{R}$ are pairwise disjoint.*

Proof. Let M_1, M_2 be MECs with a common state $s \in M_1 \cap M_2$; we show that $M_1 \cup M_2$ is an end-component, which, since two distinct MECs cannot contain one another, strictly contains M_1 and contradicts its maximality. For closure, take any $s' \in M_1 \cup M_2$, say $s' \in M_i$. As M_i is closed, $\mathcal{P}_{M_i}(s') \neq \emptyset$; and since $M_i \subseteq M_1 \cup M_2$, a distribution supported in M_i is also supported in $M_1 \cup M_2$, so $\mathcal{P}_{M_i}(s') \subseteq \mathcal{P}_{M_1 \cup M_2}(s')$ and the latter is non-empty. For strong connectivity of $G_{M_1 \cup M_2}$, note that relaxing the support constraint from $\subseteq M_i$ to $\subseteq M_1 \cup M_2$ only adds edges, so G_{M_i} is a subgraph of $G_{M_1 \cup M_2}$ restricted to M_i; hence each M_i is internally strongly connected in $G_{M_1 \cup M_2}$, and the shared state s links the two parts, making $M_1 \cup M_2$ strongly connected. Thus $M_1 \cup M_2$ is an end-component. □

Theorem 18. *Algorithm 4 returns the set of maximal end-components of $\mathcal{R}$, using $O(|\mathcal{S}|^3)$ linear-programming queries each of size polynomial in the encoding of $\mathcal{R}$, and hence runs in polynomial time. Moreover, a* single *invocation* $\mathsf{MecDecomp}(B)$*, excluding the work of its recursive calls, performs $O(|B|^2)$ linear-programming queries.*

Proof. We first argue correctness and then bound the number of LP queries. Both parts rely on the invariant that at every call $\mathsf{MecDecomp}(B)$, every MEC of $\mathcal{R}$ that intersects B is contained in B. The invariant holds for the top-level call with $B = \mathcal{S}$, since every MEC intersects $\mathcal{S}$ and is contained in it.

Closure preserves end-components. The closure loop removes a state s from B only when $\mathcal{P}_B(s) = \emptyset$. Let E be any end-component with $E \subseteq B$ and $s \in E$. Since E is closed, $\mathcal{P}_E(s) \neq \emptyset$, and $\mathcal{P}_E(s) \subseteq \mathcal{P}_B(s)$ because $E \subseteq B$; hence $\mathcal{P}_B(s) \neq \emptyset$ and s is not removed. Therefore every end-component contained in B survives the closure loop. Let $B^\star$ be the closed set obtained when the loop terminates.

Every end-component lies in a single SCC of $G_{B^\star}$. For an end-component $E \subseteq B^\star$, the support condition $\mathsf{supp}(p) \subseteq E$ is stronger than $\mathsf{supp}(p) \subseteq B^\star$, so G_E is a subgraph of $G_{B^\star}$ restricted to E. Hence E is strongly connected inside $G_{B^\star}$ and is contained in a single strongly connected component.

Base case: one SCC. Suppose $G_{B^\star}$ is strongly connected (a single SCC). Then $B^\star$ is non-empty, closed, and strongly connected, so it is an end-component. By Lemma 17 it lies in a unique MEC M of $\mathcal{R}$; since M intersects B, the invariant gives $M \subseteq B$, and M survives the closure loop, so $M \subseteq B^\star$. Combined with $B^\star \subseteq M$ (as $B^\star$ is an end-component contained in the MEC M), we get $B^\star = M$. Thus the algorithm correctly returns the single MEC $\{B^\star\}$.

Recursive case: several SCCs. If $G_{B^\star}$ has SCCs $C_1, \ldots, C_k$ with $k \geq 2$, then by the previous paragraph every end-component inside $B^\star$ lies in some C_i, and the algorithm recurses on each C_i. The invariant is maintained for each recursive call: any MEC M intersecting C_i also intersects B, so $M \subseteq B$ by the parent invariant, $M \subseteq B^\star$ by closure, and $M \subseteq C_i$ since it lies in a single SCC. By induction on the recursion, the calls collectively return exactly the MECs of $\mathcal{R}$ contained in B; for the top-level call $B = \mathcal{S}$ this is the full MEC decomposition.

Complexity. A single call, excluding its recursive calls, performs $O(|B|^2)$ LP queries: the closure loop runs at most $|B|$ rounds, each testing the $|B|$ feasibility programs $\mathcal{P}_B(s) \neq \emptyset$, and building G_B uses one maximisation LP per ordered pair in $B \times B$; the SCC computation is linear in $|B| + |E_B|$ and uses no LPs. This is the second claim of the statement.

We bound the total by counting the recursion level by level. There are at most $|\mathcal{S}|$ levels, because each child C_i is a *proper* subset of its parent: $C_i \subseteq B^\star \subseteq B$, and recursion happens only when $B^\star$ splits into $k \geq 2$ SCCs. The sets processed at any fixed level are pairwise disjoint, since the children of a node are the pairwise-disjoint SCCs of $B^\star$ and children of distinct nodes stay inside the disjoint sets of their parents. Hence each level costs at most $\sum_{B \text{ at that level}} O(|B|^2) \leq O\big(\big(\sum_B |B|\big)^2\big) = O(|\mathcal{S}|^2)$ LP queries, and the at most $|\mathcal{S}|$ levels cost $O(|\mathcal{S}|^3)$ in total. Each query is a linear program of size polynomial in the encoding of $\mathcal{R}$ and is solvable in polynomial time, so the whole algorithm runs in polynomial time. □

E Proofs of Lemmas 5 and 6

Throughout this section we fix a linearly defined RMC $\mathcal{R} = (\mathcal{S}, \mathcal{P})$, a coloring $C \colon \mathcal{S} \to [d]$, and an initial state s_0. Recall from Section 3.2 that a play π *violates* $\mathsf{Parity}(C)$ exactly when its dominant color $\max\{C(s) : s \in \mathsf{Inf}(\pi)\}$ is odd, and write

$$W_{\mathrm{odd}} = \mathsf{OddEC}(\mathcal{S})$$

for the output of Algorithm 1; the internal transition graph G_B and the restricted uncertainty sets $\mathcal{P}_B$ are those of Appendix D. The proofs rest on two properties of W_{odd}: the environment can confine a play to it while forcing an odd dominant color (Lemma 19), and every violating play must reach it (Lemma 20).

Lemma 19 (Confinement)**.** *Every end-component E of $\mathcal{R}$ whose dominant color $\max_{s \in E} C(s)$ is odd satisfies $E \subseteq W_{\mathrm{odd}}$. Moreover, there is a positional environment policy τ_{stay} under which W_{odd} is closed and, from every $s \in W_{\mathrm{odd}}$, the dominant color visited infinitely often is odd almost surely.*

Proof. First part. We show $E \subseteq W_{\mathrm{odd}}$ by induction on the recursion depth of the call $\mathsf{OddEC}(B)$ that first has $E \subseteq B$ (the top-level call has $B = \mathcal{S} \supseteq E$). Since E is an end-component contained in B and MECs are maximal, E is contained in some MEC M_i of B computed by $\mathsf{MecDecomp}(B)$. Let $m_i = \max_{s \in M_i} C(s)$. If m_i is odd, the algorithm adds M_i to its output, so $E \subseteq M_i \subseteq W_{\mathrm{odd}}$. If m_i is even, then, as $E \subseteq M_i$, its dominant color satisfies $\max_{s \in E} C(s) \leq m_i$; being odd it cannot equal the even m_i, so $\max_{s \in E} C(s) < m_i$ and no state of E has color m_i. Hence $E \cap U_i = \emptyset$ for $U_i = \{s \in M_i : C(s) = m_i\}$, i.e. $E \subseteq M_i \setminus U_i$, and the algorithm recurses on $M_i \setminus U_i$ at strictly greater depth; the inductive hypothesis gives $E \subseteq W_{\mathrm{odd}}$.

Second part. Each set added to the output is a MEC M_i (of some subset B) with odd dominant color m_i; such an M_i is an end-component of $\mathcal{R}$, since every $s \in M_i$ admits $p \in \mathcal{P}_{M_i}(s) \subseteq \mathcal{P}(s)$ with $\mathsf{supp}(p) \subseteq M_i$. Fix any such M_i. Its internal transition graph G_{M_i} is strongly connected, and for each $s \in M_i$ the convexity of $\mathcal{P}(s)$ lets us average the distributions witnessing the outgoing G_{M_i}-edges of s into a single $\tau_{\mathrm{stay}}(s) \in \mathcal{P}(s)$ with $\mathsf{supp}(\tau_{\mathrm{stay}}(s)) \subseteq M_i$ that covers all G_{M_i}-successors of s. Under τ_{stay} the set M_i is thus closed and strongly connected, hence a single BSCC, so a play started anywhere in M_i visits every state of M_i infinitely often almost surely; its dominant color is therefore m_i, which is odd. As W_{odd} is the disjoint union of these MECs, defining τ_{stay} componentwise makes W_{odd} closed with the stated property. □

Lemma 20 (Reaching W_{odd})**.** *For every positional environment policy τ,*

$$\Pr^{\tau}_{s_0}[\neg\,\mathsf{Parity}(C)] \;\le\; \Pr^{\tau}_{s_0}[\mathsf{Reach}(W_{\mathrm{odd}})].$$

Proof. Fix a positional τ. The Markov chain it induces from s_0 reaches a bottom strongly connected component (BSCC) almost surely. Any such BSCC C is closed under τ and strongly connected, and each $s \in C$ has $\tau(s) \in \mathcal{P}(s)$ with $\mathsf{supp}(\tau(s)) \subseteq C$, so C is an end-component of $\mathcal{R}$. On the event $\neg\,\mathsf{Parity}(C)$ the play visits exactly the states of its BSCC infinitely often, i.e. $\mathsf{Inf}(\pi) = C$, and its dominant color $\max_{s\in C} C(s)$ is odd; by Lemma 19 this forces $C \subseteq W_{\mathrm{odd}}$, so the play reaches W_{odd}. Hence $\neg\,\mathsf{Parity}(C) \subseteq \mathsf{Reach}(W_{\mathrm{odd}})$ up to a null set, giving the inequality. □

Proof of Lemma 5. By the first part of Lemma 19, the output $W_{\mathrm{odd}} = \mathsf{OddEC}(\mathcal{S})$ contains every end-component with an odd dominant color; conversely, by construction it is a union of MECs M_i with odd dominant color m_i, each an end-component of $\mathcal{R}$. Thus W_{odd} is exactly the set of states lying in an end-component whose dominant color is odd, and by the second part of the lemma the environment forces an odd dominant color almost surely from every state of W_{odd} via τ_{stay}, so each such state has parity value 0, as required.

For the complexity, we count the LP queries of the top-level call $\mathsf{OddEC}(\mathcal{S})$ level by level, over the two nested recursions of OddEC and MecDecomp *at once*. This is the right way to account for them: OddEC calls MecDecomp afresh at every node of its own recursion, so composing the two bounds as black boxes ($O(|B|^3)$ queries per MEC decomposition, times an OddEC recursion of depth $\Omega(|\mathcal{S}|)$) would give only $O(|\mathcal{S}|^4)$. But the cubic factor of Theorem 18 is not the cost of one MEC decomposition: a single invocation of $\mathsf{MecDecomp}(B)$ costs only $O(|B|^2)$ queries, the extra factor $|\mathcal{S}|$ being itself the number of levels of MecDecomp's recursion. Counting the levels of the two recursions together therefore adds these factors instead of multiplying them.

So consider the combined recursion, whose nodes are all the invocations of MecDecomp made during $\mathsf{OddEC}(\mathcal{S})$, each labelled by the set B it is invoked on. A node $\mathsf{MecDecomp}(B)$ has as children either the invocations $\mathsf{MecDecomp}(C_1), \ldots, \mathsf{MecDecomp}(C_k)$ on the SCCs of $B^\star$, or, if $\mathsf{MecDecomp}(B)$ returns a single MEC $M = B^\star$ whose dominant colour m is even, the single invocation $\mathsf{MecDecomp}(M \setminus C^{-1}(m))$ made by the recursive call of line 6 of Algorithm 1. This covers every invocation of MecDecomp, since each MEC M_i processed by OddEC is returned by exactly one node, the one with $B^\star = M_i$.

There are at most $|\mathcal{S}|$ levels, because a child's set is a proper subset of its parent's: $C_i \subsetneq B^\star \subseteq B$ as $B^\star$ splits into $k \ge 2$ SCCs, and $M \setminus C^{-1}(m) \subsetneq M \subseteq B$ as the maximal-colour states of M are removed before recursing. The sets at any fixed level are pairwise disjoint, because siblings are disjoint — they are distinct SCCs of $B^\star$ in the first case, and there is only one child in the second — and children of distinct nodes stay inside the disjoint sets of their parents. Hence each level costs at most $O(|\mathcal{S}|^2)$ LP queries by the per-invocation bound of Theorem 18, and $O(|\mathcal{S}|^3)$ queries suffice in total. The remaining work of OddEC — computing each m_i, testing its parity, and forming $M_i \setminus C^{-1}(m_i)$ and the union W — is $O(|B|)$ per node, hence $O(|\mathcal{S}|)$ per level and $O(|\mathcal{S}|^2)$ in total. Each LP query has size polynomial in the encoding of $\mathcal{R}$, so W_{odd} is computed in polynomial time. □

Proof of Lemma 6. Write $R = \sup_\tau \Pr^{\tau}_{s_0}[\mathsf{Reach}(W_{\mathrm{odd}})]$ for the maximal probability of reaching W_{odd}. Since the environment minimizes the probability of $\mathsf{Parity}(C)$,

$$\boldsymbol{v}^{\mathcal{R}}_{s_0}(\mathsf{Parity}(C)) = \inf_\tau \Pr^{\tau}_{s_0}[\mathsf{Parity}(C)] = 1 - \sup_\tau \Pr^{\tau}_{s_0}[\neg\,\mathsf{Parity}(C)],$$

and this value is attained by a positional environment policy, so the last supremum may be restricted to positional policies. Indeed, the parity problem in a linearly defined RMC reduces to a parity problem in an MDP controlled by the environment: at each state s let the actions be the (finitely many) vertices of the polytope $\mathcal{P}(s)$, each with the transition distribution of the corresponding corner. Every environment policy of the RMC is a policy of this MDP whose one-step choice $p \in \mathcal{P}(s)$ is a convex combination of vertex actions, and conversely; the two induce the same distributions over plays, so the parity values coincide. Since parity MDPs attain their optimal values by positional policies mapping each state to a single action, the environment has a positional optimal policy that selects one vertex of $\mathcal{P}(s)$ at each state. We show the resulting supremum equals R.

Upper bound. For every positional τ, Lemma 20 gives $\Pr^{\tau}_{s_0}[\neg\,\mathsf{Parity}(C)] \le \Pr^{\tau}_{s_0}[\mathsf{Reach}(W_{\mathrm{odd}})] \le R$.

Lower bound. Applying Theorem 3 to the safe set $\mathcal{S} \setminus W_{\mathrm{odd}}$ (equivalently, to the unsafe set W_{odd}) yields a positional environment policy τ_{reach} attaining $\Pr^{\tau_{\mathrm{reach}}}_{s_0}[\mathsf{Reach}(W_{\mathrm{odd}})] = R$. Define the positional policy

$$\tau^\star(s) = \begin{cases} \tau_{\mathrm{stay}}(s) & s \in W_{\mathrm{odd}}, \\ \tau_{\mathrm{reach}}(s) & s \notin W_{\mathrm{odd}}, \end{cases}$$

with τ_{stay} from Lemma 19. Since $\tau^\star$ agrees with τ_{reach} until W_{odd} is first hit, it reaches W_{odd} with probability R; and once inside, W_{odd} is closed under τ_{stay} and the dominant color is odd almost surely, so $\mathsf{Reach}(W_{\mathrm{odd}})$ implies $\neg\,\mathsf{Parity}(C)$ under $\tau^\star$. Hence $\Pr^{\tau^\star}_{s_0}[\neg\,\mathsf{Parity}(C)] \geq \Pr^{\tau^\star}_{s_0}[\mathsf{Reach}(W_{\mathrm{odd}})] = R$.

Combining the two bounds, $\sup_\tau \Pr^{\tau}_{s_0}[\neg\,\mathsf{Parity}(C)] = R$, so, using the safety–reachability duality with unsafe set $U = W_{\mathrm{odd}}$,

$$\boldsymbol{v}^{\mathcal{R}}_{s_0}(\mathsf{Parity}(C)) = 1 - R = \boldsymbol{v}^{\mathcal{R}}_{s_0}\big(\mathsf{Safe}(\mathcal{S} \setminus W_{\mathrm{odd}})\big). \qquad \square$$

Proof of Theorem 7. By Lemma 5, W_{odd} is computable in polynomial time. By Lemma 6 the parity value at every state equals the safety value of $\mathsf{Safe}(\mathcal{S} \setminus W_{\mathrm{odd}})$, which, together with an optimal positional environment policy, is computed in polynomial time by Theorem 3. The policy $\tau^\star$ built in the proof of Lemma 6 (reaching W_{odd} optimally and then applying τ_{stay}) is a corresponding optimal positional environment policy for $\mathsf{Parity}(C)$. $\square$

F Proof of Proposition 8 and Corollary 9

This appendix proves Proposition 8: every linearly defined RMDP satisfies the positional-determinacy condition of Proposition 8. The argument reduces a linearly defined RMDP $\mathcal{M}$ to a finite turn-based stochastic parity game $\mathcal{G}$ (the *vertex game* of $\mathcal{M}$) in which the environment's choice of a distribution from an uncertainty polytope is replaced by an adversarial choice of a *vertex* (corner) of that polytope followed by a probabilistic step. The reduction is value-preserving and carries positional strategies of $\mathcal{G}$ back to positional policies of $\mathcal{M}$; since turn-based stochastic parity games are positionally determined, so is $\mathcal{M}$. Specialising to the single-action case yields the reduction of a linearly defined RMC to an MDP used in Sections 3 and 3.2. Throughout we fix a linearly defined RMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ and a coloring $C\colon \mathcal{S} \to [d]$, and write $\varphi = \mathsf{Parity}(C)$.

F.1 Vertices of the Uncertainty Polytopes

For a state-action pair (s, a) the uncertainty set $\mathcal{P}(s, a) = \{p \in \Delta(\mathcal{S}) : A_{s,a}\, p \leq b_{s,a}\}$ is a bounded polytope (it is contained in the simplex $\Delta(\mathcal{S})$). Let $\mathsf{Vert}(\mathcal{P}(s, a))$ denote its finite set of vertices (extreme points). The Minkowski–Weyl theorem for polytopes gives the following standard fact, on which the whole reduction rests.

Lemma 21 (Vertex representation)**.** *Each polytope $\mathcal{P}(s, a)$ has finitely many vertices, and*

$$\mathcal{P}(s, a) \;=\; \Big\{ \textstyle\sum_{q \in \mathsf{Vert}(\mathcal{P}(s,a))} \lambda_q\, q \;:\; \lambda_q \geq 0,\ \sum_q \lambda_q = 1 \Big\},$$

i.e. every admissible distribution $p \in \mathcal{P}(s, a)$ is a convex combination of vertices of $\mathcal{P}(s, a)$.

Since a convex combination of distributions is again a distribution, each vertex $q \in \mathsf{Vert}(\mathcal{P}(s, a))$ is itself a distribution over $\mathcal{S}$, and it is the extreme-point distribution the environment may commit to. Lemma 21 is the only place where polytopicity is used: it is what makes the environment's continuous choice reducible to a *finite* one.

F.2 The Vertex Game

We build a turn-based stochastic parity game $\mathcal{G} = (V, V_\Diamond, V_\Box, V_\bigcirc, E, C_\mathcal{G})$ with three kinds of vertices:

- *Agent vertices* $V_\Diamond = \{\, v_s : s \in \mathcal{S} \,\}$, one per state, owned by the agent (the maximiser);
- *Environment vertices* $V_\Box = \{\, v_{s,a} : s \in \mathcal{S},\ a \in \mathcal{A} \,\}$, one per state-action pair, owned by the environment (the minimiser);
- *Probabilistic vertices* $V_\bigcirc = \{\, v_{s,a,q} : s \in \mathcal{S},\ a \in \mathcal{A},\ q \in \mathsf{Vert}(\mathcal{P}(s, a)) \,\}$, one per chosen vertex of an uncertainty polytope.

The moves are as follows. From an agent vertex v_s the agent chooses an action $a \in \mathcal{A}$ and moves to the environment vertex $v_{s,a}$. From an environment vertex $v_{s,a}$ the environment chooses a vertex $q \in \mathsf{Vert}(\mathcal{P}(s, a))$ and moves to the probabilistic vertex $v_{s,a,q}$. From a probabilistic vertex $v_{s,a,q}$ the next vertex is sampled: the game moves to the agent vertex $v_{s'}$ with probability $q(s')$ for each $s' \in \mathcal{S}$. Thus

$$E \;=\; \{\, (v_s, v_{s,a}) : a \in \mathcal{A} \,\} \cup \{\, (v_{s,a}, v_{s,a,q}) : q \in \mathsf{Vert}(\mathcal{P}(s, a)) \,\} \cup \{\, (v_{s,a,q}, v_{s'}) : q(s') > 0 \,\}.$$

The coloring $C_\mathcal{G}$ assigns to each agent vertex the color of its state, $C_\mathcal{G}(v_s) = C(s)$, and to every environment and probabilistic vertex the least color 0, which is even and dominated by every color of $[d]$:

$$C_\mathcal{G}(v_s) = C(s), \qquad C_\mathcal{G}(v_{s,a}) = C_\mathcal{G}(v_{s,a,q}) = 0.$$

A play of $\mathcal{G}$ satisfies the parity objective iff the largest color occurring infinitely often is even. The game is finite: it has $|\mathcal{S}| + |\mathcal{S}||\mathcal{A}| + \sum_{s,a} |\mathsf{Vert}(\mathcal{P}(s, a))|$ vertices.

F.3 Correspondence of Plays and Strategies

Every play of $\mathcal{G}$ from v_{s_0} has the form

$$v_{s_0}\, v_{s_0,a_0}\, v_{s_0,a_0,q_0}\, v_{s_1}\, v_{s_1,a_1}\, v_{s_1,a_1,q_1}\, v_{s_2} \cdots,$$

alternating one agent vertex, one environment vertex and one probabilistic vertex per round. Deleting the environment and probabilistic vertices yields the *projected path* $\rho(\pi) = s_0 s_1 s_2 \cdots \in \mathcal{S}^\omega$, an infinite path of $\mathcal{M}$. The projection ρ is a colour-faithful correspondence.

Lemma 22 (Play projection)**.** *For every play π of $\mathcal{G}$, the maximal color occurring infinitely often in π equals the maximal color occurring infinitely often in $\rho(\pi)$. Consequently π satisfies* $\mathsf{Parity}(C_{\mathcal{G}})$ *iff $\rho(\pi)$ satisfies* $\mathsf{Parity}(C)$.

Proof. Each round of π contributes exactly one agent vertex v_{s_i} with color $C(s_i)$ and two auxiliary vertices with color 0. Hence the multiset of colors seen infinitely often in π is that of $\rho(\pi)$ together with the color 0 (seen infinitely often, as the play is infinite). Since 0 is the least color, it never changes the maximum, so the maximal recurring color of π equals that of $\rho(\pi)$. The parity condition depends only on this maximum. □

We now match strategies. A (behavioural) agent strategy in $\mathcal{G}$ chooses, given the history ending at an agent vertex v_s, a distribution over actions; a positional (pure memoryless) agent strategy is a map $\mathcal{S} \to \mathcal{A}$. These are literally the agent policies of $\mathcal{M}$, with positional strategies corresponding to positional agent policies. For the environment, at an environment vertex $v_{s,a}$ a behavioural strategy chooses a distribution over the vertices $\mathsf{Vert}(\mathcal{P}(s,a))$; the induced one-step distribution over successor states s' is $\sum_q \lambda_q\, q(s')$, i.e. the convex combination $\sum_q \lambda_q\, q \in \mathcal{P}(s,a)$ by Lemma 21. The next lemma records that these convex combinations range over exactly $\mathcal{P}(s,a)$, so environment strategies of $\mathcal{G}$ and environment policies of $\mathcal{M}$ realise precisely the same one-step distributions.

Lemma 23 (Environment moves)**.** *Fix (s,a). The set of one-step distributions over $\mathcal{S}$ that the environment can realise at $v_{s,a}$ in $\mathcal{G}$, namely $\{\sum_q \lambda_q\, q : \lambda \in \Delta(\mathsf{Vert}(\mathcal{P}(s,a)))\}$, equals $\mathcal{P}(s,a)$. In particular, a* pure *choice of a single vertex q realises the distribution q, and every distribution $p \in \mathcal{P}(s,a)$ is realised by some (possibly randomised) choice.*

Proof. Every convex combination of vertices lies in $\mathcal{P}(s,a)$ because the polytope is convex; conversely, by Lemma 21 every $p \in \mathcal{P}(s,a)$ is such a convex combination. Committing to a single vertex q (i.e. $\lambda_q = 1$) realises q itself. □

Given a strategy profile in $\mathcal{G}$ and its counterpart in $\mathcal{M}$ realising the same one-step distributions at every vertex, the two induce, via the cylinder construction, the same probability distribution over projected paths; combined with Lemma 22 this yields value preservation.

Lemma 24 (Value preservation)**.** *For every $s \in \mathcal{S}$, the value of $\mathcal{G}$ at v_s for the objective* $\mathsf{Parity}(C_{\mathcal{G}})$ *equals $\boldsymbol{v}_s^{\mathcal{M}}(\varphi)$. Moreover, for a fixed positional agent policy σ (resp. positional environment policy τ) the value of the corresponding one-player game equals $\inf_{\tau'} \Pr_s^{\sigma,\tau'}[\varphi]$ (resp. $\sup_{\sigma'} \Pr_s^{\sigma',\tau}[\varphi]$).*

Proof. Fix any agent strategy and environment strategy in $\mathcal{G}$. By Lemma 23 there are an agent policy and an environment policy of $\mathcal{M}$ that at every state (resp. state-action pair) reproduce the same one-step distribution over successors, and conversely every pair of policies of $\mathcal{M}$ arises this way. Because the one-step distributions agree at every vertex, the two induced measures assign equal probability to every cylinder of projected paths, so $\Pr$ of the event $\{\pi : \rho(\pi) \models \mathsf{Parity}(C)\}$ in $\mathcal{G}$ equals $\Pr_s^{\sigma,\tau}[\varphi]$ in $\mathcal{M}$; by Lemma 22 the former equals the probability of $\mathsf{Parity}(C_{\mathcal{G}})$. Taking $\sup$ over agent strategies and $\inf$ over environment strategies on both sides gives $\mathrm{val}_{\mathcal{G}}(v_s) = \sup_\sigma \inf_\tau \Pr_s^{\sigma,\tau}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}}(\varphi)$. The one-player statements follow by fixing the corresponding player's strategy to the positional policy and repeating the argument. □

F.4 Proof of Proposition 8

Proof of Proposition 8. Turn-based stochastic games with a parity objective are positionally (pure-memoryless) determined: both players have optimal pure memoryless strategies, and the game has a value satisfying $\sup_\sigma \inf_\tau = \inf_\tau \sup_\sigma$ (Chatterjee and Henzinger 2012). Apply this to the vertex game $\mathcal{G}$ of $\mathcal{M}$. Let $\hat\sigma$ and $\hat\tau$ be optimal pure memoryless strategies of the agent and the environment in $\mathcal{G}$.

Map them back to $\mathcal{M}$. The strategy $\hat\sigma$ selects, at each agent vertex v_s, a single action $\sigma^*(s) \in \mathcal{A}$; this is a positional agent policy of $\mathcal{M}$. The strategy $\hat\tau$ selects, at each environment vertex $v_{s,a}$, a single vertex $q_{s,a} \in \mathsf{Vert}(\mathcal{P}(s,a))$; setting $\tau^*(s,a) = q_{s,a} \in \mathcal{P}(s,a)$ (a vertex distribution) gives a positional environment policy of $\mathcal{M}$. By Lemma 23 these choices realise exactly the same one-step distributions in $\mathcal{M}$ as $\hat\sigma, \hat\tau$ do in $\mathcal{G}$, so by Lemma 24 they are value preserving.

Optimality of $\hat\sigma$ in $\mathcal{G}$ means that against every environment strategy it secures at least the game value, which by Lemma 24 is $\boldsymbol{v}_s^{\mathcal{M}}(\varphi)$; translating through the strategy correspondence, $\inf_\tau \Pr_s^{\sigma^*,\tau}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}}(\varphi)$ for all s. Symmetrically, optimality of $\hat\tau$ gives $\sup_\sigma \Pr_s^{\sigma,\tau^*}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}}(\varphi)$ for all s. Hence

$$\inf_\tau \Pr_s^{\sigma^*,\tau}[\varphi] \;=\; \boldsymbol{v}_s^{\mathcal{M}}(\varphi) \;=\; \sup_\sigma \Pr_s^{\sigma,\tau^*}[\varphi] \qquad \text{for all } s \in \mathcal{S},$$

with σ^* and τ^* positional, which is exactly the positional-determinacy condition. In particular the two orders of optimization coincide, so $\mathcal{M}$ is determined. This proves Proposition 8. □

F.5 Proof of Corollary 9

Throughout this subsection, RMDP-Parity denotes the decision problem of Corollary 9: given a linearly defined RMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$, a coloring $C\colon \mathcal{S} \to [d]$, a state $s \in \mathcal{S}$ and a rational threshold $\lambda \in [0,1]$ written in binary, decide whether $\boldsymbol{v}_s^{\mathcal{M}}(\mathsf{Parity}(C)) \geq \lambda$. Analogously, SG-Parity denotes the problem of deciding, for a finite turn-based stochastic parity game $\mathcal{H}$ with rational transition probabilities, a vertex v of $\mathcal{H}$ and a rational $\lambda \in [0,1]$, whether $\mathrm{val}_{\mathcal{H}}(v) \geq \lambda$. We write $\|\mathcal{M}\|$ for the bit-size of the encoding of $\mathcal{M}$, and recall from the preliminaries that each uncertainty set is presented as

$$\mathcal{P}(s,a) = \Big\{ x \in \Delta(\mathcal{S}) \;\Big|\; \exists y \in \mathbb{R}^k \colon A_{s,a} \begin{bmatrix} x \\ y \end{bmatrix} \leq b_{s,a} \Big\},$$

where we assume without loss of generality that the simplex constraints defining $\Delta(\mathcal{S})$ are among the rows of $A_{s,a}\,[x;y] \leq b_{s,a}$. The two assertions of Corollary 9 are proved separately in Lemma 26 and Lemma 27 below.

Membership in $NP \cap coNP$. The certificates we use are positional policies. For the agent a positional policy is a map $\mathcal{S} \to \mathcal{A}$ and is trivially of size $O(|\mathcal{S}| \log |\mathcal{A}|)$. For the environment a positional policy must name a distribution inside each uncertainty set, so we first record that the distributions selected by the optimal environment policy constructed in the proof of Proposition 8 (vertices of the uncertainty polytopes) admit a compact encoding, even though the number of such vertices may be exponential (Remark 29).

Lemma 25 (Vertices admit compact encodings)**.** *Let $\mathcal{P}(s,a)$ be a linearly defined uncertainty set. Then every $q \in \mathsf{Vert}(\mathcal{P}(s,a))$ is a rational vector of bit-size polynomial in $\|\mathcal{M}\|$. Moreover, given a rational vector $q \in \mathbb{Q}^{\mathcal{S}}$, one can decide in time polynomial in $\|\mathcal{M}\|$ and the bit-size of q whether $q \in \mathcal{P}(s,a)$.*

Proof. Write $A_{s,a} = [A^x \mid A^y]$ according to the split of the variables into x and y, and let $Q = \{(x,y) : A^x x + A^y y \leq b_{s,a}\}$, so that $\mathcal{P}(s,a)$ is the projection of Q onto the x-coordinates. By the Fourier–Motzkin projection lemma,

$$\mathcal{P}(s,a) = \{\, x \in \mathbb{R}^{\mathcal{S}} \;:\; (\mu^\top A^x)\, x \leq \mu^\top b_{s,a} \;\text{ for all } \mu \in K \,\},$$

where $K = \{\mu \geq 0 : \mu^\top A^y = 0\}$ is the *projection cone* of Q, and it suffices to range over the finitely many extreme rays of K. Each extreme ray of K is a basic solution of a rational system whose size is bounded by $\|\mathcal{M}\|$, hence, by Cramer's rule, is a rational vector of bit-size polynomial in $\|\mathcal{M}\|$; consequently each inequality $(\mu^\top A^x)x \leq \mu^\top b_{s,a}$ in the above description has bit-size polynomial in $\|\mathcal{M}\|$, although there may be exponentially many of them. Now let q be a vertex of $\mathcal{P}(s,a)$. Since $\mathcal{P}(s,a) \subseteq \Delta(\mathcal{S})$ is a polytope of dimension at most $|\mathcal{S}|$, the vertex q is the unique solution of a subsystem of $|\mathcal{S}|$ linearly independent inequalities of the above description taken with equality. Applying Cramer's rule once more to this subsystem, whose entries are of bit-size polynomial in $\|\mathcal{M}\|$, shows that q is rational of bit-size polynomial in $\|\mathcal{M}\|$.

For the second claim, $q \in \mathcal{P}(s,a)$ holds iff the linear system $A^y y \leq b_{s,a} - A^x q$ in the unknown y is feasible, which is an LP feasibility test on rational data of size polynomial in $\|\mathcal{M}\|$ and the size of q, and is therefore decidable in polynomial time. □

Lemma 26. *RMDP-Parity $\in NP \cap coNP$.*

Proof. We show that both RMDP-Parity and its complement are in NP.

Membership in NP. The certificate is a positional agent policy $\sigma\colon \mathcal{S} \to \mathcal{A}$, of size polynomial in $\|\mathcal{M}\|$. The verifier constructs the induced RMC $\mathcal{M}^\sigma = (\mathcal{S}, \mathcal{P}^\sigma)$ with $\mathcal{P}^\sigma(s) = \mathcal{P}(s,\sigma(s))$; this is again linearly defined, is obtained from $\mathcal{M}$ by discarding the uncertainty sets of the unselected actions, and hence is computable in polynomial time. By Theorem 7 the verifier then computes $\boldsymbol{v}_s^\sigma = \inf_\tau \Pr_s^{\sigma,\tau}[\varphi]$ in time polynomial in $\|\mathcal{M}\|$; in particular $\boldsymbol{v}_s^\sigma$ is a rational of bit-size polynomial in $\|\mathcal{M}\|$, being the output of a polynomial-time computation, so the verifier can compare it with λ exactly in polynomial time. It accepts iff $\boldsymbol{v}_s^\sigma \geq \lambda$.

Soundness: for every positional agent policy σ we have $\boldsymbol{v}_s^\sigma \leq \sup_{\sigma'} \inf_\tau \Pr_s^{\sigma',\tau}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}}(\varphi)$, so an accepting certificate witnesses $\boldsymbol{v}_s^{\mathcal{M}}(\varphi) \geq \lambda$. Completeness: if $\boldsymbol{v}_s^{\mathcal{M}}(\varphi) \geq \lambda$, then the positional policy σ^* of Proposition 8 satisfies $\boldsymbol{v}_s^{\sigma^*} = \boldsymbol{v}_s^{\mathcal{M}}(\varphi) \geq \lambda$ and is accepted.

Membership in $coNP$. We give an NP procedure for the complement, i.e. for deciding $\boldsymbol{v}_s^{\mathcal{M}}(\varphi) < \lambda$. The certificate is a positional environment policy τ, presented as a table of rational distributions $\tau(s,a) \in \mathcal{P}(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$; the verifier first checks $\tau(s,a) \in \mathcal{P}(s,a)$ for every pair, which by Lemma 25 takes polynomial time, and rejects the certificate otherwise. Fixing τ leaves the MDP $\mathcal{M}^\tau = (\mathcal{S}, \mathcal{A}, \delta)$ with $\delta(s,a) = \tau(s,a)$, whose transition probabilities are exactly the entries of the certificate and whose size is therefore polynomial in the size of the certificate. Since agent policies of $\mathcal{M}$ played against the fixed τ are precisely the policies of $\mathcal{M}^\tau$, we have $\sup_\sigma \Pr_s^{\sigma,\tau}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}^\tau}(\mathsf{Parity}(C))$. The quantitative parity problem on MDPs is solvable in polynomial time (Chatterjee and Henzinger 2012): one computes the maximal end-component decomposition, retains those end-components in which the agent can enforce an even dominant colour, and solves a maximal reachability LP

for their union; in particular the value is a rational of bit-size polynomial in the size of $\mathcal{M}^\tau$. The verifier computes this value and accepts iff it is smaller than λ.

Soundness: for every positional environment policy τ we have $\sup_\sigma \Pr_s^{\sigma,\tau}[\varphi] \geq \sup_\sigma \inf_{\tau'} \Pr_s^{\sigma,\tau'}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}}(\varphi)$, so an accepting certificate witnesses $\boldsymbol{v}_s^{\mathcal{M}}(\varphi) < \lambda$. Completeness: if $\boldsymbol{v}_s^{\mathcal{M}}(\varphi) < \lambda$, take the positional environment policy τ^* constructed in the proof of Proposition 8. It satisfies $\sup_\sigma \Pr_s^{\sigma,\tau^*}[\varphi] = \boldsymbol{v}_s^{\mathcal{M}}(\varphi) < \lambda$ and, crucially, it selects at each pair (s,a) a *vertex* $q_{s,a} \in \mathsf{Vert}(\mathcal{P}(s,a))$; by Lemma 25 each $q_{s,a}$ is rational of bit-size polynomial in $\|\mathcal{M}\|$, so τ^* is a certificate of polynomial size and is accepted. □

We stress that neither direction materialises the vertex game $\mathcal{G}$: the NP verifier runs the polynomial-time LP-based RMC solver of Section 3 on the compact facet description, and the $coNP$ verifier only ever writes down one vertex per state-action pair rather than all of them.

Hardness. For the lower bound we exhibit the converse of the vertex-game construction: stochastic games are a special case of linearly defined RMDPs, in which the environment's uncertainty set is the convex hull of the Dirac distributions on the available successors.

Lemma 27. *SG-Parity reduces to RMDP-Parity in polynomial time. Hence RMDP-Parity is at least as hard as SG-Parity.*

Proof. Let $\mathcal{H} = (V, V_\Diamond, V_\Box, V_\bigcirc, E, C_\mathcal{H})$ be a turn-based stochastic parity game with a rational transition distribution $p_v \in \Delta(V)$ at every $v \in V_\bigcirc$, together with a vertex v_0 and a rational λ. We build a linearly defined RMDP $\mathcal{M}_\mathcal{H} = (V, \mathcal{A}, \mathcal{P})$ with $\mathcal{A} = \{a_1, \dots, a_m\}$, where m is the maximum out-degree in $\mathcal{H}$, and coloring $C = C_\mathcal{H}$, as follows. Fix for each vertex v an enumeration $v^1, \dots, v^{\deg(v)}$ of its successors.

- For $v \in V_\Diamond$ (an agent vertex) and $i \leq \deg(v)$, put $\mathcal{P}(v, a_i) = \{\delta_{v^i}\}$, the singleton containing the Dirac distribution on the i-th successor.
- For $v \in V_\Box$ (an environment vertex), put $\mathcal{P}(v, a_i) = \mathrm{conv}\{\delta_{v^1}, \dots, \delta_{v^{\deg(v)}}\} = \{p \in \Delta(V) : p(u) = 0 \text{ for all } u \text{ with } (v,u) \notin E\}$ for every i.
- For $v \in V_\bigcirc$ (a probabilistic vertex), put $\mathcal{P}(v, a_i) = \{p_v\}$ for every i.

For $v \in V_\Diamond$ and $i > \deg(v)$ we set $\mathcal{P}(v, a_i) = \mathcal{P}(v, a_1)$, so that the action set is uniform and no action is spurious. Every uncertainty set above is cut out by linear equalities and the simplex constraints, hence is linearly defined without auxiliary variables, and $\mathcal{M}_\mathcal{H}$ is computable from $\mathcal{H}$ in polynomial time.

It remains to show $\boldsymbol{v}_v^{\mathcal{M}_\mathcal{H}}(\mathsf{Parity}(C)) = \mathrm{val}_\mathcal{H}(v)$ for every $v \in V$, from which the reduction follows by asking RMDP-Parity about $\mathcal{M}_\mathcal{H}$, v_0 and λ. Consider the vertex game $\mathcal{G}$ of $\mathcal{M}_\mathcal{H}$ constructed above. For $v \in V_\Box$ the distributions $\delta_{v^1}, \dots, \delta_{v^{\deg(v)}}$ are distinct unit vectors and therefore affinely independent, so each of them is an extreme point of their convex hull and no other point is: $\mathsf{Vert}(\mathcal{P}(v, a_i)) = \{\delta_{v^1}, \dots, \delta_{v^{\deg(v)}}\}$. For $v \in V_\Diamond \cup V_\bigcirc$ the uncertainty sets are singletons and thus have a single vertex. Hence in $\mathcal{G}$: at $v \in V_\Diamond$ the agent chooses one of the successors of v and the play proceeds deterministically to it; at $v \in V_\Box$ the agent has no meaningful choice, the environment chooses a vertex δ_{v^i}, and the play proceeds deterministically to v^i; at $v \in V_\bigcirc$ neither player has a choice and the next vertex is sampled from p_v. Thus $\mathcal{G}$ is exactly $\mathcal{H}$ with every edge subdivided by at most two auxiliary vertices of colour 0, and with a dummy choice inserted at the vertices of $V_\Box \cup V_\bigcirc$. Since 0 is dominated by every colour of $[d]$, the argument of Lemma 22 applies verbatim: the maximal colour occurring infinitely often along a play of $\mathcal{G}$ equals that of the corresponding play of $\mathcal{H}$, and the strategy correspondence of Lemma 24 maps strategies of the two games to one another while preserving the induced distributions over projected paths. Consequently $\mathrm{val}_\mathcal{G}(v) = \mathrm{val}_\mathcal{H}(v)$, and by Lemma 24 applied to $\mathcal{M}_\mathcal{H}$ we get $\boldsymbol{v}_v^{\mathcal{M}_\mathcal{H}}(\mathsf{Parity}(C)) = \mathrm{val}_\mathcal{G}(v) = \mathrm{val}_\mathcal{H}(v)$, as required. □

Lemma 26 and Lemma 27 together prove Corollary 9. Two consequences are worth recording. First, since no polynomial-time algorithm for SG-Parity is known (the best known bounds are randomised sub-exponential (Chatterjee and Henzinger 2006a)) a polynomial-time algorithm for the quantitative parity problem on linearly defined RMDPs would resolve a long-standing open problem; this is the sense in which the exponential iteration bound of Theorem 12 is not an artefact of our analysis. Second, membership in $NP \cap coNP$ implies that RMDP-Parity is not NP-hard unless $NP = coNP$, so the source of hardness is not NP-hardness but the same "$NP \cap coNP$ barrier" as for stochastic games.

Remark 28 (The RMC specialisation)**.** *When $|\mathcal{A}| = 1$ the agent has no choice and the agent vertices merge with the (unique) environment vertices, so $\mathcal{G}$ becomes a* one-player *stochastic game controlled by the environment, i.e. an MDP whose actions at state s are the vertices $\mathsf{Vert}(\mathcal{P}(s))$ and whose transition under vertex q is q itself. Every environment policy of the RMC is a policy of this MDP whose one-step choice is a convex combination of vertex actions, and conversely, so the two induce the same distributions over plays and the parity values coincide (Lemma 24). Since parity MDPs admit optimal positional policies, the environment has a positional optimal policy selecting one vertex of $\mathcal{P}(s)$ at each state. This is precisely the reduction invoked in the proof of Lemma 6 and in Section 3.2.*

Remark 29 (Size of the reduction)**.** *The vertex game $\mathcal{G}$ is value-equivalent to $\mathcal{M}$ and could in principle be solved directly by a stochastic-game (resp. MDP) parity solver, computing $\boldsymbol{v}^{\mathcal{M}}(\varphi)$ exactly. However its size is governed by $\sum_{s,a} |\mathsf{Vert}(\mathcal{P}(s,a))|$, and a polytope described by polynomially many facets in dimension $|\mathcal{S}|$ can have $|\mathsf{Vert}(\mathcal{P}(s,a))|$ exponential in $|\mathcal{S}|$ (for instance the L_1 ball, whose vertex count grows with $|\mathcal{S}|$). Thus $\mathcal{G}$ may be exponentially larger than $\mathcal{M}$, and even writing it down need*

not be possible in polynomial space. This is exactly the inefficiency that motivates the polynomial time and space, LP-based algorithms of Section 3 and Section 4, which operate on the compact facet description of the polytopes rather than on their vertices.

G Proofs of Section 4

This appendix provides the proofs of the results of Section 4. Throughout we use the notation and standing objects of that section: the linearly defined RMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$, the coloring C and objective $\varphi = \mathsf{Parity}(C)$, the induced RMC $\mathcal{M}^\sigma$ and its value vector $\boldsymbol{v}^\sigma$, the value classes $\mathsf{VC}^\sigma(r)$, the one-step optima $\nu(s,a) = \min_{p \in \mathcal{P}(s,a)} p \cdot \boldsymbol{v}^\sigma$, tight actions $\mathcal{A}^{\boldsymbol{v}^\sigma}(s)$, tight faces $\mathcal{P}^{\boldsymbol{v}^\sigma}(s,a)$, the improvable set I, and the trap value-class RMDPs $\mathcal{M}_r$ over $U_r \cup \{\bot\}$ (where the agent is restricted to tight actions, the environment to tight faces, and all mass leaving U_r is collapsed onto the losing sink $\bot$ by the linear map κ_r) together with their colorings C_r, almost-sure regions $W_r = \mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r)) \cap U_r$ and positional almost-sure winning policies σ_r on W_r. Recall that $\sigma_r(s) \in \mathcal{A}^{\boldsymbol{v}^\sigma}(s)$ for every $s \in W_r$ by construction, since $\mathcal{M}_r$ offers no other actions.

We use freely the fact, established in Section 4, that $p \cdot \boldsymbol{u} = \boldsymbol{u}_s$ for every tight action $a \in A^{\boldsymbol{u}}(s)$ and every p in the tight face $\mathcal{P}^{\boldsymbol{u}}(s,a)$, the support property of W_r recorded in Lemma 37 below, and the elementary facts about one-step minima: since each uncertainty set is a nonempty compact polytope and $p \mapsto p \cdot \boldsymbol{u}$ is linear, every infimum $\inf_{p \in \mathcal{P}(s,a)} p \cdot \boldsymbol{u}$ is attained, and the set of minimisers is the nonempty face of $\mathcal{P}(s,a)$ obtained by adjoining the single linear inequality $p \cdot \boldsymbol{u} \le \nu_{\boldsymbol{u}}(s,a)$, whose vertices are vertices of $\mathcal{P}(s,a)$. In particular every $\arg\min$ below is nonempty, and so is every uncertainty set of $\mathcal{M}_r$, being the image of a nonempty set under κ_r. Hence, every $\mathcal{M}_r$ is a legitimate linearly defined RMDP and Proposition 8 applies to it.

Markov-chain preliminaries. When both policies are positional the pair (σ, τ) induces a finite Markov chain over $\mathcal{S}$ with transition matrix $P(s,t) = \tau(s, \sigma(s))(t)$. We use the standard theory of finite Markov chains (Baier and Katoen 2008): with probability 1 the play eventually enters a closed recurrent class $\mathcal{RC}$ and visits all its states infinitely often, so $\mathsf{Inf}(\pi) = \mathcal{RC}$ almost surely conditioned on entering $\mathcal{RC}$. Call $\mathcal{RC}$ *even* if $\max C(\mathcal{RC})$ is even and *odd* otherwise; conditioned on entering $\mathcal{RC}$, the play satisfies φ with probability 1 if $\mathcal{RC}$ is even and 0 if it is odd. Hence, writing $\Diamond\mathcal{RC}$ for the event of being absorbed in $\mathcal{RC}$,

$$\Pr_s^{\sigma,\tau}[\varphi] \;=\; \sum_{\mathcal{RC} \text{ even}} \Pr_s^{\sigma,\tau}[\Diamond \mathcal{RC}]. \tag{4}$$

We also record the *harmonicity* of the satisfaction probability of a prefix-independent objective under a positional pair: writing $w_s := \Pr_s^{\sigma,\tau}[\varphi]$,

$$w_s \;=\; \sum_{t \in \mathcal{S}} P(s,t)\, w_t, \qquad \text{i.e.} \qquad w = Pw. \tag{5}$$

This is immediate from the Markov property together with prefix-independence of φ: conditioned on the first transition $s \to t$, the remaining play is distributed as a play from t under the same positional pair, and deleting the one-step prefix does not change whether the play satisfies φ.

G.1 Optimality conditions

Proposition 8 yields the usual one-step optimality (Bellman) conditions. Note that, by compactness of the uncertainty sets, the inner operators below are minima and not merely infima.

Proposition 30 (Optimality conditions)**.** *For every $s \in \mathcal{S}$,*

$$\boldsymbol{v}_s^{\mathcal{M}}(\varphi) \;=\; \max_{a \in \mathcal{A}} \min_{p \in \mathcal{P}(s,a)} \; p \cdot \boldsymbol{v}^{\mathcal{M}}(\varphi). \tag{6}$$

Moreover, for the RMC $\mathcal{M}^\sigma$ induced by a positional agent policy σ,

$$\boldsymbol{v}_s^{\sigma} \;=\; \min_{p \in \mathcal{P}(s,\sigma(s))} p \cdot \boldsymbol{v}^{\sigma} \qquad \textit{for every } s \in \mathcal{S}, \tag{7}$$

and the minimum in (7) *is attained* simultaneously at all states *by the positional environment policy of Proposition 8.*

Proof of Proposition 30. Both identities peel off the first transition out of s and identify the value of the remaining play with the value vector again. This is legitimate because parity is *prefix-independent*, i.e. whether a path satisfies $\mathsf{Parity}(C)$ depends only on $\mathsf{Inf}(\pi)$, so deleting or prepending a finite prefix never changes membership.

Proof of (7). Write $u := \boldsymbol{v}^\sigma$. By the last clause of Proposition 8 there is a positional environment policy τ^* of $\mathcal{M}^\sigma$ with $\Pr_t^{\tau^*}[\varphi] = u_t$ for every t. Applying (5) to the chain it induces gives $u_s = \tau^*(s) \cdot u \ge \min_{p \in \mathcal{P}(s,\sigma(s))} p \cdot u$. Conversely, fix $p \in \mathcal{P}(s,\sigma(s))$ and let τ_p be the environment policy that plays p on the first step at s and follows τ^* afterwards. It is admissible because $p \in \mathcal{P}(s,\sigma(s))$. By prefix-independence, $\Pr_s^{\tau_p}[\varphi] = \sum_t p(t)\, \Pr_t^{\tau^*}[\varphi] = p \cdot u$, whence $u_s = \inf_\tau \Pr_s^\tau[\varphi] \le p \cdot u$. Minimising over p gives $u_s \le \min_p p \cdot u$, and the two inequalities give (7). The final clause of the statement is the first display: $\tau^*(s)$ attains the minimum at every s at once.

Proof of (6). Write $v := \boldsymbol{v}^{\mathcal{M}}(\varphi)$.

($\leq$) By Proposition 8 there is a positional agent policy σ^* with $\boldsymbol{v}^{\sigma^*} = v$. Applying (7) to σ^* at s, with $a^* = \sigma^*(s)$,

$$v_s = \min_{p \in \mathcal{P}(s,a^*)} p \cdot v \;\leq\; \max_{a \in \mathcal{A}} \min_{p \in \mathcal{P}(s,a)} p \cdot v.$$

($\geq$) By Proposition 8 fix a positional environment policy τ^* with $\sup_\sigma \Pr_t^{\sigma,\tau^*}[\varphi] = v_t$ for every t. Fixing τ^* turns $\mathcal{M}$ into an ordinary MDP with transition function $\delta(t,a) = \tau^*(t,a)$ whose parity value at t is v_t. This MDP has a positional optimal agent policy $\bar{\sigma}$, so $\Pr_t^{\bar{\sigma},\tau^*}[\varphi] = v_t$ for every t. Fix $a \in \mathcal{A}$ and let σ_a play a on the first step at s and follow $\bar{\sigma}$ afterwards. By prefix-independence,

$$\Pr_s^{\sigma_a,\tau^*}[\varphi] = \sum_{t \in \mathcal{S}} \tau^*(s,a)(t) \Pr_t^{\bar{\sigma},\tau^*}[\varphi] = \tau^*(s,a) \cdot v \;\geq\; \min_{p \in \mathcal{P}(s,a)} p \cdot v.$$

Since $v_s = \sup_\sigma \Pr_s^{\sigma,\tau^*}[\varphi] \geq \Pr_s^{\sigma_a,\tau^*}[\varphi]$, we get $v_s \geq \min_{p \in \mathcal{P}(s,a)} p \cdot v$ for every a, and taking the maximum over the finite set $\mathcal{A}$ gives ($\geq$). □

G.2 Formal Definition of Value-Class RMDPs

Throughout this subsection we fix a positional agent policy σ, write $\boldsymbol{u} := \boldsymbol{v}^\sigma$ for the value vector of the induced RMC $\mathcal{M}^\sigma$, and fix a value r with $\boldsymbol{u}$ with $U_r := \mathsf{VC}^\sigma(r) \neq \emptyset$. Recall from Section 4 that an action a is *tight* at s if $\nu(s,a) = \boldsymbol{u}_s$, that $\mathcal{A}^{\boldsymbol{u}}(s)$ denotes the set of tight actions at s, and that $\mathcal{P}^{\boldsymbol{u}}(s,a)$ denotes the *tight face* at (s,a).

The construction of $\mathcal{M}_r$ restricts $\mathcal{M}$ to the states of U_r, to the tight actions, and to the tight faces, and merges everything outside U_r into a single fresh state. The merging is carried out by the following map.

Definition 31 (Collapse map)**.** *Let $\perp \notin \mathcal{S}$ be a fresh state and let $c_r : \mathcal{S} \to U_r \cup \{\perp\}$ fix every state of U_r and send every state of $\mathcal{S} \setminus U_r$ to $\perp$. The* collapse map *κ_r is the push-forward of distributions along c_r, i.e. $\kappa_r : \Delta(\mathcal{S}) \to \Delta(U_r \cup \{\perp\})$ with*

$$\kappa_r(p)(t) = p(t) \quad (t \in U_r), \qquad \kappa_r(p)(\perp) = \sum_{t \in \mathcal{S} \setminus U_r} p(t).$$

We extend κ_r to sets of distributions by taking images, $\kappa_r(X) = \{\kappa_r(p) \mid p \in X\}$.

Thus κ_r leaves the mass inside the value class where it is and lumps all remaining mass onto $\perp$. The two cases of Definition 31 are exhaustive and disjoint, so $\kappa_r(p)$ is again a distribution. Note that κ_r is the restriction to $\Delta(\mathcal{S})$ of a linear map $\mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{U_r \cup \{\perp\}}$; in particular it maps polytopes to polytopes, and it is in general not injective, since it forgets how the escaping mass is distributed over $\mathcal{S} \setminus U_r$.

Definition 32 (Value-class RMDP)**.** *The* value-class RMDP *of σ at r is the RMDP $\mathcal{M}_r = (\mathcal{S}_r, \mathcal{A}_r, \mathcal{P}_r)$ with coloring $C_r : \mathcal{S}_r \to [d+1]$ given by:*

1. *$\mathcal{S}_r = U_r \uplus \{\perp\}$ and $\mathcal{A}_r = \mathcal{A} \uplus \{a_\perp\}$;*
2. *for $s \in U_r$ and $a \in \mathcal{A}^{\boldsymbol{u}}(s)$,*
$$\mathcal{P}_r(s,a) = \kappa_r(\mathcal{P}^{\boldsymbol{u}}(s,a))\,;$$
3. *$\mathcal{P}_r(s,a) = \{\delta_\perp\}$ for $s \in U_r$ and $a \notin \mathcal{A}^{\boldsymbol{u}}(s)$, and $\mathcal{P}_r(\perp,a) = \{\delta_\perp\}$ for every $a \in \mathcal{A}_r$;*
4. *$C_r(s) = C(s)$ for $s \in U_r$, and $C_r(\perp) = 2\lceil d/2 \rceil + 1$.*

The state $\perp$ is absorbing and carries an odd color dominating every color of $[d]$, so any play reaching $\perp$ visits only $\perp$ from then on and violates $\mathsf{Parity}(C_r)$. Leaving the value class is therefore losing for the agent in $\mathcal{M}_r$, irrespective of the values in $\mathcal{M}$ of the states left for; likewise, by item (3), playing a non-tight action is losing. Consequently an optimal agent policy of $\mathcal{M}_r$ plays only tight actions on its almost-sure winning region, and we may and do treat $\mathcal{A}^{\boldsymbol{u}}(s)$ as the actions genuinely available at $s \in U_r$.

Lemma 33 (Well-definedness)**.** *Every uncertainty set of $\mathcal{M}_r$ is a nonempty polytope, and $\mathcal{M}_r$ is a linearly defined RMDP of description size $O(|\mathcal{M}|)$. Moreover $\mathcal{A}^{\boldsymbol{u}}(s) \neq \emptyset$ for every $s \in U_r$, and $\sum_r |\mathcal{M}_r| = O(|\mathcal{M}|)$, the sum ranging over the values of $\boldsymbol{u}$.*

Proof. The policy σ cannot be improved quantitatively, so, the action $\sigma(s)$ is tight at every $s \in \mathcal{S}$, so $\mathcal{A}^{\boldsymbol{u}}(s) \neq \emptyset$. For $a \in \mathcal{A}^{\boldsymbol{u}}(s)$ the tight face $\mathcal{P}^{\boldsymbol{u}}(s,a)$ is the face of $\mathcal{P}(s,a)$ cut out by the single inequality $p \cdot \boldsymbol{u} \leq \nu(s,a)$, hence a nonempty polytope, and $\mathcal{P}_r(s,a)$ is its image under the linear map κ_r, hence again a nonempty polytope. For linear definability,

$$\mathcal{P}_r(s,a) = \Big\{ q \;\Big|\; \exists p :\; A_{s,a}\, p \leq b_{s,a},\; p \cdot \boldsymbol{u} \leq \boldsymbol{u}_s,\; q = \kappa_r(p) \Big\},$$

which presents $\mathcal{P}_r(s,a)$ as the projection onto the q-coordinates of a polyhedron in the variables (q,p), obtained from the description of $\mathcal{P}(s,a)$ by adjoining one inequality and the defining equalities of κ_r. Note that only $\boldsymbol{u}$ enters this description; the one-step optima $\nu(s,a)$ are used solely in the tightness test of items (2) and (3). Hence $\mathcal{M}_r$ has description size $O(|\mathcal{M}|)$. Finally the value classes are pairwise disjoint and $\mathcal{S}_r = U_r \uplus \{\perp\}$, so $\sum_r |\mathcal{S}_r| \leq |\mathcal{S}| + |\mathcal{S}|$ and the total description size of all value-class RMDPs is $O(|\mathcal{M}|)$. □

Since Lemma 33 makes $\mathcal{M}_r$ a linearly defined RMDP, Proposition 8 applies to it, as do the results of Section 3 instantiated at a fixed positional agent policy. We write $W_r = \mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r)) \cap U_r$ for its almost-sure winning region and σ_r for a corresponding positional almost-sure winning policy.

G.3 Transfer between the RMDP and its trap value-class RMDPs

The trap value-class RMDP $\mathcal{M}_r$ lives over $U_r \cup \{\bot\}$, so its one-step choices are distributions over $U_r \cup \{\bot\}$ and not elements of $\mathcal{P}(s,a) \subseteq \Delta(\mathcal{S})$. The following lemma does the translation once and for all. It is used at every point below where a play of $\mathcal{M}$ is recognised as a play of $\mathcal{M}_r$ or conversely. Recall the collapse map

$$\kappa_r : \Delta(\mathcal{S}) \to \Delta(U_r \cup \{\bot\}), \qquad \kappa_r(p)(t) = p(t) \;\; (t \in U_r), \qquad \kappa_r(p)(\bot) = p(\mathcal{S} \setminus U_r),$$

and that $\mathcal{P}_r(s,a) = \kappa_r(\mathcal{P}^{\boldsymbol{v}^\sigma}(s,a))$ for $s \in U_r$ and a tight at s.

Lemma 34 (Transfer)**.** *Let $\boldsymbol{u} = \boldsymbol{v}^\sigma$, let $r < 1$ with $U_r \neq \emptyset$, let $s \in U_r$ and let $a \in \mathcal{A}^{\boldsymbol{u}}(s)$. Then:*

1. (Push-forward.) *For every $p \in \mathcal{P}^{\boldsymbol{u}}(s,a)$ we have $\kappa_r(p) \in \mathcal{P}_r(s,a)$.*
2. (Pull-back.) *For every $q \in \mathcal{P}_r(s,a)$ there is $p \in \mathcal{P}^{\boldsymbol{u}}(s,a)$ with $\kappa_r(p) = q$. Such a p is a legal environment choice of $\mathcal{M}$ at (s,a) and satisfies $p \cdot \boldsymbol{u} = \boldsymbol{u}_s$.*
3. (Plays.) *Let σ' be a positional agent policy of $\mathcal{M}$ that plays tight actions at every state of U_r, and let $X \subseteq U_r$. Suppose τ is a positional environment policy of $\mathcal{M}$ such that $\tau(t,\sigma'(t))$ lies in the tight face at $(t,\sigma'(t))$ and is supported in X. Then $\kappa_r \circ \tau$ restricted to X is a legal environment policy of $\mathcal{M}_r$, σ' restricted to X is a legal agent policy of $\mathcal{M}_r$, and for every $x \in X$ the two induced measures on plays coincide: the play of $\mathcal{M}$ from x never leaves X and the corresponding play of $\mathcal{M}_r$ never visits $\bot$.*

Proof of Lemma 34. (1) is the definition of $\mathcal{P}_r(s,a)$ as the image of the tight face. (2) holds because a set-image consists exactly of the points having a preimage. The two properties of p are $\mathcal{P}^{\boldsymbol{u}}(s,a) \subseteq \mathcal{P}(s,a)$ and the definition of the tight face together with tightness of a. For (3), we check the three assertions in turn. The agent's actions are tight, hence available in $\mathcal{M}_r$. The environment's choices lie in $\mathcal{P}_r$ by (2). These choices are supported in $X \subseteq U_r$, so $\kappa_r(\tau(t,\sigma'(t))) = \tau(t,\sigma'(t))$ by (1). The two transition kernels therefore agree on X, and neither play ever leaves X. Equal kernels on X give equal measures on cylinders, hence equal measures on plays. □

Note that, by part (4), on plays that stay in U_r the models $\mathcal{M}$ and $\mathcal{M}_r$ are literally the same Markov chain, and C_r agrees with C there. So, such a play satisfies φ if and only if it satisfies $\mathsf{Parity}(C_r)$. This is the only fact about the collapse that the correctness proofs use.

G.4 Markov-chain lemmas

Lemma 35 ((Sub/super)martingale decomposition)**.** *Let (σ,τ) be positional, inducing a finite Markov chain with matrix P, and let $\boldsymbol{u} \in [0,1]^{\mathcal{S}}$.*

1. *If $(P\boldsymbol{u})(s) \geq \boldsymbol{u}(s)$ for all s, then $\boldsymbol{u}$ is constant on every closed recurrent class $\mathcal{RC}$. Write $\boldsymbol{u}_{\mathcal{RC}}$ for this value. Moreover, $\boldsymbol{u}(s) \leq \sum_{\mathcal{RC}} \Pr_s[\Diamond \mathcal{RC}]\, \boldsymbol{u}_{\mathcal{RC}}$ for all s.*
2. *Symmetrically, if $(P\boldsymbol{u})(s) \leq \boldsymbol{u}(s)$ for all s, then $\boldsymbol{u}$ is constant on every closed recurrent class and $\boldsymbol{u}(s) \geq \sum_{\mathcal{RC}} \Pr_s[\Diamond \mathcal{RC}]\, \boldsymbol{u}_{\mathcal{RC}}$.*

Moreover, in either case the defining inequality is an equality *at every state of every closed recurrent class.*

Proof of Lemma 35. We prove (1); (2) is symmetric. Let $\mathcal{RC}$ be a closed recurrent class with stationary distribution μ, so $\mu > 0$ on $\mathcal{RC}$ and μ is supported on $\mathcal{RC}$. Since $\mathcal{RC}$ is closed, $\sum_{x \in \mathcal{RC}} \mu(x)(P\boldsymbol{u})(x) = \sum_{y \in \mathcal{RC}} \boldsymbol{u}(y) \sum_{x \in \mathcal{RC}} \mu(x) P(x,y) = \sum_{y \in \mathcal{RC}} \mu(y)\boldsymbol{u}(y)$ by stationarity. Comparing with the pointwise inequality $(P\boldsymbol{u})(x) \geq \boldsymbol{u}(x)$ and using $\mu > 0$, every such inequality is an *equality* on $\mathcal{RC}$; thus $\boldsymbol{u}$ restricted to $\mathcal{RC}$ is harmonic for an irreducible finite chain and is therefore constant. For the inequality, $\boldsymbol{u}(X_n)$ is a bounded submartingale, so $\boldsymbol{u}(s) \leq \mathbb{E}[\boldsymbol{u}(X_n)]$ for all n. The play is absorbed in some $\mathcal{RC}$ almost surely and $\boldsymbol{u} \equiv \boldsymbol{u}_{\mathcal{RC}}$ there, so $\boldsymbol{u}(X_n) \to \boldsymbol{u}_{\mathcal{RC}(\pi)}$ almost surely, and bounded convergence gives $\boldsymbol{u}(s) \leq \sum_{\mathcal{RC}} \Pr_s[\Diamond \mathcal{RC}]\, \boldsymbol{u}_{\mathcal{RC}}$. □

The next lemma states that a policy that is almost-sure winning for a prefix-independent objective on the whole almost-sure region can never be forced to leave it.

Lemma 36 (Invariance of almost-sure regions)**.** *Let $\mathcal{M}'$ be an RMDP, let $W = \mathsf{AS}_{\mathcal{M}'}(\varphi')$ for a prefix-independent objective φ', and let σ_W be an agent policy that is almost-sure winning from every state of W. Then for every $w \in W$ and every environment policy τ, $\Pr_w^{\sigma_W,\tau}[\forall n : X_n \in W] = 1$.*

Proof of Lemma 36. Suppose towards a contradiction that under some τ the play from w leaves W with positive probability, and fix a history h of positive probability ending in a state $s' \notin W$. Let τ' be any environment policy and let τ'' be the environment policy that behaves as τ along h and as τ' afterwards. Then h still has positive probability under (σ_W, τ''). Since σ_W is almost-sure winning from w, $\Pr_w^{\sigma_W,\tau''}[\varphi'] = 1$, and conditioning on the positive-probability event h and using prefix-independence of φ' gives $\Pr_{s'}^{(\sigma_W)|_h,\tau'}[\varphi'] = 1$. As τ' was arbitrary, the shifted policy $(\sigma_W)|_h$ is almost-sure winning from s', i.e. $s' \in W$, a contradiction. □

Applied to the trap value-class RMDPs $\mathcal{M}_r$, Lemma 36 yields the property of W_r announced in Section 4 and used throughout the correctness proofs below: the environment choices available in $\mathcal{M}_r$ cannot push the play out of W_r. Since $\bot \notin W_r$, this single statement also says that the play cannot leave the value class U_r.

Lemma 37 (Support in W_r)**.** *Assume $I = \emptyset$, and let $r < 1$ with $U_r \neq \emptyset$. Then $\mathrm{supp}(q) \subseteq W_r$ for every $s \in W_r$ and every $q \in \mathcal{P}_r(s, \sigma_r(s))$. In particular $q(\bot) = 0$, so no environment choice of $\mathcal{M}_r$ available against σ_r leaves U_r.*

Proof of Lemma 37. The objective $\mathsf{Parity}(C_r)$ is prefix-independent and σ_r is almost-sure winning from every state of $W_r \subseteq \mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r))$, so Lemma 36 applied to $\mathcal{M}_r$ gives $\Pr_s^{\sigma_r,\tau}[\forall n : X_n \in W_r] = 1$ for every environment policy τ of $\mathcal{M}_r$. Were some $q \in \mathcal{P}_r(s, \sigma_r(s))$ to place positive mass outside W_r, the environment policy playing q at s would leave W_r with positive probability. □

Remark 38. *The policy σ_r plays only* tight *actions on W_r: the only actions available at $s \in U_r$ in $\mathcal{M}_r$ are the tight ones $\mathcal{A}^{\boldsymbol{v}^\sigma}(s)$, which form a nonempty set because $\sigma(s)$ is tight by* (7)*.*

G.5 Environment spoiling

Proposition 39 (Environment spoiling on on value-class RMDPs)**.** *Let $r < 1$ with $U_r \neq \emptyset$ and suppose $W_r = \mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r)) = \emptyset$. Then the environment has a positional policy τ_r on $\mathcal{M}_r$ such that $\Pr_s^{\sigma'',\tau_r}[\mathsf{Parity}(C_r)] = 0$ for every agent policy σ'' and every state s.*

Proof of Proposition 39. Write $\varphi_r := \mathsf{Parity}(C_r)$. The instance $\mathcal{M}_r$ is a linearly defined RMDP: at $s \in U_r$ and a tight, $\mathcal{P}_r(s,a) = \kappa_r(\mathcal{P}^{\boldsymbol{v}^\sigma}(s,a))$ is the image under a linear map of the face of $\mathcal{P}(s,a)$ cut out by one additional linear inequality, hence a nonempty polytope, presented as the projection of a linear system in the variables (q,p). Also, $\mathcal{P}_r(\bot, a) = \{\delta_\bot\}$. Proposition 8 requires only that the uncertainty sets be nonempty bounded polytopes, and hence applies to $\mathcal{M}_r$ with objective φ_r. It yields that there exists a positional agent policy σ^* and a positional environment policy τ_r such that $\Pr_s^{\sigma^*,\tau_r}[\varphi_r] = \boldsymbol{v}_s^{\mathcal{M}_r}(\varphi_r)$. We claim $\boldsymbol{v}_s^{\mathcal{M}_r}(\varphi_r) = 0$ for every s. Suppose not, and fix s_0 with $\boldsymbol{v}_{s_0}^{\mathcal{M}_r}(\varphi_r) > 0$. Being positional, (σ^*, τ_r) induces a finite Markov chain over $U_r \cup \{\bot\}$, so by (4) applied with the coloring C_r,

$$\boldsymbol{v}_s^{\mathcal{M}_r}(\varphi_r) = \Pr_s^{\sigma^*,\tau_r}[\varphi_r] = \sum_{\mathcal{RC} \text{ even}} \Pr_s[\Diamond \mathcal{RC}],$$

where "even" refers to $\max C_r(\mathcal{RC})$. As $\boldsymbol{v}_{s_0}^{\mathcal{M}_r}(\varphi_r) > 0$, some even class $\mathcal{RC}$ is reached with positive probability. Fix $x \in \mathcal{RC}$. From x the chain stays in $\mathcal{RC}$ almost surely, so $\boldsymbol{v}_x^{\mathcal{M}_r}(\varphi_r) = \Pr_x^{\sigma^*,\tau_r}[\varphi_r] = 1$. But σ^* secures the value, so $\inf_\tau \Pr_x^{\sigma^*,\tau}[\varphi_r] = \boldsymbol{v}_x^{\mathcal{M}_r}(\varphi_r) = 1$, i.e. $\Pr_x^{\sigma^*,\tau}[\varphi_r] = 1$ for *every* environment policy τ of $\mathcal{M}_r$. Thus σ^* is almost-sure winning from x in $\mathcal{M}_r$, so $x \in \mathsf{AS}_{\mathcal{M}_r}(\varphi_r)$, contradicting $\mathsf{AS}_{\mathcal{M}_r}(\varphi_r) = \emptyset$.

Therefore $\boldsymbol{v}_s^{\mathcal{M}_r}(\varphi_r) = 0$ for all s, and since τ_r attains the value, $\Pr_s^{\sigma'',\tau_r}[\mathsf{Parity}(C_r)] = 0$ for every agent policy σ'' and every state s. □

G.6 Correctness of the improvement steps

Proof of Lemma 10. Write $\boldsymbol{u} := \boldsymbol{v}^\sigma$. Fix a positional environment policy τ and let P be the matrix of the chain induced by (σ', τ). For $s \notin I$ we have $\sigma'(s) = \sigma(s)$ and, by (7), $(P\boldsymbol{u})(s) = \tau(s,\sigma(s)) \cdot \boldsymbol{u} \geq \min_{p \in \mathcal{P}(s,\sigma(s))} p \cdot \boldsymbol{u} = \boldsymbol{u}_s$. For $s \in I$, $(P\boldsymbol{u})(s) \geq \min_{p \in \mathcal{P}(s,\sigma'(s))} p \cdot \boldsymbol{u} = \nu(s,\sigma'(s)) = \max_a \nu(s,a) > \boldsymbol{u}_s$, since $\sigma'(s)$ maximises $\nu(s,\cdot)$. Hence $\boldsymbol{u}$ is a submartingale for P, and by Lemma 35(1) it is constant on every closed recurrent class $\mathcal{RC}$ with $\boldsymbol{u}(s) \leq \sum_{\mathcal{RC}} \Pr_s[\Diamond \mathcal{RC}]\, \boldsymbol{u}_{\mathcal{RC}}$.

First, no closed recurrent class meets I: on $\mathcal{RC}$ the submartingale inequalities are equalities (Lemma 35), whereas at states of I the inequality is strict. Hence $\sigma' = \sigma$ on every closed recurrent class $\mathcal{RC}$.

Second, every closed recurrent class $\mathcal{RC}$ with $\boldsymbol{u}_{\mathcal{RC}} > 0$ is even. Let $x \in \mathcal{RC}$ and let $\widetilde{\tau}$ be any environment policy of $\mathcal{M}^\sigma$ that agrees with τ on $\mathcal{RC}$. Since $\mathcal{RC}$ is closed under the kernels of (σ,τ) and $\sigma' = \sigma$ there, the play from x under $(\sigma, \widetilde{\tau})$ stays in $\mathcal{RC}$ forever and satisfies φ with probability 1 if $\mathcal{RC}$ is even and 0 if odd. But $\Pr_x^{\sigma,\widetilde{\tau}}[\varphi] \geq \inf_{\tau''} \Pr_x^{\sigma,\tau''}[\varphi] = \boldsymbol{v}_x^\sigma = \boldsymbol{u}_{\mathcal{RC}} > 0$, so $\mathcal{RC}$ is even.

Combining with (4) and using that odd classes have $\boldsymbol{u}_{\mathcal{RC}} = 0$,

$$\Pr_s^{\sigma',\tau}[\varphi] = \sum_{\mathcal{RC} \text{ even}} \Pr_s[\Diamond \mathcal{RC}] \geq \sum_{\mathcal{RC}} \Pr_s[\Diamond \mathcal{RC}]\, \boldsymbol{u}_{\mathcal{RC}} \geq \boldsymbol{u}_s.$$

Taking the infimum over positional τ, which by Proposition 8 applied to the RMC $\mathcal{M}^{\sigma'}$ computes $\boldsymbol{v}^{\sigma'}$, gives $\boldsymbol{v}^{\sigma'} \geq \boldsymbol{v}^{\sigma}$.

For strictness at $s \in I$: by (7) applied to $\mathcal{M}^{\sigma'}$ and monotonicity of $\boldsymbol{u} \mapsto \min_p p \cdot \boldsymbol{u}$,

$$\begin{aligned} \boldsymbol{v}_s^{\sigma'} &= \min_{p \in \mathcal{P}(s,\sigma'(s))} p \cdot \boldsymbol{v}^{\sigma'} \;\geq\; \min_{p \in \mathcal{P}(s,\sigma'(s))} p \cdot \boldsymbol{v}^{\sigma} \\ &= \max_a \min_{p \in \mathcal{P}(s,a)} p \cdot \boldsymbol{v}^{\sigma} > \boldsymbol{v}_s^{\sigma}, \end{aligned}$$

where the last equality uses that $\sigma'(s)$ attains the outer maximum defining I. □

Proof of Lemma 11. Write $\boldsymbol{u} := \boldsymbol{v}^{\sigma}$. Since $I = \emptyset$,

$$\boldsymbol{u}_s = \max_a \min_{p \in \mathcal{P}(s,a)} p \cdot \boldsymbol{u} \qquad \text{for all } s, \tag{8}$$

and the action $\sigma(s)$ attains it by (7); hence every action played by σ' is tight: $\sigma(s)$ is tight at s for every s, and $\sigma_r(s)$ is tight at s for every $s \in W_r$ by construction of $\mathcal{M}_r$ (Remark 38).

Soundness. Fix a positional environment policy τ of $\mathcal{M}$ with matrix P (for policy σ'). Tightness gives $(P\boldsymbol{u})(s) \geq \boldsymbol{u}_s$ for all s, so Lemma 35(1) applies. It suffices to show that every closed recurrent class $\mathcal{RC}$ with $\boldsymbol{u}_{\mathcal{RC}} > 0$ is even. The bound $\Pr_s^{\sigma',\tau}[\varphi] \geq \boldsymbol{u}_s$ then follows exactly as in Lemma 10, and taking the infimum over positional τ gives $\boldsymbol{v}^{\sigma'} \geq \boldsymbol{v}^{\sigma}$.

Let $\mathcal{RC}$ be a closed recurrent class and put $r := \boldsymbol{u}_{\mathcal{RC}}$. Since $\boldsymbol{u} \equiv r$ on $\mathcal{RC}$ we have $\mathcal{RC} \subseteq U_r$, so C_r agrees with C on $\mathsf{Inf}(\pi) = \mathcal{RC}$. Moreover, by the last clause of Lemma 35 the submartingale inequality is an equality at every $x \in \mathcal{RC}$, that is, $\tau(x, \sigma'(x)) \cdot \boldsymbol{u} = \boldsymbol{u}_x$. Since $\sigma'(x)$ is tight, this places $\tau(x, \sigma'(x))$ in the tight face $\mathcal{P}^{\boldsymbol{u}}(x, \sigma'(x))$. It is moreover supported in $\mathcal{RC}$, as $\mathcal{RC}$ is closed. Hence Lemma 34(3), applied with $X = \mathcal{RC}$, identifies the play of $\mathcal{M}$ from any $x \in \mathcal{RC}$ under (σ', τ) with a play of $\mathcal{M}_r$ under $(\sigma', \kappa_r \circ \tau)$ that never visits $\bot$. Two cases.

Case 1: $\mathcal{RC} \cap W_r = \emptyset$. Then $\sigma' = \sigma$ on $\mathcal{RC}$, and the argument of Lemma 10 shows that if $r > 0$ then $\mathcal{RC}$ is even.

Case 2: $\mathcal{RC} \cap W_r \neq \emptyset$. Let $x \in \mathcal{RC} \cap W_r$. At states of W_r the policy σ' plays σ_r, and by the previous paragraph the environment's choices on $\mathcal{RC}$ are, after κ_r, choices of $\mathcal{M}_r$; hence Lemma 37 applies and every successor of a state of $\mathcal{RC} \cap W_r$ lies in W_r, while lying in $\mathcal{RC}$ because $\mathcal{RC}$ is closed. Thus the play from x never leaves $\mathcal{RC} \cap W_r$, and since $\mathcal{RC}$ is a recurrent class every state of $\mathcal{RC}$ is visited almost surely, so $\mathcal{RC} \subseteq W_r$ and σ' follows σ_r throughout $\mathcal{RC}$. Consequently the play from x is a play of $\mathcal{M}_r$ consistent with σ_r, started in $x \in \mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r))$. It therefore satisfies $\mathsf{Parity}(C_r)$ almost surely. Now $\mathsf{Inf}(\pi) = \mathcal{RC}$ almost surely, and $C_r = C$ on $\mathcal{RC}$. Hence $\max C(\mathcal{RC})$ is even, i.e. $\mathcal{RC}$ is even.

This proves the claim, and exactly as in Lemma 10 we conclude $\boldsymbol{v}^{\sigma'} \geq \boldsymbol{v}^{\sigma}$.

Strictness on W. Let $x \in W_r$, so $\boldsymbol{u}_x = r < 1$. By Proposition 8 applied to the RMC $\mathcal{M}^{\sigma'}$, fix a positional environment policy τ^* attaining $\boldsymbol{v}^{\sigma'}$ simultaneously at all states, i.e. $\boldsymbol{v}_t^{\sigma'} = \Pr_t^{\sigma',\tau^*}[\varphi]$ for all t. Call a state t *loose* if $\tau^*(t, \sigma'(t)) \cdot \boldsymbol{u} > \boldsymbol{u}_t$ and let L be the set of loose states. Observe that since $\sigma'(t)$ is tight, a state t is *loose* if $\tau^*(t, \sigma'(t))$ lies off the tight face $\mathcal{P}^{\boldsymbol{u}}(t, \sigma'(t))$. Let T be the first time the play started at x visits L. Note that $\{T = \infty\}$ is exactly the event that the play never visits L.

We first show that $\Pr[\varphi \mid T = \infty] = 1$ whenever $\Pr[T = \infty] > 0$. Before the play visits L, the environment's choices lie on tight faces, so their κ_r-images are available in $\mathcal{M}_r$ by Lemma 34(2). Since $\sigma' = \sigma_r$ on W_r, Lemma 37 keeps the play inside $W_r \subseteq U_r$. In particular $X_n \in W_r$ for every $n \leq T$, and for every n when $T = \infty$. Define an environment policy $\widetilde{\tau}$ of $\mathcal{M}_r$ by $\widetilde{\tau}(t, a) := \kappa_r(\tau^*(t, a))$ whenever $t \in W_r \setminus L$ and a is tight at t (legal precisely because t is not loose) and by an arbitrary element of $\mathcal{P}_r(t, a)$ otherwise. By Lemma 34(3) the chains induced by (σ', τ^*) in $\mathcal{M}$ and by $(\sigma', \widetilde{\tau})$ in $\mathcal{M}_r$ assign the same measure to every set of paths from x avoiding L, since their kernels agree at every state of $W_r \setminus L$. Under $(\sigma_r, \widetilde{\tau})$ the play from x stays in W_r and satisfies $\mathsf{Parity}(C_r)$ almost surely, hence also φ almost surely, as it never visits $\bot$ and $C_r = C$ on U_r. Therefore $\Pr_x^{\sigma',\tau^*}[\{T = \infty\} \cap \neg\varphi] = \Pr_x^{\sigma',\widetilde{\tau}}[\{T = \infty\} \cap \neg\varphi] = 0$, which is the claim.

If $L = \emptyset$ then $T = \infty$ almost surely and $\boldsymbol{v}_x^{\sigma'} = \Pr_x^{\sigma',\tau^*}[\varphi] = 1 > r$, as required. Otherwise set

$$\varepsilon := \min \left\{ \tau^*(t, \sigma'(t)) \cdot \boldsymbol{u} - \boldsymbol{u}_t \;\middle|\; t \in L \right\} \;>\; 0,$$

a minimum over at most $|\mathcal{S}|$ states. On $\{T < \infty\}$ we have $X_T \in W_r \cap L$, hence $\boldsymbol{u}(X_T) = r$, and the step taken at time T gives

$$\mathbb{E}[\boldsymbol{u}(X_{T+1}) \mid T < \infty] = \tau^*(X_T, \sigma'(X_T)) \cdot \boldsymbol{u} \;\geq\; r + \varepsilon.$$

By the strong Markov property, prefix-independence of φ, and $\Pr_t^{\sigma',\tau^*}[\varphi] = \boldsymbol{v}_t^{\sigma'} \geq \boldsymbol{v}_t^{\sigma} = \boldsymbol{u}_t$ (soundness) for the positional continuation from any state t,

$$\Pr[\varphi \mid T < \infty] = \mathbb{E}\big[\boldsymbol{v}^{\sigma'}(X_{T+1}) \mid T < \infty\big] \;\geq\; \mathbb{E}\big[\boldsymbol{u}(X_{T+1}) \mid T < \infty\big] \;\geq\; r + \varepsilon.$$

Combining, since $\boldsymbol{v}_x^{\sigma'} = \Pr_x^{\sigma',\tau^*}[\varphi]$,

$$\boldsymbol{v}_x^{\sigma'} \;\geq\; \Pr[T=\infty]\cdot 1 + \Pr[T<\infty]\cdot(r+\varepsilon) \;\geq\; \min\{1, r+\varepsilon\} \;>\; r = \boldsymbol{u}_x,$$

the last inequality because $r < 1$. □

G.7 Optimality of fixpoints, termination, and complexity

Theorem 40 (Optimality of fixpoints)**.** *Let σ be a positional agent policy whose value vector $\boldsymbol{v}^\sigma$ satisfies*

(i) $\boldsymbol{v}_s^\sigma = \max_a \min_{p\in\mathcal{P}(s,a)} p\cdot\boldsymbol{v}^\sigma$ for all s (equivalently $I=\emptyset$), and
(ii) $W_r = \mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r)) = \emptyset$ for every value $r<1$ of $\boldsymbol{v}^\sigma$.

Then $\boldsymbol{v}^\sigma = \boldsymbol{v}^{\mathcal{M}}(\varphi)$ and σ is optimal. Moreover an optimal positional environment policy exists, obtained by combining, on each value class of value <1, the environment's spoiling policies of the value-class RMDPs (Proposition 39) with value-minimising choices elsewhere.

Proof of Theorem 40. Write $\boldsymbol{u} := \boldsymbol{v}^\sigma$. Since $\boldsymbol{u}$ is guaranteed by σ against every environment policy, $\boldsymbol{v}^{\mathcal{M}}(\varphi) \geq \boldsymbol{u}$ pointwise, it remains to find a single environment policy $\widehat{\tau}$ with

$$\Pr_s^{\sigma',\widehat{\tau}}[\varphi] \leq \boldsymbol{u}_s \qquad \text{for every agent policy } \sigma' \text{ and every } s. \tag{9}$$

Indeed, (9) gives $\boldsymbol{v}_s^{\mathcal{M}}(\varphi) \leq \boldsymbol{u}_s = \boldsymbol{v}_s^\sigma \leq \boldsymbol{v}_s^{\mathcal{M}}(\varphi)$, so $\boldsymbol{v}^\sigma = \boldsymbol{v}^{\mathcal{M}}(\varphi)$, the policy σ is optimal (it attains $\boldsymbol{u}$), and $\widehat{\tau}$ attains $\inf_\tau \sup_{\sigma'}$ and is an optimal environment policy by Proposition 8.

Construction of $\widehat{\tau}$. By (ii) and Proposition 39, for every value $r<1$ of $\boldsymbol{u}$ the environment has a positional policy τ_r *on* $\mathcal{M}_r$ that spoils the trap objective uniformly: $\Pr_s^{\sigma'',\tau_r}[\mathsf{Parity}(C_r)] = 0$ for every agent policy σ'' and every state s of $\mathcal{M}_r$. Equivalently, against τ_r the play in $\mathcal{M}_r$ almost surely either reaches $\bot$ or remains in U_r forever and violates the original parity condition there. Define the positional environment policy $\widehat{\tau}$ of $\mathcal{M}$ at each pair (s,a):

- if $s\in U_r$ for some $r<1$ and a is tight at s: by Lemma 34(2) pick $p_{s,a}\in\mathcal{P}^{\boldsymbol{u}}(s,a)$ with $\kappa_r(p_{s,a}) = \tau_r(s,a)$ and let $\widehat{\tau}(s,a) := p_{s,a}$. This is the *uncollapsed* form of the choice prescribed by τ_r, and it is a legal environment choice of $\mathcal{M}$;
- otherwise (a not tight at s, or $\boldsymbol{u}_s = 1$): let $\widehat{\tau}(s,a)\in\arg\min_{p\in\mathcal{P}(s,a)} p\cdot\boldsymbol{u}$, which is nonempty because $\mathcal{P}(s,a)$ is a nonempty compact polytope.

In all cases $\widehat{\tau}(s,a)\cdot\boldsymbol{u} \leq \boldsymbol{u}_s$: for a tight action, by definition, $p_{s,a}\cdot\boldsymbol{u} = \boldsymbol{u}_s$, the dot product being taken over the *original* vector $\boldsymbol{u}$, i.e. before the collapse and hence including the states outside U_r. For a non-tight action $\min_p p\cdot\boldsymbol{u} < \boldsymbol{u}_s$ by (i). Finally, if $\boldsymbol{u}_s = 1$ then trivially $p\cdot\boldsymbol{u}\leq 1$.

Proof of (9). Fix a starting state s. With $\widehat{\tau}$ fixed and positional, the agent faces an ordinary MDP, which by Remark 28 admits a positional optimal agent policy for the parity objective. So, it suffices to bound $\Pr_s^{\sigma',\widehat{\tau}}[\varphi]$ for positional σ'. Let P be the matrix of the finite Markov chain induced by $(\sigma',\widehat{\tau})$. By what was discussed before and the definition of $\widehat{\tau}$, $(P\boldsymbol{u})(t)\leq\boldsymbol{u}_t$ for all t, so $\boldsymbol{u}$ is a supermartingale and by Lemma 35(2) it is constant on every closed recurrent class $\mathcal{RC}$, with $\boldsymbol{u}_s \geq \sum_{\mathcal{RC}} \Pr_s[\Diamond\mathcal{RC}]\,\boldsymbol{u}_{\mathcal{RC}}$.

Claim: every even closed recurrent class $\mathcal{RC}$ has $\boldsymbol{u}_{\mathcal{RC}} = 1$. Suppose instead $\boldsymbol{u}_{\mathcal{RC}} = r < 1$; then $\mathcal{RC}\subseteq U_r$. On $\mathcal{RC}$ the supermartingale inequalities are equalities, so for every $x\in\mathcal{RC}$ the action $a=\sigma'(x)$ is tight (a non-tight action gives a strict decrease) and hence $\widehat{\tau}(x,a) = p_{x,a}$ is the uncollapsed tight-face choice prescribed by τ_r. As $\mathcal{RC}$ is closed, these choices are supported in $\mathcal{RC}\subseteq U_r$, so Lemma 34(3) with $X=\mathcal{RC}$ identifies the play from x under $(\sigma',\widehat{\tau})$, which stays in $\mathcal{RC}$ forever with probability 1, with a play of $\mathcal{M}_r$ under τ_r that never reaches $\bot$. By the spoiling guarantee of τ_r such a play violates $\mathsf{Parity}(C_r)$ almost surely, and since it never leaves U_r, where $C_r = C$, it violates φ almost surely. As $\mathsf{Inf}(\pi) = \mathcal{RC}$ almost surely, $\max C(\mathcal{RC})$ is odd, contradicting that $\mathcal{RC}$ is even. This proves the claim.

By (4) and the claim,

$$\Pr_s^{\sigma',\widehat{\tau}}[\varphi] = \sum_{\mathcal{RC}\text{ even}} \Pr_s[\Diamond\mathcal{RC}] = \sum_{\mathcal{RC}\text{ even}} \Pr_s[\Diamond\mathcal{RC}]\,\boldsymbol{u}_{\mathcal{RC}} \leq \sum_{\mathcal{RC}} \Pr_s[\Diamond\mathcal{RC}]\,\boldsymbol{u}_{\mathcal{RC}} \leq \boldsymbol{u}_s,$$

which is (9). The optimal environment policy $\widehat{\tau}$ is positional and has exactly the announced structure: uncollapsed tight-face (spoiling) choices inside each value class of value <1, and value-minimising choices elsewhere. □

Proof of Theorem 12. If $\mathsf{Improve}(\mathcal{M},\sigma)\neq\sigma$, then by Lemma 10 or Lemma 11 we have $\boldsymbol{v}^{\sigma'}\geq\boldsymbol{v}^\sigma$ with strict inequality in at least one coordinate. The value vectors are thus strictly increasing in the product order along the run, so no positional policy is visited twice and the loop terminates within the number of positional agent policies, $|\mathcal{A}|^{|\mathcal{S}|}$. On termination σ is a fixpoint: $I=\emptyset$ and $W_r=\emptyset$ for every value $r<1$, the conditions of Theorem 40. By Theorem 40 the returned policy is optimal and $\boldsymbol{v}^\sigma = \boldsymbol{v}^{\mathcal{M}}(\varphi)$. □

Theorem 41 (Complexity)**.** PI *performs at most* $|\mathcal{A}|^{|\mathcal{S}|}$ *iterations. One iteration solves* $O(|\mathcal{S}|^3 + |\mathcal{S}| \cdot |\mathcal{A}|)$ *linear programs, each of size polynomial in* $|\mathcal{M}|$*, performs at most* $|\mathcal{S}|$ *almost-sure parity computations on polytopic RMDPs of description size* $O(|\mathcal{M}|)$*, and performs* $O(|\mathcal{S}|^2 \cdot |\mathcal{A}|)$ *additional bookkeeping. Consequently, writing* $T_{\mathrm{AS}}(n)$ *for the cost of one almost-sure parity computation on an instance of size* n*, one iteration runs in time* $\mathrm{poly}(|\mathcal{M}|) + |\mathcal{S}| \cdot T_{\mathrm{AS}}(O(|\mathcal{M}|))$*, and the whole algorithm in time* $|\mathcal{A}|^{|\mathcal{S}|} \cdot \big(\mathrm{poly}(|\mathcal{M}|) + |\mathcal{S}| \cdot T_{\mathrm{AS}}(O(|\mathcal{M}|))\big)$.

Proof of Theorem 41. The iteration bound is Theorem 12. Consider one call to Improve.

Line 1 computes the parity value vector of the linearly defined RMC $\mathcal{M}^\sigma$. By Theorem 7 and its proof this takes $O(|\mathcal{S}|^3)$ linear programs: computing the almost-sure violation set $W_{\mathrm{odd}} = \mathsf{OddEC}(\mathcal{S})$ performs $O(|\mathcal{S}|^3)$ feasibility and maximisation programs (Lemma 5), and the safety program $(\star)$, which has $O(|\mathcal{S}|^2)$ variables and polynomially many constraints, is solved once per source state. Note that the cubic bound of Lemma 5 covers the *whole* nested recursion of OddEC and MecDecomp, and is not the cost of one MEC decomposition multiplied by the number of OddEC nodes: taking the two recursions together, there are at most $|\mathcal{S}|$ levels, the sets processed at any fixed level are pairwise disjoint, and one invocation of $\mathsf{MecDecomp}(B)$ costs only $O(|B|^2)$ linear programs (Theorem 18), so each level costs $O(|\mathcal{S}|^2)$ programs and the total is $O(|\mathcal{S}|^3)$ rather than $O(|\mathcal{S}|^4)$, See the proof of Lemma 5 in Appendix E. Lines 2–4 solve exactly $|\mathcal{S}| \cdot |\mathcal{A}|$ linear programs, one per pair (s, a), each of them a minimisation of a linear functional over the facet description of $\mathcal{P}(s, a)$. Line 5 and the switch of line 8 are $O(|\mathcal{S}| \cdot |\mathcal{A}|)$ comparisons. This accounts for the $O(|\mathcal{S}|^3 + |\mathcal{S}| \cdot |\mathcal{A}|)$ programs of the statement; each has size polynomial in $|\mathcal{M}|$ and is therefore solvable in time polynomial in $|\mathcal{M}|$.

The loop of lines 11–15 is entered at most $|\mathcal{S}|$ times, once per nonempty value class of value < 1. Each iteration of the loop assembles $\mathcal{M}_r$ and C_r and performs one almost-sure parity computation. The assembly needs no further optimisation. Indeed $\mathcal{M}_r$ has state set $U_r \cup \{\bot\}$. The actions available at $s \in U_r$ are the tight ones, obtained by comparing the already computed $\nu(s, a)$ with $\boldsymbol{v}^\sigma_s$. The colors are inherited from C on U_r, with the single fresh odd color $2\lceil d/2 \rceil + 1$ at $\bot$ and each uncertainty set is

$$\mathcal{P}_r(s, a) = \Big\{ q \;\Big|\; \exists p :\; A_{s,a}\, p \le b_{s,a},\;\; p \cdot \boldsymbol{v}^\sigma \le \boldsymbol{v}^\sigma_s,\;\; q = \kappa_r(p) \Big\},$$

a linear system in (q, p) built from the description of $\mathcal{P}(s, a)$ by adding one inequality and the defining equalities of κ_r. Note that the optima $\nu(s, a)$ enter only through the tightness test: the uncertainty sets themselves are described using $\boldsymbol{v}^\sigma$ alone.

Two consequences for the size. First, each $\mathcal{M}_r$ has description size $O(|\mathcal{M}|)$ and is written down in time $O(|\mathcal{M}|)$. Second, and more sharply, the value classes are pairwise disjoint and $\mathcal{M}_r$ contains only the states of U_r, so $\sum_{r<1} |\mathcal{M}_r| = O(|\mathcal{M}|)$: the instances handed to the almost-sure parity procedure during one iteration have *total* description size $O(|\mathcal{M}|)$. Hence the cost of the loop is $\sum_{r<1} T_{\mathrm{AS}}(|\mathcal{M}_r|) \le |\mathcal{S}| \cdot T_{\mathrm{AS}}(O(|\mathcal{M}|))$ plus $O(|\mathcal{M}|)$ bookkeeping. Grouping the identification of the value classes, the patching of line 17 and the comparisons above gives the stated $O(|\mathcal{S}|^2 \cdot |\mathcal{A}|)$ bookkeeping, and multiplying by the iteration bound gives the overall running time. □

Remark 42 (Space of the almost-sure subroutine)**.** *The almost-sure parity procedure of Asadi et al. (2026b) runs in polynomial space, and it is this property that underlies the third item of Theorem 12. Concretely, the procedure is a recursion of depth at most* $|\mathcal{S}|$ *that stores, at each frame, only a subset of the state space together with* $O(\log |\mathcal{S}|)$ *counters. That is,* $O(|\mathcal{S}|)$ *bits per frame and* $O(|\mathcal{S}|^2)$ *bits over the whole stack, and whose only other work consists of feasibility and optimisation queries against the linear descriptions of the uncertainty sets. Each such query is a linear program of size polynomial in the encoding of the instance, hence solvable in polynomial time and polynomial space, and the space it occupies is released before the next query is issued. The recursion therefore uses polynomial space in total, even though its running time is only quasi-polynomial.*

Remark 43. *A reachability objective* $\mathsf{Reach}(\mathcal{T})$ *is the parity objective of the coloring that assigns color* 2 *to the (absorbing) target states and color* 1 *elsewhere, so our algorithm applies unchanged. For this coloring the qualitative improvement never happens: every value class with* $r < 1$ *contains no target state, so every state of* U_r *has odd color* 1 *in* C *and every state outside* U_r *has the fresh dominating odd color in* C_r*; hence all states are odd-colored under* C_r *and* $\mathsf{AS}_{\mathcal{M}_r}(\mathsf{Parity}(C_r)) = \emptyset$ *automatically.*

H Additional Experimental Details

Benchmarks. We evaluate both algorithms on three benchmarks. Garnet and Inventory Management support both objectives; Frozen Lake supports only reachability.

- **Garnet.** The N states are partitioned into $k + 1$ disjoint subsets $G_0, \ldots, G_k$ (we fix $k = 5$), with G_0 holding half of the states and containing the initial state s_0. Every state has three actions, each leading to $m = \lceil \sqrt{N} \rceil$ distinct successors drawn uniformly at random without replacement. For a state in G_i with $i > 0$, all successors are drawn from within G_i, so the subset is never left. For a state in G_0, the successors are drawn from within G_0, but each action additionally diverts a small random fraction of its probability mass to a uniformly random state of a uniformly random subset among $G_1, \ldots, G_k$; this is how an action taken in G_0 can leave G_0 and enter another subset. The nominal transition distribution over the successors of every state–action pair is drawn uniformly from the probability simplex (i.e., $\mathrm{Dirichlet}(1, \ldots, 1)$), and is then wrapped in an L_∞ or L_1 uncertainty ball of radius $0.5/m$ or 0.1, respectively.

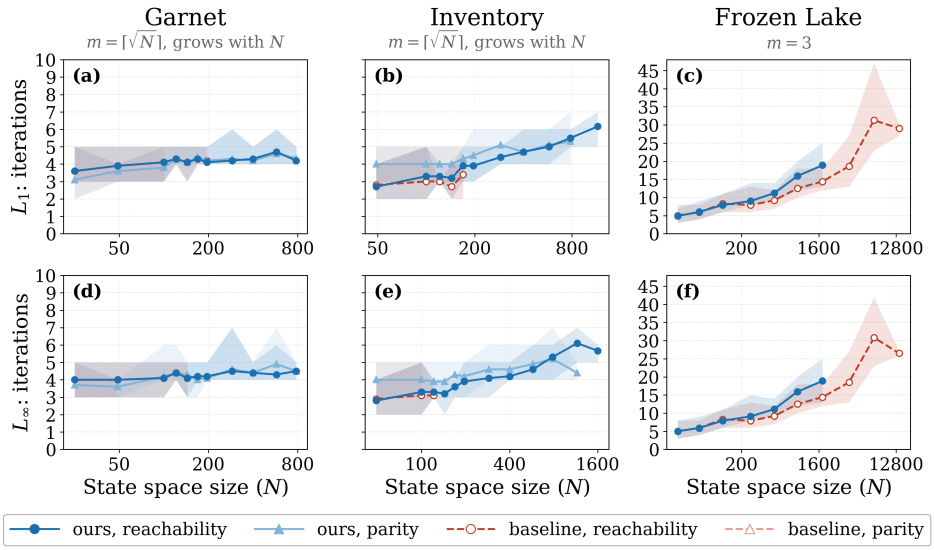

Figure 2: Comparison between number of iterations of the policy iteration algorithm on our benchmarks. The number of iterations is much smaller than the theoretical worst-case bound.

For the reachability objective, each subset independently contains one target state and one trap state, and the goal is to reach the target state of some subset. For the parity objective, every state is assigned a color (priority) drawn uniformly from $\{1, \ldots, |G_i|\}$, and the goal is to satisfy the induced parity condition. In the scaling experiments the branching factor is coupled to the size, $m = \lceil\sqrt{N}\rceil$, so that larger instances are also denser.

- **Inventory Management.** The state is the stock level $s \in \{0, \ldots, N\}$ of a warehouse of capacity N. At each step the agent orders $a \in \{0, \ldots, a_{\max}\}$ new units, the demand $d \in \{0, \ldots, d_{\max}\}$ materializes, and the stock moves deterministically to $s' = s + a - d$. If the demand exceeds the available stock ($d > s + a$), or the delivery pushes the stock over the capacity ($s + a - d > N$), the run instead ends in one of two absorbing failure states. We fix $d_{\max} = \lceil\sqrt{N}\rceil$ and $a_{\max} = \lceil 0.75\, d_{\max}\rceil$, so the number of possible demand values, and hence the branching factor, is $m = d_{\max} + 1 = \Theta(\sqrt{N})$. The nominal demand distribution over $\{0, \ldots, d_{\max}\}$ is a lightly perturbed uniform (each outcome is weighted by an independent $U[0.75, 1.25]$ factor and the weights are then normalized), drawn independently per state, and is wrapped in an L_∞ or L_1 uncertainty ball of radius $\delta = 0.25/m$ or 0.1, respectively.

  Both objectives use a threshold K drawn from $U[0.35, 0.65] \cdot N$. Starting from $s_0 = 0$, the reachability objective is to bring the stock up to $s \geq K$, and the parity objective is to keep the stock at $s \geq K$ infinitely often. Both objectives also share a second threshold T, drawn uniformly from $\{d_{\max}, \ldots, K\}$, which governs a larger order option: once the stock reaches $s \geq T$, an additional order of $d_{\max}$ units becomes available to the agent on top of the amounts $\{0, \ldots, a_{\max}\}$.
- **Frozen Lake.** The agent walks on a $k \times k$ grid from the top-left cell toward the bottom-right target cell, so the state count is $N = k^2$. Each move goes in the intended direction or in one of the two perpendicular directions with probability $\frac{1}{3}$ each, giving a branching factor of $m \leq 3$ (cut down at the grid's border). A random 20% of the cells are holes, which are absorbing failure states. The nominal move distribution of each state is wrapped in an L_∞ or L_1 uncertainty ball whose radius is drawn per state from $U[0, 0.125]$ or $U[0, 0.25]$, respectively. Because the branching factor stays bounded by 3 regardless of N, this benchmark scales to much larger state spaces than the other two. On Frozen Lake we experiment only with the reachability objective (reaching the target cell).

Remark 44. *The bounds of Theorem 18 and Theorem 41 are worst-case upper bounds on the* number *of primitive queries against the uncertainty sets, and on all of our benchmarks the number actually issued is considerably smaller. The gap is not incidental, and it is not a property of the particular instances: it follows from the branching factor, and it applies to any family of linearly defined RMDPs whose uncertainty sets are described over few coordinates. We record it here because it is what reconciles the cubic bound of Theorem 41 with the measured solve times of Figure 1.*

Remark 45. *Figure 2 records how many iterations our algorithm and the baseline take on each benchmark. For both of them, the observed counts remain well below the theoretical worst case, which is exponential in the number of states. This agrees with what is commonly reported for policy iteration on stochastic games, where the iteration count is typically polynomial in practice even though no subexponential bound is known. The practical behaviour of our algorithms is thus considerably better than the worst-case analysis predicts.*