On the Complexity of Discounted Robust MDPs with L_p Uncertainty Sets

Ali Asadi
Institute of Science and Technology Austria
Klosterneuburg, Austria
ali.asadi@ista.ac.at

Krishnendu Chatterjee
Institute of Science and Technology Austria
Klosterneuburg, Austria
krishnendu.chatterjee@ista.ac.at

Alipasha Montaseri
EPFL
Lausanne, Switzerland
alipasha.montaseri@epfl.ch

Ali Shafiee
Institute of Science and Technology Austria
Klosterneuburg, Austria
ali.shafiee@ista.ac.at

Abstract

A basic model in sequential decision making is the Markov decision process (MDP), which is extended to Robust MDPs (RMDPs) by allowing uncertainty in transition probabilities and optimizing against the worst-case transition probabilities from the uncertainty sets. The class of (s, a)-rectangular RMDPs with L_p uncertainty sets provides a flexible and expressive model for such problems. We study this class of RMDPs with a discounted-sum cost criterion and a constant discount factor. The existence of an efficient algorithm for this class is a fundamental theoretical question in optimization and sequential decision making. Previous results only establish a strongly polynomial-time algorithm for L_∞ uncertainty sets. In this work, our main results are as follows: (a) we show that for any compact uncertainty set, the policy iteration algorithm for RMDPs is strongly polynomial with oracle access to solutions of Robust Markov chains (RMCs); (b) we present strongly polynomial-time bounds on the policy iteration algorithm for RMCs with L_1 and L_∞ uncertainty sets; and (c) we establish hardness results for RMCs with L_p uncertainty sets for integer p satisfying $1 < p < \infty$. Finally, motivated by our theoretical bounds, we present experimental results showing how fast policy iteration converges for RMDPs with L_1 and L_∞ uncertainty sets.

1 Introduction

Robust Markov Decision Processes. In a Markov decision process (MDP), an agent interacts with a stochastic environment of finite state space, where, given the current state and the choice of an action of the agent, the interaction with the environment updates the state probabilistically [Put94]. This is the basic model in sequential decision making, where the common assumption is that the transition function is known. In many practical scenarios, the transition function is not known, but rather constructed from data, which results in a level of uncertainty. This leads to the study of *Robust Markov decision processes* (RMDPs) [NEG05, Iye05]. In RMDPs, the assumption of MDPs is relaxed by considering that there is an uncertainty set containing the original transition function, which is chosen by an adversary. We consider (s, a)-rectangular RMDPs where the uncertainty sets are separate for each state-action pair. This class has been the standard in the literature [Iye05, NEG05].

Discounted-sum costs and values. The classical objective in MDPs and RMDPs for the agent is to minimize the discounted-sum cost criterion, where each state is assigned a cost and given a discount factor; the discounted cost of a run (infinite trajectory) is the discounted sum of the costs of states

Authors are listed in alphabetical order.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

appearing in the run. The value of an RMDP is the minimum expected discounted cost that can be guaranteed by the agent against all possible choices of the transition function from the uncertainty sets by the adversary.

Exact value computation. While approximation is important, exact value computation is relevant for two reasons. First, it is a fundamental algorithmic problem. For discounted MDPs with a fixed discount factor, [Ye11] gives a strongly polynomial algorithm for computing the exact value. Second, value iteration easily gives additive approximation guarantees for discounted MDPs and RMDPs. Multiplicative approximation can be harder. For example, in the stochastic games studied in [CMSS23], a multiplicative approximation can be used to compute the exact value. Exact computation provides a stronger guarantee than approximation.

L_p uncertainty sets and motivation. In this work, we focus on constant discount factors and primarily on L_p uncertainty sets, for $p \in \mathbb{N} \cup \{\infty\}$. The main motivation to study L_p uncertainty sets is as follows: for learning transition probabilities given the experimental samples, estimates of the uncertainty sets can be obtained using concentration bounds, which are then naturally approximated with L_p uncertainty sets [WOS$^+$03, BRPH21, GLD00, Iye05]. Thus, the study of RMDPs with various L_p uncertainty sets is a fundamental algorithmic problem that we address in this work.

Previous computational results. Linear programming solves MDPs with discounted cost in polynomial time [d'E63, Der72]. The existence of a strongly polynomial-time algorithm where the number of arithmetic operations is polynomial independent of the bit length is a fundamental theoretical question. The seminal work of [Ye11] presents the first strongly-polynomial time algorithm for constant discount factors for MDPs, and [HMZ13] extends it to stochastic games. Recently, a strongly-polynomial time bound was established for RMDPs with L_∞ uncertainty sets by [ACG$^+$26a]. While previous results exist for L_∞ uncertainty sets, a comprehensive algorithmic study of RMDPs with L_p uncertainty sets is missing, which is the focus of this work.

Our contributions. Our main contributions are as follows. Robust Markov chains (RMCs) are a special class of RMDPs where the agent does not have the choice of actions. First, for any compact uncertainty sets, we bound the number of policy iteration steps and calls to an RMC solver by a polynomial in the numbers of states and actions, for a fixed discount factor (Theorem 1). This gives a strongly polynomial algorithm for RMDPs whenever the RMC subproblems admit a strongly polynomial algorithm. Second, for L_1 and L_∞ uncertainty sets, we show that RMCs can be solved in strongly polynomial time. While this result exists for L_∞ sets [ACG$^+$26a], our analysis improves the previous result by a factor of the number of states, and for L_1 sets, our result is novel (Theorem 2). Third, we establish hardness results for RMCs with L_p uncertainty sets for $1 < p < \infty$, and also present upper bounds for L_p uncertainty sets (Theorem 3). Finally, motivated by these theoretical bounds, we complement our analysis with experimental results showing how fast policy iteration converges for RMDPs with L_1 and L_∞ uncertainty sets.

Technical contributions. The main technical contributions are threefold. First, we present a policy iteration algorithm for RMDPs that provides a reduction from RMDPs to RMCs. Second, we introduce a cumulative transition-probability potential with respect to the optimal distribution chosen by the adversary, which eliminates suboptimal regions of the uncertainty sets instead of their exponentially many vertices. With this potential, we obtain a new strongly-polynomial bound for L_1 sets and an improved bound for L_∞ sets. To the best of our knowledge, the previous potential function presented in [ACG$^+$26a] does not extend to L_1 sets. Third, for other L_p uncertainty sets with integer $1 < p < \infty$, we prove hardness via a reduction from p-ROOT-SUM, based on a new equivalence between p-ROOT-SUM and $\frac{p}{p-1}$-ROOT-SUM, and give an upper bound in the existential theory of the reals by encoding the adversary's optimization problem via KKT conditions.

Related works. The exact value computation for RMDPs with various L_p uncertainty sets is the focus of this work. Several related results have been established for the value approximation for more general models. For discounted-cost RMDPs, the value iteration algorithm is established concurrently in [NEG05, Iye05]. Moreover, policy iteration is also introduced in [Iye05] for value approximation. Another variant of policy iteration was established in [KS13] along with convergence guarantees. It is noteworthy that none of the above results present polynomial or strongly-polynomial bounds for the exact value computation. RMDPs have also been studied with other objectives. Robust control under temporal logic specifications was studied in [WTM12]. For long-run average reward, undiscounted total reward, and stochastic shortest path objectives, [MWW25] gives algorithms with

precision guarantees. For reachability and parity objectives, [ACG+26c] studies whether the agent can ensure the objective with probability one. For parity objectives with linearly defined uncertainty sets, [ACG+26b] studies exact value computation. Our work focuses on exact discounted-sum values.

2 Preliminaries

Notation. Let S be a set. We denote the set of all probability measures over S by $\Delta(S)$, and its power set by 2^S. We use $[n]$ to represent the set of all natural numbers from 1 to n. The sets of natural numbers and non-negative integer numbers are denoted by $\mathbb{N}$ and $\mathbb{N}_0$, respectively. For vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, we write $\boldsymbol{v} \geqslant \boldsymbol{u}$ to indicate that every component of $\boldsymbol{v}$ is greater than or equal to the corresponding component of $\boldsymbol{u}$. Moreover, we use $\text{size}(n) = \lceil \log_2(|n|+1) \rceil + 1$, and $\text{size}(\frac{p}{q}) = \text{size}(p) + \text{size}(q)$ as the bit-size of integer n and rational number $\frac{p}{q}$. For a set X, we write X^* for the set of all finite sequences over X, including the empty sequence, and X^+ for the set of all non-empty finite sequences.

Robust Markov Decision Processes. A Robust Markov Decision Process (RMDP) is a tuple $\mathcal{R} = (\mathcal{S}, c, \mathcal{A}, \mathit{Succ}, \mathcal{P})$, where the state space $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$ is a finite set of size n. The action space $\mathcal{A}$ is a finite set of size m. The function $c \colon \mathcal{S} \to \mathbb{R}$ gives the cost of visiting each state. We use state-dependent costs to simplify notation. Our results also extend to state-action costs $c(s,a)$. The function $\mathit{Succ} \colon \mathcal{S} \times \mathcal{A} \to 2^{\mathcal{S}}$ maps a state and an action to possible next states. The function $\mathcal{P} \colon \mathcal{S} \times \mathcal{A} \to 2^{\Delta(\mathcal{S})}$ gives a set of possible transition probabilities, where $\mathcal{P}(s,a) \subseteq \Delta(\mathit{Succ}(s,a))$. We assume that $\mathcal{P}(s,a)$ is a compact non-empty set.

Remark 1. *We assume (s,a)-rectangular uncertainty, which means that the adversary observes the chosen action before selecting a transition function from the uncertainty set. This assumption is standard in the literature [Iye05, NEG05]. Other classes of uncertainty are not the focus of our work.*

Dynamics. At first, the initial state is $s_0 \in \mathcal{S}$. At each time step $t \geqslant 0$, the agent first chooses an action $a_t \in \mathcal{A}$. Then, the adversary selects a transition probability distribution $\mathfrak{p}_t \in \mathcal{P}(s_t, a_t)$. The subsequent state s_{t+1} is drawn probabilistically from the distribution $\mathfrak{p}_t$. A run is an infinite sequence of states and actions $\omega \coloneqq (s_0, a_0, s_1, \ldots)$ where for $t \geqslant 0$, there exists a distribution $\mathfrak{p}_t \in \mathcal{P}(s_t, a_t)$ such that $\mathfrak{p}_t(s_{t+1}) > 0$. A history is a finite sequence of states and actions $h \coloneqq (s_0, a_0, \ldots, s_t)$.

Policies. A history-dependent agent policy is denoted as $\sigma \colon (\mathcal{S} \times \mathcal{A})^* \times \mathcal{S} \to \Delta(\mathcal{A})$. An agent policy is memoryless if it only depends on the current state, i.e., $\sigma \colon \mathcal{S} \to \Delta(\mathcal{A})$ and is pure if it prescribes deterministic actions, i.e., $\sigma \colon (\mathcal{S} \times \mathcal{A})^* \times \mathcal{S} \to \mathcal{A}$. We say an agent policy is positional if it is pure and memoryless. A history-dependent adversary policy is denoted as $\tau \colon (\mathcal{S} \times \mathcal{A})^+ \to \Delta(\Delta(\mathcal{S}))$ such that for every finite sequence of states and actions $h = (s_0, a_0, \ldots, s_t, a_t)$, we have $\tau(h) \in \Delta(\mathcal{P}(s_t, a_t))$. The notions of memoryless, pure, and positional adversary policies are defined analogously.

Probability Measures. For a history h, the cone of h is the set of runs where h is their prefix. Given an agent policy σ, an adversary policy τ, and an initial state s_0, the unique probability measure over Borel sets of runs is denoted by $\mathbb{P}^{\sigma,\tau}_{s_0}(.)$, which is defined by Carathéodory's extension theorem by extending the natural definition over cones of runs [Bil12]. The expectation with respect to this probability measure is denoted by $\mathbb{E}^{\sigma,\tau}_{s_0}(.)$.

Discounted Cost Criterion. A criterion is a measurable function that assigns a real number to every run. For a run $\omega = (s_0, a_0, \ldots)$, the discounted cost criterion is defined as $\mathit{Disc}(\omega) \coloneqq \sum_{t \in \mathbb{N}_0} \gamma^t c(s_t)$. The objective of the agent is to minimize the expected discounted cost criterion, while the adversary wants to maximize it.

Remark 2. *For the discounted cost criterion, positional policies for both agent and adversary are as powerful as history-dependent policies [Iye05, NEG05]. Therefore, we restrict our attention to positional policies in the rest of the paper.*

Values. The discounted value of a state s is defined as

$$\boldsymbol{v}^\star_{\mathcal{R}}(s) \coloneqq \inf_{\sigma} \sup_{\tau} \mathbb{E}^{\sigma,\tau}_{s}(\mathit{Disc}(\omega)) \,. \tag{1}$$

We refer to $\boldsymbol{v}^\star_{\mathcal{R}}$ as the optimal value vector. The superscript $\star$ denotes optimality and should not be confused with the Kleene star X^*. We drop the $\mathcal{R}$ label when it is clear from the context.

Optimal Policies. An agent policy $\sigma^\star$ is optimal if it guarantees the value for every state $s \in \mathcal{S}$, i.e.,

$$\sup_{\tau} \mathbb{E}_s^{\sigma^\star,\tau}(\mathit{Disc}(\omega)) = \boldsymbol{v}^\star_{\mathcal{R}}(s)\,.$$

The notion of optimal adversary policy is defined analogously.

Robust MCs. A Robust Markov Chain (RMC) is a special case of an RMDP that has one possible action for the agent; hence, we can drop the action set. An RMC is $\mathcal{M} = (\mathcal{S}, c, \mathit{Succ}, \mathcal{P})$. The value of RMC is denoted by $\boldsymbol{v}^\star_{\mathcal{M}}(s)$. We drop the $\mathcal{M}$ label when it is clear from the context.

Markov Chains. A Markov Chain (MC) is a special case of an RMC where for every state s, $\mathcal{P}(s)$ is a singleton. A MC is $M = (\mathcal{S}, c, P)$, where $P\colon \mathcal{S} \to \Delta(\mathcal{S})$ are the fixed transition probabilities. The value of an MC is denoted by $\boldsymbol{v}^\star_M(s)$. We drop the M label when it is clear from the context.

Problem Statement. We consider the following computational problems.

Problem DiscRMDP**.** Given an RMDP $\mathcal{R}$ with a constant discount factor $\gamma \in [0, 1)$, compute the value vector $\boldsymbol{v}^\star_{\mathcal{R}} \in \mathbb{R}^{|\mathcal{S}|}$, optimal agent policy $\sigma^\star\colon \mathcal{S} \to \mathcal{A}$, and optimal adversary policy $\tau^\star\colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$.

Problem DiscRMC**.** Given an RMC $\mathcal{M}$ with a constant discount factor $\gamma \in [0, 1)$, compute the value vector $\boldsymbol{v}^\star_{\mathcal{M}} \in \mathbb{R}^{|\mathcal{S}|}$ and optimal adversary policy $\tau^\star\colon \mathcal{S} \to \Delta(\mathcal{S})$.

The decision version of the above problems is, given an initial state s and a threshold λ, decide whether $\boldsymbol{v}^\star(s) \geqslant \lambda$.

3 Overview of Results

In this section, we give an overview of our results. First, we recall some complexity classes on which our results rely and then state our main theoretical contributions.

3.1 Complexity Definitions

Strongly Polynomial Algorithms. An algorithm is strongly polynomial if its number of arithmetic operations is polynomial in the input dimensions, independent of the bit-length of the numerical data, and the bit-length of its intermediate values is polynomial in the input's bit-length. Recall that an algorithm is said to run in polynomial time if its overall execution time is bounded by a polynomial in the input's bit-length. Note that strongly-polynomial time implies polynomial time, but not vice versa. Thus, strongly-polynomial time gives stronger efficiency guarantees.

p-ROOT-SUM Complexity. The p-ROOT-SUM problem asks whether, given a sequence of positive integers a_i (for $i \in [n]$) and an integer threshold α, the following inequality holds:

$$\sum_{i=1}^{n} a_i^{\frac{1}{p}} \geqslant \alpha$$

For $p = 2$, this corresponds to the classic Square-Root-Sum problem, which is a major and long-standing open problem in computational geometry since 1976 [GGJ76]. This problem is known to be in PSPACE, but neither polynomial-time algorithms are known nor even NP upper bounds are established [ABKPM09]. We define the p-ROOT-SUM complexity class to contain all computational problems that are polynomial-time reducible to the p-ROOT-SUM problem.

Existential theory of the reals. A sentence in the *existential theory of the reals* (ETR) is of the form

$$\exists x_1 \ldots \exists x_k\ \varphi(x_1, \ldots, x_k)$$

where $\varphi(x_1, \ldots, x_k)$ is a *quantifier-free* formula in the language of ordered fields over the reals $\mathbb{R}$. More concretely, atomic sub-formulas are $f(x_1, \ldots, x_k) = 0$ or $f(x_1, \ldots, x_k) < 0$, where $f \in \mathbb{Q}[x_1, \ldots, x_k]$ is a polynomial over $\mathbb{Q}$, and arbitrary formulas are built by Boolean connectives, e.g., conjunction, disjunction, and negation. We denote by $\exists\mathbb{R}$ the complexity class of problems that can be reduced in polynomial time to the problem of deciding the validity of an ETR sentence. This class naturally contains NP and is known to be contained within PSPACE [Can88].

3.2 Main Results

In this section, we state our main results. The main algorithm to solve RMDPs and RMCs is the policy iteration algorithm (Algorithm 1 and Algorithm 2): In RMDPs, this algorithm iterates over positional strategies of the agent, and in RMCs, this algorithm iterates over positional strategies of the adversary. The key goal is to obtain bounds on the number of iterations. Our results are as follows:

Algorithm 1 Policy Iteration For RMDPs (`RMDP-PI`)

Input: RMDP $\mathcal{R} = (\mathcal{S}, c, \mathcal{A}, \mathit{Succ}, \mathcal{P})$, discount factor γ
Output: The optimal policy $\sigma^\star$
$\sigma^0 \leftarrow$ arbitrary agent policy
$t \leftarrow 0$
repeat
  $\tau^t \leftarrow \texttt{RMC-Oracle}(\mathcal{R}^{\sigma^t}, \gamma)$ ▷ Policy Evaluation: Invoking the RMC Oracle
  $\boldsymbol{v}^t \leftarrow (\mathbb{I} - \gamma P^{\sigma^t, \tau^t})^{-1} c$
  $t \leftarrow t + 1$
  for all $s \in \mathcal{S}$ **do** ▷ Policy Improvement
    $\sigma^t(s) \leftarrow \operatorname{argmin}_{a \in \mathcal{A}} \max_{\mathfrak{p} \in \mathcal{P}(s,a)} \{c(s) + \gamma \mathfrak{p}^\top \boldsymbol{v}^{t-1}\}$
until $\sigma^t = \sigma^{t-1}$
return σ^t

Tie-breaking Rule: Choose the minimizing action with the smallest index.

Algorithm 2 Policy Iteration For RMCs (`RMC-PI`)

Input: RMC $\mathcal{M} = (\mathcal{S}, c, \mathit{Succ}, \mathcal{P})$, discount factor γ
Output: The optimal adversary policy $\tau^\star$
$\tau^0 \leftarrow$ arbitrary adversary policy
$t \leftarrow 0$
repeat
  $\boldsymbol{v}^t \leftarrow (\mathbb{I} - \gamma P^{\tau^t})^{-1} c$ ▷ Policy Evaluation
  $t \leftarrow t + 1$
  for all $s \in \mathcal{S}$ **do** ▷ Policy Improvement
    $\tau^t(s) \leftarrow \operatorname{argmax}_{\mathfrak{p} \in \mathcal{P}(s)} \{c(s) + \gamma \mathfrak{p}^\top \boldsymbol{v}^{t-1}\}$
until $\tau^t = \tau^{t-1}$
return τ^t

Tie-breaking Rule: For L_1 and L_∞ sets, use Algorithms 3 and 4, respectively, with ties between equal-value states broken by increasing state index.

Theorem 1. *The policy iteration algorithm for Problem* DiscRMDP*, Algorithm 1, requires* $\mathcal{O}(nm \frac{\log(1-\gamma)}{\log \gamma})$ *iterations and solutions of Problem* DiscRMC *for any compact uncertainty sets. In particular, if Problem* DiscRMC *admits a strongly polynomial-time algorithm, then Problem* DiscRMDP *also admits a strongly polynomial-time algorithm.*

Our following results for RMCs are related to L_p uncertainty sets, which are defined as follows:
Assumption 1. *The uncertainty sets are defined as L_p balls, for $p \in \mathbb{N} \cup \{\infty\}$, around a nominal transition function $\hat{\mathfrak{p}}_s \in \Delta(\mathit{Succ}(s))$. For RMCs, this means $\mathcal{P}(s) = \{\mathfrak{p} \in \Delta(\mathit{Succ}(s)) \mid \|\mathfrak{p} - \hat{\mathfrak{p}}_s\|_p \leqslant \delta(s)\}$ for a nominal transition $\hat{\mathfrak{p}}_s$ and radius $\delta(s)$. This definition extends naturally to RMDPs via $\hat{\mathfrak{p}}_{s,a}$ and $\delta(s,a)$ for each state-action pair.*
Theorem 2. *The policy iteration algorithm for Problem* DiscRMC*, Algorithm 2, for L_1 and L_∞ uncertainty sets requires $\mathcal{O}(n^3 \log n \log\left(\frac{1-\gamma}{n}\right) / \log \gamma)$ (i.e., strongly polynomial for constant discount factors) iterations.*
Theorem 3. *For any integer $p > 1$, the following statements hold:*

1. *The decision version of Problem* DiscRMC *with L_p uncertainty sets is p-ROOT-SUM-hard;*
2. *The decision version of Problem* DiscRMC *with L_p uncertainty sets is in $\exists\mathbb{R}$.*

Uncertainty sets	PI iterations	Reference
RMDPs, Algorithm 1 with an RMC solver		
L_∞	$O(nm)$	[ACG$^+$26a]
Compact	$\boldsymbol{O(nm)}$	**Theorem 1**
RMCs, Algorithm 2		
L_∞	$O(n^4 \log^2 n)$	[ACG$^+$26a]
$\boldsymbol{L_\infty}$	$\boldsymbol{O(n^3 \log^2 n)}$	**Theorem 2**
$\boldsymbol{L_1}$	$\boldsymbol{O(n^3 \log^2 n)}$	**Theorem 2**

Table 1: Comparison of policy iteration bounds for a fixed discount factor $0 < \gamma < 1$. Here, n is the number of states and m is the number of actions. RMDPs use (s, a)-rectangular uncertainty sets. The RMDP rows count outer iterations with access to an RMC solver. The RMC rows count iterations of the robust policy-evaluation algorithm. Thus, the RMDP bounds alone are not full running-time bounds. Our results are shown in bold.

Significance. We now discuss the significance of our results, which present a comprehensive picture of RMDPs and RMCs with L_p uncertainty sets and a constant discount factor. Our first result bounds the number of policy iteration steps and calls to an RMC solver for arbitrary compact uncertainty sets. For a fixed discount factor, these bounds are polynomial in the numbers of states and actions. The second result presents strongly polynomial time bounds for RMCs with L_1 and L_∞ uncertainty sets. Although the strongly-polynomial-time bounds for the L_∞ case were previously established by [ACG$^+$26a], our analysis yields a tighter bound by a factor of n; and the result for L_1 uncertainty sets is novel. Table 1 compares our policy iteration bounds with the previous bounds for L_∞ uncertainty sets. Finally, the first item of the third result shows computational hardness for L_p uncertainty sets with $1 < p < \infty$ (i.e., other than L_1 and L_∞). The last item of the third result presents upper bounds for RMCs with L_p uncertainty sets. Together, these results complete the study of efficient algorithms for RMDPs and RMCs with a constant discount factor with L_p uncertainty sets. In the following sections, we provide the main proof ideas of all these results.

4 Overview of Theorem 1: Strongly Polynomial PI for RMDPs via RMC Oracles

In this section, we provide a brief overview of the proof of Theorem 1. The theorem states that for any compact uncertainty set and a constant discount factor, the policy iteration algorithm for Problem DiscRMDP (Algorithm 1) terminates after $\mathcal{O}\left(nm\frac{\log(1-\gamma)}{\log\gamma}\right)$ iterations given oracle access to the solution of Problem DiscRMC. The full proof is presented in Appendix A. The key insight is that, after fixing an agent policy σ, we obtain an RMC, and consequently, the policy-evaluation step can be handled by an RMC oracle call. Therefore, our main task is to show that the number of iterations is polynomial. We organize the proof into four main steps. We follow the action-elimination approach of [Ye11]. The main difference is the potential function. For RMDPs, the potential function must account for the adversary's choice of a transition distribution for each action. Our potential function extends this approach to compact uncertainty sets.

Robust Bellman Optimality Operator. We first define the robust Bellman optimality operator as

$$(\mathcal{T}\boldsymbol{v})_s := c(s) + \gamma \min_{a\in\mathcal{A}} \max_{\mathfrak{p}\in\mathcal{P}(s,a)} \mathfrak{p}^\top \boldsymbol{v}.$$

Since $\mathcal{A}$ is finite and each uncertainty set $\mathcal{P}(s, a)$ is compact, the minimum and maximum are attained. The operator $\mathcal{T}$ is a γ-contraction in the infinity norm and therefore has a unique fixed point $\boldsymbol{v}^\star$, which is the optimal value vector. Moreover, for every fixed agent policy σ, the induced value vector $\boldsymbol{v}^\sigma$ satisfies $\mathcal{T}\boldsymbol{v}^\sigma \preccurlyeq \boldsymbol{v}^\sigma$, and consequently $\boldsymbol{v}^\star \preccurlyeq \boldsymbol{v}^\sigma$. Using this contraction property, we show that `RMDP-PI` generates a monotonically non-increasing sequence of value vectors, i.e., $\boldsymbol{v}^{t+1} \preccurlyeq \boldsymbol{v}^t$. Moreover, the value error decreases exponentially, i.e.,

$$\left\|\boldsymbol{v}^t - \boldsymbol{v}^\star\right\|_\infty \leqslant \gamma^t \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty,$$

which gives a convergence rate that is later used to bound the number of iterations.

Potential Function. We next define a potential function that quantifies the cost of choosing an action a instead of an optimal action at state s, where both choices are evaluated against the optimal value vector. Formally,

$$f(s,a) \coloneqq \max_{\mathfrak{p}\in\mathcal{P}(s,a)} \mathfrak{p}^\top \boldsymbol{v}^\star - \max_{\mathfrak{p}\in\mathcal{P}(s,\sigma^\star(s))} \mathfrak{p}^\top \boldsymbol{v}^\star.$$

Intuitively, $f(s,a)$ measures the continuation cost incurred by choosing a at state s instead of $\sigma^\star(s)$.

For any agent policy σ, the potential gives two complementary bounds. First, if $\sigma(s) = a$, then the value error at state s satisfies

$$\boldsymbol{v}^\sigma_s - \boldsymbol{v}^\star_s \geqslant \gamma f(s,a).$$

Second, if $\hat{s} \in \operatorname{argmax}_{s\in\mathcal{S}} f(s,\sigma(s))$, then the value error is bounded by

$$\|\boldsymbol{v}^\sigma - \boldsymbol{v}^\star\|_\infty \leqslant \frac{\gamma}{1-\gamma} f(\hat{s},\sigma(\hat{s})).$$

These inequalities establish lower and upper bounds on the difference between an arbitrary policy's value vector and the optimal value vector, and are later used to prove that Algorithm 1 terminates in a polynomial number of iterations.

Elimination of Actions with Maximum Potential. We combine the lower and upper bounds above with the exponential convergence rate. Let σ^l be a non-optimal policy at iteration l, and let $\hat{s} \in \operatorname{argmax}_{s\in\mathcal{S}} f(s,\sigma^l(s))$. If a later policy σ^k selects the same action at $\hat{s}$ after at least $L \coloneqq \lfloor \log_\gamma(1-\gamma) \rfloor + 1$ iterations, then the lower bound on the potential function contradicts the exponential decay of the value error. Hence, the state-action pair $(\hat{s},\sigma^l(\hat{s}))$ is permanently removed after L iterations. There are at most nm state-action pairs. Since at least one state-action pair is eliminated every L iterations until an optimal policy is reached, the fixed tie-breaking rule ensures termination within $\mathcal{O}(nmL)$ iterations.

Reduction to One RMC Oracle Call. Finally, we show that the policy-improvement step can be implemented using an RMC oracle. For every state-action pair (s,a), the algorithm needs to compute

$$c(s) + \gamma \max_{\mathfrak{p}\in\mathcal{P}(s,a)} \mathfrak{p}^\top \boldsymbol{v}^{t-1}.$$

Instead of solving these maximization problems separately, we construct a batch RMC $\mathcal{M}_{\text{batch}}$. This RMC has one transient state $z_{s,a}$ for every pair $(s,a) \in \mathcal{S}\times\mathcal{A}$ and one absorbing state x_s for every original state $s \in \mathcal{S}$. The absorbing state x_s is assigned value $\boldsymbol{v}^{t-1}_s$, and the uncertainty set at $z_{s,a}$ is exactly $\mathcal{P}(s,a)$. Therefore, we have $\boldsymbol{v}^\star_{\text{batch}}(z_{s,a}) = c(s) + \gamma \max_{\mathfrak{p}\in\mathcal{P}(s,a)} \mathfrak{p}^\top \boldsymbol{v}^{t-1}$. Thus, all action-improvement quantities can be computed with one RMC oracle call on an RMC with $\mathcal{O}(nm)$ states. The construction reuses the original uncertainty sets, with transitions redirected to the corresponding absorbing states. Together with the policy-evaluation call, each iteration uses two RMC oracle calls. Our bound counts arithmetic operations and oracle calls, assuming exact arithmetic. Intermediate values, including the costs of the batch RMC, may be irrational. Therefore, this bound alone does not guarantee polynomial running time in the standard bit model. However, if the RMC oracle is a strongly polynomial-time algorithm, then Algorithm 1 is a strongly polynomial-time algorithm for Problem DiscRMDP.

5 Overview of Theorem 2: Complexity of RMC with L_1, L_∞ Uncertainty Sets

In this section, we provide a brief overview of the proof of Theorem 2. The theorem states that for L_1 and L_∞ uncertainty sets and a constant discount factor, the policy iteration algorithm for Problem DiscRMC (Algorithm 2) terminates in a polynomial number of iterations. Since each iteration runs in strongly polynomial time, this yields a strongly polynomial algorithm under L_1 and L_∞ uncertainty sets with a constant discount factor. Combined with Theorem 1, this completes the proof that RMDPs with L_1 or L_∞ uncertainty sets and a constant discount factor are solvable in strongly polynomial time. The full proof is presented in Appendix B.

The challenge of analyzing policy iteration for RMCs, in contrast to standard MDPs, is that the adversary's action space is the continuous uncertainty set $\mathcal{P}(s)$. Although the worst-case distribution lies on the boundary of the L_1 or L_∞ ball, the boundary can have exponentially many vertices (e.g. L_1 ball of dimension n has up to 2^n vertices). Thus, a naive enumeration over boundary points fails. Our argument bypasses this by exploiting the algebraic structure of the distributions returned by the policy improvement step and combining it with a number-theoretic combinatorial bound. We organize the proof into four main steps.

Policy Improvement and Structural Property. The policy improvement step of Algorithm 2 requires solving $\max_{\mathfrak{p}\in\mathcal{P}(s)} \mathfrak{p}^\top \boldsymbol{v}$. For L_1, the homotopy method of [HPW21] sorts the successor states by descending value and greedily shifts the available budget from the lowest-value tail to the highest-value head. For L_∞, the algorithm by [BPH21] performs an analogous transfer subject to the entry-wise constraint $|\mathfrak{p}_{s'} - \hat{\mathfrak{p}}_{s,s'}| \leqslant \delta_s$. Both procedures run in linear time after sorting the values, and expose a clean structural property: every coordinate of the resulting distribution is either equal to the nominal $\hat{\mathfrak{p}}_{s,s'}$, shifted from it by a fixed value ($\pm\delta_s$ for L_∞ and $\pm\delta_s/2$ for L_1), or saturated at 0 or 1, with at most one "incomplete" coordinate. This property is used in the convergence analysis.

Potential Function and Sandwich Bounds. Fix a state s and order its successors $s_1, \ldots, s_{|\mathit{Succ}(s)|}$ in descending order of $\boldsymbol{v}^\star$, breaking ties by state index. Let $F^\star(s,i)$ be the cumulative transition probability of the optimal distribution returned by the corresponding greedy procedure with this order. For an adversary policy τ, define the cumulative transition $F^\tau(s,i) \coloneqq \sum_{j=1}^{i} \mathfrak{p}^\tau_{s,s_j}$. A first observation is that $F^\star(s,i) \geqslant F^\tau(s,i)$ for every feasible τ and every pair of state and index (s,i). We then define the potential function $f_\tau(s,i)$ which is guaranteed to be non-negative:

$$f_\tau(s,i) \;\coloneqq\; \big(F^\star(s,i) - F^\tau(s,i)\big)\big(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}}\big).$$

Using the Bellman equations and summation by parts, we obtain a *lower bound* on the per-state value gap, $\boldsymbol{v}^\star_s - \boldsymbol{v}^\tau_s \geqslant \gamma f_\tau(s,i)$. A complementary *upper bound* is shown to bound the global value error by the worst-case potential, namely

$$\|\boldsymbol{v}^\star - \boldsymbol{v}^\tau\|_\infty \;\leqslant\; \frac{\gamma n}{1-\gamma}\,\hat{f}_\tau, \qquad \text{where } \hat{f}_\tau \coloneqq \max_{s\in\mathcal{S},\, i} f_\tau(s,i).$$

These inequalities sandwich the global value error by a single scalar quantity associated with the worst state-index pair.

Halving of the Cumulative Gap. Combining the sandwich bounds with the standard exponential convergence $\|\boldsymbol{v}^\star - \boldsymbol{v}^t\|_\infty \leqslant \gamma^t \|\boldsymbol{v}^\star - \boldsymbol{v}^0\|_\infty$ yields the key halving lemma: if (s,i) maximizes the potential at iteration t, then within $L \coloneqq \lceil \log_\gamma\!\big(\frac{1-\gamma}{2n}\big) \rceil$ subsequent iterations the gap $F^\star(s,i) - F^\tau(s,i)$ must shrink by at least a factor of two; otherwise, the lower bound at iteration l and the upper bound at iteration t would contradict the exponential decay rate of policy iteration.

Combinatorial Bound on the Number of Halvings. The structural property of policy iteration (the first step) implies that for every state s and every policy τ visited by the algorithm, the cumulative gap $F^\star(s,i) - F^\tau(s,i)$ is a signed sum of $\mathcal{O}(n)$ numbers (the nominal probabilities $\hat{\mathfrak{p}}_{s,j}$ together with 1 and multiples of δ_s), with constant integer coefficients. A combinatorial lemma [ACG+26a] shows that any such set of signed sums takes $\mathcal{O}(n \log n)$ distinct binary scales $\lfloor \log_2 |y| \rfloor$. Since every halving drops the gap to a smaller binary scale, each of the at most n^2 state-index pairs (s,i) can have at most $\mathcal{O}(n \log n)$ halvings. Combined with the L-iteration cost per halving from the above step, this bounds the total number of policy iterations by $\mathcal{O}(n^3 \log n \cdot L)$, which gives the polynomial bound of Theorem 2. The same argument improves the previously known bound of [ACG+26a] for the L_∞ case by a factor of $|\mathcal{S}|$, and gives the first polynomial bound in the L_1 case.

6 Overview of Theorem 3: Complexity of RMC with L_p Uncertainty Sets

In this section, we provide a brief overview of the proof for Theorem 3. This theorem states that for any integer $p > 1$, solving the decision version of the problem DiscRMC with L_p uncertainty sets is p-ROOT-SUM-hard, and the problem belongs to $\exists\mathbb{R}$ complexity class.

We break down the proof into three steps: (1) Establishing the equivalence between the complexity classes p-ROOT-SUM and $\frac{p}{p-1}$-ROOT-SUM, (2) Proving the problem's hardness via a reduction from the $\frac{p}{p-1}$-ROOT-SUM problem to the decision problem, and (3) Showing the problem belongs to the $\exists\mathbb{R}$ by encoding it as an $\exists\mathbb{R}$ formula. The full proof of Theorem 3 is detailed in Appendix C.

1. Equivalence of ROOT-SUM Problems. Initially, we simplify the underlying complexity classes. We prove that the p-ROOT-SUM problem is polynomial-time equivalent to the $\frac{p}{p-1}$-ROOT-SUM problem. We achieve this through a sequence of reductions. First, we show that the p-ROOT-SUM problem is reducible to the kp-ROOT-SUM problem for any constant integer $k > 1$. Next, to handle fractional roots, we introduce a "Greedy p-th Power Decomposition" algorithm, which breaks down any positive integer n into a sum of p-th powers using $\mathcal{O}(\log \log n)$ terms in polynomial time. Using

this decomposition, we prove that the p-ROOT-SUM problem can be reduced to the $\frac{p}{q}$-ROOT-SUM problem for any integer q where $\gcd(p, q) = 1$. Combining these steps shows that the denominator in the fractional root does not change the complexity class, establishing the polynomial-time equivalence between the p-ROOT-SUM and $\frac{p}{p-1}$-ROOT-SUM problems.

2. Proving Hardness via Reduction. We prove that the decision version of the problem DiscRMC is at least as hard as solving the $\frac{p}{p-1}$-ROOT-SUM problem. We do this by constructing a three-layered RMC that forces the RMC's Bellman equations to mimic the ROOT-SUM inequality. In an RMC, the adversary chooses transition probabilities from an L_p uncertainty set to maximize the expected cost. Formally, this corresponds to finding the maximum inner product between the transition probability vector and the value vector of successor states. According to Hölder's inequality [Fol99], maximizing an inner product under an L_p constraint evaluates the dual norm, which is the $L_{\frac{p}{p-1}}$ norm. This dual norm computes the fractional powers for the $\frac{p}{p-1}$-ROOT-SUM problem. By this property, we construct the three-layered RMC as follows:

- *Absorbing States:* At the base layer, we create absorbing states with fixed costs, which are equal to the decomposition of the input integers a_i by the "Greedy p-th Power Decomposition" algorithm.
- *Transient States:* The middle layer has transient states for each a_i, which are connected to corresponding absorbing states. Because the adversary picks worst-case transitions from an L_p ball, the value of each transient state is proportional to the $L_{\frac{p}{p-1}}$ norm of the corresponding absorbing state values, which is proportional to $a_i^{\frac{p-1}{p}}$.
- *Initial State:* The top layer is a single initial state that transitions uniformly to all transient states. Its value is the average of the transient states' values, which is $\sum_{i=1}^{n} a_i^{\frac{p-1}{p}}$.

By structuring the RMC this way, the adversary's optimal choices compute the fractional roots for us. Finally, deciding whether the value of the initial state meets the threshold λ is equal to deciding if the sum meets the threshold α. Thus, the decision problem is p-ROOT-SUM-hard.

3. ETR Upper Bound. Finally, we prove that the decision version of problem DiscRMC is in $\exists\mathbb{R}$. We show this by modeling the main problem (computing exact values and optimal strategy) as a convex optimization problem. Because this optimization problem satisfies Slater's condition, we characterize its optimal solution using the Karush-Kuhn-Tucker (KKT) conditions [BV04]. Because p is a constant integer, the derivative of these constraints produces integer powers. This ensures that the optimization problem can be expressed by polynomial equations and inequalities. To complete the proof, we combine these polynomial KKT conditions with the Bellman equations to create a system of polynomial equations and inequalities. By definition, finding a valid solution to a system of polynomial equations and inequalities over the real numbers is in $\exists\mathbb{R}$.

7 Experiments

In this section, we empirically evaluate `RMDP-PI` (Algorithm 1) across five benchmarks under L_1 and L_∞ uncertainty sets, motivated by the strongly polynomial theoretical bounds we established for these specific cases. Our primary goal is to observe how many iterations Algorithm 1 takes to converge on these benchmarks. Additionally, we analyze the total number of iterations `RMC-PI` (Algorithm 2) performed over each iteration of Algorithm 1.

Implementation Details. All experiments are implemented in Python 3.11.14 using exact calculations. Specifically, every transition probability, reward, discount factor, and uncertainty radius is stored and processed using Python's built-in `fractions.Fraction` class. The linear system that arises during policy evaluation is solved exactly via a Bareiss fraction-free Gaussian elimination procedure implemented directly on top of arbitrary-precision rationals from the `gmpy2` library. The inner L_1 and L_∞ oracles (Algorithm 3 and Algorithm 4) are procedures that sort values and shift probability masses, involving only rational subtractions and comparisons. All experiments were executed on a single machine with an Intel Core Ultra 5 225U processor and 16GB of RAM.

Benchmarks. We evaluate `RMDP-PI` (Algorithm 1) on five environments, scaling the state-space size up to $n = 256$. Four are standard: **Gridworld** [SB18], a stochastic grid-navigation task with an absorbing goal and a trap; **Inventory Management** [Put94], where the agent chooses order quantities against stochastic demand; **Machine Replacement** [WKR13], in which a degrading machine can be

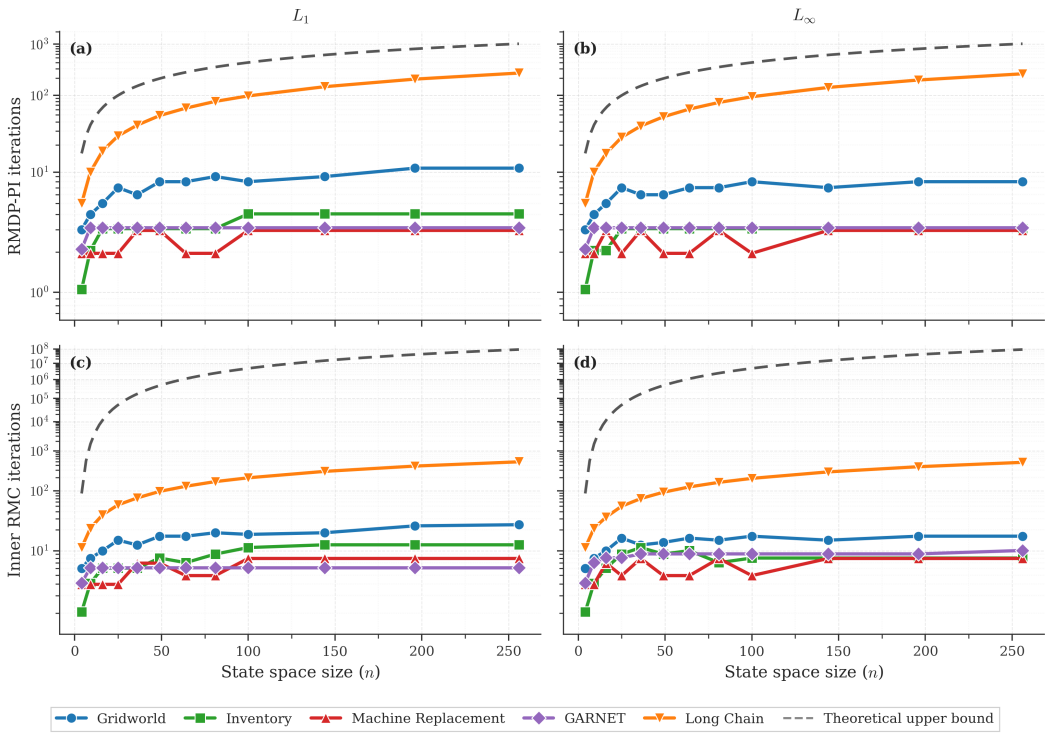


Figure 1: Outer `RMDP-PI` (a, b) and total inner `RMC-PI` (c, d) iterations versus state-space size n on five benchmarks, under L_1 (left) and L_∞ (right) uncertainty sets, with $\gamma = 0.5$, $\delta = 0.05$. Dashed line: theoretical upper bound.

operated, repaired, or replaced; and **GARNET** [AMT95], an unstructured random-MDP baseline. The fifth, **Long Chain**, is a construction we introduce that forces $\Theta(n)$ outer policy-iteration steps. The complete dynamics for each environment are given in Appendix D.

Results. Figure 1 reports the outer `RMDP-PI` and total inner `RMC-PI` iteration counts at $\gamma = 0.5$, $\delta = 0.05$, under both L_1 and L_∞ uncertainty sets; similar plots for different values of γ, δ are presented in Appendix D (Figures 2 to 4). We show $\gamma = 0.5$ in the main text for space reasons. Across the tested discount factors, we observe similar trends in the iteration counts, while the theoretical bounds increase as γ approaches 1.

Across both norms, iteration counts on the four standard benchmarks (Gridworld, Inventory, Machine Replacement, GARNET) grow very slowly with n and stay in the low double digits even at $n = 256$, while Long Chain shows the $\Theta(n)$ growth it was designed to produce. All curves sit several orders of magnitude below the worst-case upper bound (dashed grey), and the gap only widens as $\gamma \to 1$.

8 Conclusion and Limitations

Our work presents a comprehensive picture of algorithmic studies of RMDPs and RMCs with a constant discount factor. There are some limitations of this work, which naturally give rise to future directions. First, this work focuses on L_p uncertainty sets, and generalization to other sets is an interesting direction. Second, for general L_p uncertainty sets, there is a gap between the lower and upper bound (p-root-sum hardness and ETR upper bound), and a tighter complexity result is another interesting direction. Third, for L_1 and L_∞ uncertainty sets, our work gives strongly polynomial upper bounds for policy iteration. However, lower bounds have not yet been studied in the context of RMDPs.

Acknowledgments

We thank Negar Babashah for helpful discussions. This research was partially supported by Austrian Science Fund (FWF) 10.55776/COE12 and by the ERC CoG 863818 (ForM-SMArt) grant.

References

[ABKPM09] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009.

[ACG+26a] Ali Asadi, Krishnendu Chatterjee, Ehsan Goharshady, Mehrdad Karrabi, Alipasha Montaseri, and Carlo Pagano. Strongly polynomial time complexity of policy iteration for l_∞ robust mdps. *arXiv preprint arXiv:2601.23229*, 2026.

[ACG+26b] Ali Asadi, Krishnendu Chatterjee, Ehsan Kafshdar Goharshady, Mehrdad Karrabi, Alipasha Montaseri, and Ali Shafiee. Quantitative analysis of ω-regular robust MDPs, 2026.

[ACG+26c] Ali Asadi, Krishnendu Chatterjee, Ehsan Kafshdar Goharshady, Mehrdad Karrabi, and Ali Shafiee. Qualitative analysis of ω-regular objectives on robust MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 36137–36145, 2026.

[AMT95] Thomas Welsh Archibald, Ken IM McKinnon, and Lyn C Thomas. On the generation of markov decision processes. *Journal of the Operational Research Society*, 46(3):354–361, 1995.

[Bil12] Patrick Billingsley. *Probability and Measure*. Wiley, 2012.

[BPH21] Bahram Behzadian, Marek Petrik, and Chin Pang Ho. Fast algorithms for l_∞-constrained s-rectangular robust mdps. *Advances in Neural Information Processing Systems*, 34:25982–25992, 2021.

[BRPH21] Bahram Behzadian, Reazul Hasan Russel, Marek Petrik, and Chin Pang Ho. Optimizing percentile criterion using robust mdps. In *AISTATS*, Proceedings of Machine Learning Research, pages 1009–1017. PMLR, 2021.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Can88] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing STOC*, STOC '88, page 460–467. Association for Computing Machinery, 1988.

[CMSS23] Krishnendu Chatterjee, Tobias Meggendorfer, Raimundo Saona, and Jakub Svoboda. Faster algorithm for turn-based stochastic games with bounded treewidth. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4590–4605. SIAM, 2023.

[d'E63] Francois d'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98–108, 1963.

[Der72] Cyrus Derman. *Finite State Markovian Decision Processes*. Academic Press, 1972.

[Fol99] Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 1999.

[GGJ76] M. R. Garey, Ronald L. Graham, and David S. Johnson. Some np-complete geometric problems. In *STOC*, pages 10–22. ACM, 1976.

[GLD00] Robert Givan, Sonia M. Leach, and Thomas L. Dean. Bounded-parameter markov decision processes. *Artif. Intell.*, 122(1-2):71–109, 2000.

[HMZ13] Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

[HPW21] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Partial policy iteration for l1-robust markov decision processes. *Journal of Machine Learning Research*, 22(275):1–46, 2021.

[HS00] Marc Hindry and Joseph H. Silverman. *Diophantine Geometry: An Introduction*. Springer, 2000.

[Iye05] Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[KS13] David L. Kaufman and Andrew J. Schaefer. Robust modified policy iteration. *INFORMS J. Comput.*, 25(3):396–410, 2013.

[MWW25] Tobias Meggendorfer, Maximilian Weininger, and Patrick Wienhöft. Solving robust Markov Decision Processes: Generic, reliable, efficient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26631–26641, 2025.

[NEG05] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[WKR13] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[WOS+03] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, page 125, 2003.

[WTM12] Eric M. Wolff, Ufuk Topcu, and Richard M. Murray. Robust control of uncertain Markov Decision Processes with temporal logic specifications. In *51st IEEE Conference on Decision and Control*, pages 3372–3379. IEEE, 2012.

[Ye11] Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

A Proof of Theorem 1: Strongly Polynomial PI for RMDPs via RMC Oracles

In this section, we show that policy iteration finds an optimal policy and value vector using a polynomial number of iterations and RMC oracle calls, for compact uncertainty sets and a fixed discount factor. We count exact arithmetic operations and oracle calls. The oracle must accept the possibly irrational costs of the constructed RMCs. We detail the procedure in Algorithm 1.

Let $\mathcal{R} = (\mathcal{S}, c, \mathcal{A}, \mathit{Succ}, \mathcal{P})$ denote the input RMDP. We use $\mathcal{R}^\sigma$ to denote the RMC induced by fixing the strategy σ for the agent. We first define the Bellman operator and state its properties in Lemma 1. These properties guarantee the exponential convergence rate of `RMDP-PI`, as shown in Lemma 3. Next, we introduce a potential function. We use this function in Lemma 4 to establish lower and upper bounds on the value error of the policies. Finally, Lemma 7 applies these bounds to prove that `RMDP-PI` terminates in a polynomial number of iterations.

A.1 Robust Bellman Optimality Operator

We define the robust Bellman optimality operator $\mathcal{T}$ for the RMDP $\mathcal{R}$ as follows:

$$(\mathcal{T}\boldsymbol{v})_s = \mathbf{c}_s + \gamma \inf_{a \in \mathcal{A}} \sup_{\mathfrak{p} \in \mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v}.$$

Note that because the action space $\mathcal{A}$ is finite, the infimum is always achieved. Thus, we can use $\inf_{a\in\mathcal{A}}$ and $\min_{a\in\mathcal{A}}$ interchangeably. Also, because the uncertainty set $\mathcal{P}_{s,a}$ is compact, and the inner

product $\mathfrak{p}^\top \boldsymbol{v}$ is a continuous function with respect to $\mathfrak{p}$, the supremum over this set is attainable. Consequently, we can similarly use $\sup_{\mathfrak{p}\in\mathcal{P}_{s,a}}$ and $\max_{\mathfrak{p}\in\mathcal{P}_{s,a}}$ interchangeably.

Furthermore, for a fixed agent policy σ, its value vector $\boldsymbol{v}^\sigma$ is the unique fixed point of the policy evaluation operator. For any state $s \in \mathcal{S}$, it satisfies the following Bellman equation:

$$\boldsymbol{v}_s^\sigma = \mathbf{c}_s + \gamma \max_{\mathfrak{p}\in\mathcal{P}_{s,\sigma(s)}} \mathfrak{p}^\top \boldsymbol{v}^\sigma$$

Let P^σ denote the transition matrix induced by the agent's policy σ and the adversary's optimal response, where the adversary's transition probabilities at each state s are given by $\operatorname{argmax}_{\mathfrak{p}\in\mathcal{P}_{s,\sigma(s)}} \mathfrak{p}^\top \boldsymbol{v}^\sigma$. In matrix form, the value vector can be expressed as:

$$\boldsymbol{v}^\sigma = \mathbf{c} + \gamma P^\sigma \boldsymbol{v}^\sigma = (\mathbb{I} - \gamma P^\sigma)^{-1}\mathbf{c} \tag{2}$$

The following lemma establishes that $\mathcal{T}$ is a contraction mapping with a unique fixed point. This is a standard result in robust dynamic programming [Iye05].

Lemma 1. *The following statements hold:*

- *For all $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{R}^n$, $\|\mathcal{T}\boldsymbol{u} - \mathcal{T}\boldsymbol{v}\|_\infty \leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$.*
- *For every agent policy σ with value vector $\boldsymbol{v}^\sigma$, $\mathcal{T}\boldsymbol{v}^\sigma \preccurlyeq \boldsymbol{v}^\sigma$.*
- *There exists a unique $\boldsymbol{v}^\star \in \mathbb{R}^n$ such that $\mathcal{T}\boldsymbol{v}^\star = \boldsymbol{v}^\star$.*

Proof. • To prove the contraction property, fix a state $s \in \mathcal{S}$. Using the properties of the supremum and infimum, we can bound the difference as follows:

$$\begin{aligned}
(\mathcal{T}\boldsymbol{u})_s - (\mathcal{T}\boldsymbol{v})_s &= \gamma \left(\inf_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{u} - \inf_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v} \right) && \text{(Definition of } \mathcal{T}\text{)} \\
&\leqslant \gamma \sup_{a\in\mathcal{A}} \left(\sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{u} - \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v} \right) && (\inf f - \inf g \leqslant \sup(f-g)) \\
&\leqslant \gamma \sup_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \left(\mathfrak{p}^\top \boldsymbol{u} - \mathfrak{p}^\top \boldsymbol{v} \right) && (\sup f - \sup g \leqslant \sup(f-g)) \\
&= \gamma \sup_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top (\boldsymbol{u} - \boldsymbol{v}) && \text{(Linearity)} \\
&\leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty && \text{(Since } \mathfrak{p} \text{ is a distribution)}
\end{aligned}$$

Applying symmetric reasoning by swapping $\boldsymbol{u}$ and $\boldsymbol{v}$ yields $(\mathcal{T}\boldsymbol{v})_s - (\mathcal{T}\boldsymbol{u})_s \leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$. Combining these gives $|(\mathcal{T}\boldsymbol{u})_s - (\mathcal{T}\boldsymbol{v})_s| \leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$. Since this holds for all states $s \in \mathcal{S}$, it establishes $\|\mathcal{T}\boldsymbol{u} - \mathcal{T}\boldsymbol{v}\|_\infty \leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$.

- To prove the second statement, we evaluate the operator on the value vector $\boldsymbol{v}^\sigma$ for a given policy σ. For any state $s \in \mathcal{S}$, we have:

$$\begin{aligned}
(\mathcal{T}\boldsymbol{v}^\sigma)_s &= \mathbf{c}_s + \gamma \inf_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v}^\sigma && \text{(Definition of } \mathcal{T}\text{)} \\
&\leqslant \mathbf{c}_s + \gamma \sup_{\mathfrak{p}\in\mathcal{P}_{s,\sigma(s)}} \mathfrak{p}^\top \boldsymbol{v}^\sigma && \text{(Restricting infimum to } \sigma(s)\text{)} \\
&= \boldsymbol{v}_s^\sigma && \text{(Definition of } \boldsymbol{v}^\sigma\text{)}
\end{aligned}$$

This proves $\mathcal{T}\boldsymbol{v}^\sigma \preccurlyeq \boldsymbol{v}^\sigma$.

- Finally, because $\mathcal{T}$ is a contraction mapping on the complete metric space $\mathbb{R}^n$ equipped with the infinity norm, the Banach fixed-point theorem guarantees the existence of a unique vector $\boldsymbol{v}^\star \in \mathbb{R}^n$ such that $\mathcal{T}\boldsymbol{v}^\star = \boldsymbol{v}^\star$.

□

Now, we show that $\boldsymbol{v}^\star$ has a lower value in each state than any other value vector induced by an arbitrary agent policy.

Lemma 2. *For any arbitrary agent policy σ with value vector $\boldsymbol{v}^\sigma$, the optimal value vector $\boldsymbol{v}^\star$ satisfies $\boldsymbol{v}^\star \preccurlyeq \boldsymbol{v}^\sigma$.*

Proof. We first establish the monotonicity of the robust Bellman operator $\mathcal{T}$. Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}$ be two value vectors such that $\boldsymbol{u} \preccurlyeq \boldsymbol{v}$. For any state $s \in \mathcal{S}$:

$$\begin{aligned}(\mathcal{T}\boldsymbol{u})_s &= \mathbf{c}_s + \gamma \inf_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{u} \\ &\leqslant \mathbf{c}_s + \gamma \inf_{a\in\mathcal{A}} \sup_{\mathfrak{p}\in\mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v} \qquad \text{(Since } \boldsymbol{u} \preccurlyeq \boldsymbol{v} \text{ and } \mathfrak{p} \geqslant \mathbf{0}\text{)} \\ &= (\mathcal{T}\boldsymbol{v})_s\end{aligned}$$

Thus, $\mathcal{T}$ is a monotonic operator: $\boldsymbol{u} \preccurlyeq \boldsymbol{v} \implies \mathcal{T}\boldsymbol{u} \preccurlyeq \mathcal{T}\boldsymbol{v}$.

From Lemma 1, we know that for any agent policy σ, the operator evaluated at the policy's value vector yields $\mathcal{T}\boldsymbol{v}^\sigma \preccurlyeq \boldsymbol{v}^\sigma$.

By repeatedly applying the monotonic operator $\mathcal{T}$ to both sides of this inequality, we generate a chain of non-increasing vectors:

$$\begin{aligned}\mathcal{T}^2\boldsymbol{v}^\sigma &\preccurlyeq \mathcal{T}\boldsymbol{v}^\sigma \preccurlyeq \boldsymbol{v}^\sigma \\ \mathcal{T}^k\boldsymbol{v}^\sigma &\preccurlyeq \boldsymbol{v}^\sigma \quad \text{for any integer } k \geqslant 1\end{aligned}$$

Because $\mathcal{T}$ is a contraction mapping (Lemma 1), the sequence $\mathcal{T}^k\boldsymbol{v}^\sigma$ converges to the unique optimal fixed point $\boldsymbol{v}^\star$ as $k \to \infty$. Taking the limit of both sides preserves the non-strict inequality:

$$\begin{aligned}\lim_{k\to\infty} \mathcal{T}^k\boldsymbol{v}^\sigma &\preccurlyeq \boldsymbol{v}^\sigma \\ \boldsymbol{v}^\star &\preccurlyeq \boldsymbol{v}^\sigma\end{aligned}$$

This completes the proof. □

We next show that Algorithm 1 generates a non-increasing sequence of value functions that converges exponentially to the optimal value. This property guarantees monotonic policy improvement at each iteration.

Lemma 3. *Let σ^t and σ^{t+1} denote consecutive policies generated by `RMDP-PI` (Algorithm 1), and let $\boldsymbol{v}^t$ denote the value function of σ^t. For all $t \geqslant 0$, the following statements hold:*

- $\boldsymbol{v}^{t+1} \preccurlyeq \boldsymbol{v}^t$.
- $\left\|\boldsymbol{v}^t - \boldsymbol{v}^\star\right\|_\infty \leqslant \gamma^t \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty$.

Proof. • Let P^t be the transition matrix induced by σ^t and an optimal adversarial response to σ^t. We first establish that $\boldsymbol{v}^{t+1} \preccurlyeq \boldsymbol{v}^t$. We express the difference as follows:

$$\begin{aligned}\boldsymbol{v}^t - \boldsymbol{v}^{t+1} &= \boldsymbol{v}^t - (\mathbb{I} - \gamma P^{t+1})^{-1}\mathbf{c} && \text{(By Equation 2)} \\ &= (\mathbb{I} - \gamma P^{t+1})^{-1}(\mathbb{I} - \gamma P^{t+1})\left[\boldsymbol{v}^t - (\mathbb{I} - \gamma P^{t+1})^{-1}\mathbf{c}\right] && \text{(Multiplying by } \mathbb{I}\text{)} \\ &= (\mathbb{I} - \gamma P^{t+1})^{-1}\left[(\mathbb{I} - \gamma P^{t+1})\boldsymbol{v}^t - \mathbf{c}\right] && \text{(Expanding the product)} \\ &= (\mathbb{I} - \gamma P^{t+1})^{-1}\left[\boldsymbol{v}^t - (\mathbf{c} + \gamma P^{t+1}\boldsymbol{v}^t)\right] && \text{(Rearranging terms)} \\ &\geqslant (\mathbb{I} - \gamma P^{t+1})^{-1}\left[\boldsymbol{v}^t - \mathcal{T}\boldsymbol{v}^t\right] && \text{(By Definition of } \mathcal{T}\text{)}\end{aligned}$$

The final inequality follows from the definition of $\mathcal{T}$ and the non-negativity of the matrix $(\mathbb{I} - \gamma P^{t+1})^{-1}$, which we establish below. By Lemma 1, we know that $\boldsymbol{v}^t - \mathcal{T}\boldsymbol{v}^t \geqslant \mathbf{0}$. To evaluate $(\mathbb{I}-\gamma P^{t+1})^{-1}$, we can express it using the Neumann series expansion. Because P^{t+1} is a stochastic transition matrix, its spectral radius (maximum magnitude of eigenvalues) is 1. Multiplying it by the discount factor $\gamma \in (0, 1)$ bounds the spectral radius of γP^{t+1} to less than 1. This shows the inverse exists and can be expressed as the convergent infinite series:

$$(\mathbb{I} - \gamma P^{t+1})^{-1} = \sum_{k=0}^{\infty} (\gamma P^{t+1})^k$$

Furthermore, because $\gamma > 0$ and every entry in the probability matrix P^{t+1} is non-negative, every matrix in this infinite sum is also non-negative. Consequently, $(\mathbb{I} - \gamma P^{t+1})^{-1}$ is non-negative, and multiplying it by the non-negative vector $[\boldsymbol{v}^t - \mathcal{T}\boldsymbol{v}^t]$ ensures that the final product is non-negative. Therefore, $\boldsymbol{v}^t - \boldsymbol{v}^{t+1} \geqslant \mathbf{0}$, which proves $\boldsymbol{v}^{t+1} \preccurlyeq \boldsymbol{v}^t$.

- Next, we show that $\boldsymbol{v}^{t+1} \preccurlyeq \mathcal{T}\boldsymbol{v}^t$. We analyze the difference:

$$\begin{aligned}
\mathcal{T}\boldsymbol{v}^t - \boldsymbol{v}^{t+1} &\succcurlyeq \mathbf{c} + \gamma P^{t+1}\boldsymbol{v}^t - \boldsymbol{v}^{t+1} && \text{(Definition of } \mathcal{T} \text{ and } \sigma^{t+1}\text{)} \\
&= (\mathbb{I} - \gamma P^{t+1})\boldsymbol{v}^{t+1} + \gamma P^{t+1}\boldsymbol{v}^t - \boldsymbol{v}^{t+1} && \text{(Substituting for } \mathbf{c}\text{)} \\
&= \boldsymbol{v}^{t+1} - \gamma P^{t+1}\boldsymbol{v}^{t+1} + \gamma P^{t+1}\boldsymbol{v}^t - \boldsymbol{v}^{t+1} && \text{(Expanding the first term)} \\
&= \gamma P^{t+1}(\boldsymbol{v}^t - \boldsymbol{v}^{t+1}) && \text{(Simplifying)}
\end{aligned}$$

  The first inequality holds because σ^{t+1} is greedy for $\boldsymbol{v}^t$ and $P_s^{t+1} \in \mathcal{P}_{s,\sigma^{t+1}(s)}$. Since $\gamma > 0$, $P^{t+1} \geqslant 0$, and $\boldsymbol{v}^t \succcurlyeq \boldsymbol{v}^{t+1}$ (as proved in the previous step), we conclude that $\boldsymbol{v}^{t+1} \preccurlyeq \mathcal{T}\boldsymbol{v}^t$.

  Finally, we prove the second statement of the lemma using induction on t. The base case for $t = 0$ holds trivially because substituting $t = 0$ yields $\left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty \leqslant \gamma^0 \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty$, which simplifies to $\left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty \leqslant \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty$. For $t > 0$, we have:

$$\begin{aligned}
\left\|\boldsymbol{v}^t - \boldsymbol{v}^\star\right\|_\infty &\leqslant \left\|\mathcal{T}\boldsymbol{v}^{t-1} - \boldsymbol{v}^\star\right\|_\infty && (\boldsymbol{v}^\star \preccurlyeq \boldsymbol{v}^t \preccurlyeq \mathcal{T}\boldsymbol{v}^{t-1}) \\
&= \left\|\mathcal{T}\boldsymbol{v}^{t-1} - \mathcal{T}\boldsymbol{v}^\star\right\|_\infty && \text{(Since } \boldsymbol{v}^\star \text{ is the fixed point)} \\
&\leqslant \gamma \left\|\boldsymbol{v}^{t-1} - \boldsymbol{v}^\star\right\|_\infty && \text{(By contraction in Lemma 1)} \\
&\leqslant \gamma^t \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty && \text{(By induction)}
\end{aligned}$$

□

A.2 Potential Function

We next define the potential function. Intuitively, the potential of an action a in state s measures the additional cost incurred by deviating from the optimal policy $\sigma^\star$ and selecting a instead of $\sigma^\star(s)$.

Definition 1 (Potential function). *Given a state $s \in \mathcal{S}$ and an action $a \in \mathcal{A}$, the potential function $f(s, a)$ is defined as:*

$$f(s,a) := \sup_{\mathfrak{p} \in \mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v}^\star - \sup_{\mathfrak{p} \in \mathcal{P}_{s,\sigma^\star(s)}} \mathfrak{p}^\top \boldsymbol{v}^\star.$$

Next, we use the potential function to establish lower and upper bounds on the difference between an arbitrary policy's value vector and the optimal value.

Lemma 4. *Let σ be an arbitrary policy for the agent. The following statements hold:*

- ***(Lower bound)** If $\sigma(s) = a$ for some state $s \in \mathcal{S}$, then $\boldsymbol{v}_s^\sigma - \boldsymbol{v}_s^\star \geqslant \gamma f(s, a)$.*
- ***(Upper bound)** Let $\hat{s} = \operatorname{argmax}_{s \in \mathcal{S}} f(s, \sigma(s))$. Then $\|\boldsymbol{v}^\sigma - \boldsymbol{v}^\star\|_\infty \leqslant \frac{\gamma}{1-\gamma} f(\hat{s}, \sigma(\hat{s}))$.*

Proof. • **(Lower bound)** We lower bound the value difference as follows:

$$\begin{aligned}
\boldsymbol{v}_s^\sigma - \boldsymbol{v}_s^\star &= \left(\mathbf{c}_s + \gamma \sup_{\mathfrak{p} \in \mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v}^\sigma\right) - \left(\mathbf{c}_s + \gamma \sup_{\mathfrak{p} \in \mathcal{P}_{s,\sigma^\star(s)}} \mathfrak{p}^\top \boldsymbol{v}^\star\right) && \text{(By Equation 2)} \\
&= \gamma \left(\sup_{\mathfrak{p} \in \mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v}^\sigma - \sup_{\mathfrak{p} \in \mathcal{P}_{s,\sigma^\star(s)}} \mathfrak{p}^\top \boldsymbol{v}^\star\right) && \text{(Simplifying)} \\
&\geqslant \gamma \left(\sup_{\mathfrak{p} \in \mathcal{P}_{s,a}} \mathfrak{p}^\top \boldsymbol{v}^\star - \sup_{\mathfrak{p} \in \mathcal{P}_{s,\sigma^\star(s)}} \mathfrak{p}^\top \boldsymbol{v}^\star\right) && (\boldsymbol{v}^\sigma \succcurlyeq \boldsymbol{v}^\star \text{By Lemma 2}) \\
&= \gamma f(s,a) && \text{(By Definition)}
\end{aligned}$$

- **(Upper bound)** Assume τ is the adversary's best response to σ, and $\tau^\star$ is the adversary's best response to $\sigma^\star$. Let $P^{\sigma,\tau}$ and $P^\star$ denote the corresponding transition matrices under these fixed policies. We expand the value difference using Equation 2:

$$\begin{aligned}
\boldsymbol{v}^\sigma - \boldsymbol{v}^\star &= (\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}\mathbf{c} - (\mathbb{I} - \gamma P^\star)^{-1}\mathbf{c} \\
&= \left((\mathbb{I} - \gamma P^{\sigma,\tau})^{-1} - (\mathbb{I} - \gamma P^\star)^{-1}\right)\mathbf{c}.
\end{aligned}$$

Applying the matrix identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$ yields:

$$\begin{aligned}
\boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star} &= (\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}\Big((\mathbb{I} - \gamma P^{\star}) - (\mathbb{I} - \gamma P^{\sigma,\tau})\Big)(\mathbb{I} - \gamma P^{\star})^{-1}\mathbf{c} \\
&= \gamma\,(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}(P^{\sigma,\tau} - P^{\star})(\mathbb{I} - \gamma P^{\star})^{-1}\mathbf{c} && \text{(Simplifying terms)} \\
&= \gamma\,(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}(P^{\sigma,\tau} - P^{\star})\boldsymbol{v}^{\star} && \text{(Substituting } \boldsymbol{v}^{\star}\text{)}
\end{aligned}$$

Since $\boldsymbol{v}^{\sigma} \geqslant \boldsymbol{v}^{\star}$, we know $\boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star} \geqslant \mathbf{0}$. We bound the vector $(P^{\sigma,\tau} - P^{\star})\boldsymbol{v}^{\star}$ component-wise. For any state $s \in \mathcal{S}$:

$$\begin{aligned}
((P^{\sigma,\tau} - P^{\star})\boldsymbol{v}^{\star})_s &= P_s^{\sigma,\tau}\boldsymbol{v}^{\star} - P_s^{\star}\boldsymbol{v}^{\star} \\
&= \left(\operatorname*{argmax}_{\mathfrak{p}\in\mathcal{P}_{s,\sigma(s)}} \mathfrak{p}^{\top}\boldsymbol{v}^{\sigma}\right)^{\top}\boldsymbol{v}^{\star} - \sup_{\mathfrak{p}\in\mathcal{P}_{s,\sigma^{\star}(s)}} \mathfrak{p}^{\top}\boldsymbol{v}^{\star} && \text{(Definition of } P^{\sigma,\tau}, P^{\star}\text{)} \\
&\leqslant \sup_{\mathfrak{p}\in\mathcal{P}_{s,\sigma(s)}} \mathfrak{p}^{\top}\boldsymbol{v}^{\star} - \sup_{\mathfrak{p}\in\mathcal{P}_{s,\sigma^{\star}(s)}} \mathfrak{p}^{\top}\boldsymbol{v}^{\star} && \text{(Taking worst case in l.h.s)} \\
&= f(s,\sigma(s)) && \text{(Definition of } f\text{)}
\end{aligned}$$

Let $\mathbf{f}^{\sigma}$ be the vector where $\mathbf{f}_s^{\sigma} = f(s,\sigma(s))$. Thus, $(P^{\sigma,\tau} - P^{\star})\boldsymbol{v}^{\star} \leqslant \mathbf{f}^{\sigma}$. Because the matrix $(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}$ is non-negative, multiplying it preserves the inequality, forming a non-negative bounding chain:

$$\mathbf{0} \leqslant \boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star} \leqslant \gamma\,(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}\mathbf{f}^{\sigma}$$

Because all vectors are non-negative, taking the infinity norm preserves the inequality:

$$\begin{aligned}
\|\boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star}\|_{\infty} &\leqslant \left\|\gamma\,(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}\mathbf{f}^{\sigma}\right\|_{\infty} \\
&\leqslant \gamma\,\left\|(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}\right\|_{\infty}\|\mathbf{f}^{\sigma}\|_{\infty} && \text{(Submultiplicativity of } \ell_{\infty}\text{)}
\end{aligned}$$

We bound the first term using a Neumann series expansion by the same approach as Lemma 3. Because $P^{\sigma,\tau}$ is a stochastic matrix, $\left\|(P^{\sigma,\tau})^i\right\|_{\infty} = 1$. The series converges as follows:

$$\left\|(\mathbb{I} - \gamma P^{\sigma,\tau})^{-1}\right\|_{\infty} = \left\|\sum_{i=0}^{\infty}(\gamma P^{\sigma,\tau})^i\right\|_{\infty} \leqslant \sum_{i=0}^{\infty}\gamma^i\left\|(P^{\sigma,\tau})^i\right\|_{\infty} = \frac{1}{1-\gamma}.$$

Finally, we substitute this bound into the main inequality:

$$\begin{aligned}
\|\boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star}\|_{\infty} &\leqslant \frac{\gamma}{1-\gamma}\,\|\mathbf{f}^{\sigma}\|_{\infty} \\
&= \frac{\gamma}{1-\gamma}\cdot\max_{s\in\mathcal{S}} f(s,\sigma(s)) && \text{(Definition of } \mathbf{f}^{\sigma} \text{ and } \ell_{\infty}\text{)} \\
&= \frac{\gamma}{1-\gamma} f(\hat{s},\sigma(\hat{s})) && \text{(Definition of } \hat{s}\text{)}
\end{aligned}$$

This completes the proof.

□

We combine the lower and upper bounds to relate the value errors of two policies. Specifically, we show that if both policies choose the same action at the state with the maximum potential, the value error decreases by at most a factor of $1 - \gamma$.

Lemma 5. *Let σ and σ' be arbitrary policies such that $\boldsymbol{v}^{\sigma} \geqslant \boldsymbol{v}^{\sigma'}$. Let $\hat{s} = \operatorname{argmax}_{s \in \mathcal{S}} f(s, \sigma(s))$, and assume $\sigma'(\hat{s}) = \sigma(\hat{s})$. Then:*

$$\left\| \boldsymbol{v}^{\sigma'} - \boldsymbol{v}^{\star} \right\|_{\infty} \geqslant (1-\gamma) \left\| \boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star} \right\|_{\infty}.$$

Proof. We bound the value error as follows:

$$\begin{aligned}
\left\| \boldsymbol{v}^{\sigma'} - \boldsymbol{v}^{\star} \right\|_{\infty} &\geqslant \boldsymbol{v}^{\sigma'}_{\hat{s}} - \boldsymbol{v}^{\star}_{\hat{s}} && \text{(Definition of } \ell_{\infty} \text{ norm)} \\
&\geqslant \gamma f(\hat{s}, \sigma'(\hat{s})) && \text{(Lower bound from Lemma 4)} \\
&= \gamma f(\hat{s}, \sigma(\hat{s})) && \text{(Since } \sigma'(\hat{s}) = \sigma(\hat{s})\text{)} \\
&\geqslant (1-\gamma) \left\| \boldsymbol{v}^{\sigma} - \boldsymbol{v}^{\star} \right\|_{\infty} && \text{(Upper bound from Lemma 4)}
\end{aligned}$$

This completes the proof. □

A.3 Elimination of Actions with Maximum Potential

We use the convergence rate found in Lemma 3 to show that the action selected in the state with maximum potential cannot repeat in the algorithm for more than $L := \lfloor \log_{\gamma}(1-\gamma) \rfloor + 1$ consecutive iterations.

Lemma 6. *Let $\sigma^0, \ldots, \sigma^T$ denote the sequence of agent policies generated by `RMDP-PI`. For any non-optimal policy σ^l in this sequence, let $\hat{s} = \operatorname{argmax}_{s \in \mathcal{S}} f(s, \sigma^l(s))$. Then, $\sigma^k(\hat{s}) \neq \sigma^l(\hat{s})$ for all $k \geqslant l + L$.*

Proof. Assume for contradiction that there exists $k \geqslant l + L$ such that $\sigma^k(\hat{s}) = \sigma^l(\hat{s})$. Because the value vectors are non-increasing, $\boldsymbol{v}^{\sigma^l} \geqslant \boldsymbol{v}^{\sigma^k}$. Applying Lemma 5 yields:

$$\left\| \boldsymbol{v}^{\sigma^k} - \boldsymbol{v}^{\star} \right\|_{\infty} \geqslant (1-\gamma) \left\| \boldsymbol{v}^{\sigma^l} - \boldsymbol{v}^{\star} \right\|_{\infty}.$$

However, the global convergence rate from Lemma 3 dictates:

$$\left\| \boldsymbol{v}^{\sigma^k} - \boldsymbol{v}^{\star} \right\|_{\infty} \leqslant \gamma^{k-l} \left\| \boldsymbol{v}^{\sigma^l} - \boldsymbol{v}^{\star} \right\|_{\infty}.$$

Since σ^l is not optimal, $\left\| \boldsymbol{v}^{\sigma^l} - \boldsymbol{v}^{\star} \right\|_{\infty} > 0$. Combining the bounds and dividing by this positive term yields:

$$1 - \gamma \leqslant \gamma^{k-l}.$$

However, $k - l \geqslant L$ implies $\gamma^{k-l} \leqslant \gamma^L < 1 - \gamma$, which is a contradiction. □

Finally, we combine the elimination steps to establish a polynomial upper bound on the total number of iterations.

Lemma 7. *`RMDP-PI` terminates with an optimal policy in $\mathcal{O}\left(n \cdot m \cdot \frac{\log(1-\gamma)}{\log \gamma}\right)$ iterations.*

Proof. There are exactly $n \cdot m$ distinct state-action pairs. By Lemma 6, if σ^l is non-optimal, $\hat{s} \in \operatorname{argmax}_{s \in \mathcal{S}} f(s, \sigma^l(s))$, and $\hat{a} = \sigma^l(\hat{s})$, the policy will not select $\hat{a}$ at $\hat{s}$ for any iteration $k \geqslant l + L$. Consequently, the algorithm permanently eliminates at least one state-action pair every L iterations until an optimal policy is reached. With fixed tie-breaking, at most two further iterations are needed for the stopping test to succeed. Therefore,

$$T \leqslant n \cdot m \cdot L + 2 = \mathcal{O}(n \cdot m \cdot \log_{\gamma}(1-\gamma)) = \mathcal{O}(n \cdot m \cdot \frac{\log(1-\gamma)}{\log(\gamma)}).$$

□

A.4 Reduction to One RMC Oracle Call

We show that the maximization part of the policy improvement step can be computed for all states and actions simultaneously with one call to the RMC oracle with the same size as the input size ($\mathcal{O}(nm)$).

Lemma 8. *The policy improvement step in Line 10 of Algorithm 1 for all states $s \in \mathcal{S}$ can be computed with one call to an RMC oracle on an RMC with $\mathcal{O}(nm)$ states.*

Proof. In Line 10 of `RMDP-PI`, the algorithm updates the policy by computing:

$$\sigma^t(s) \leftarrow \operatorname*{argmin}_{a \in \mathcal{A}} \max_{\mathfrak{p} \in \mathcal{P}(s,a)} \{c(s) + \gamma \mathfrak{p}^\top \boldsymbol{v}^{t-1}\}$$

To evaluate this inner maximization for all state-action pairs using a single RMC oracle call, we construct a batch RMC $\mathcal{M}_{\text{batch}} = (\mathcal{S}_{\text{batch}}, c_{\text{batch}}, \mathit{Succ}_{\text{batch}}, \mathcal{P}_{\text{batch}})$. We introduce a set of transient states to represent each state-action pair, and a set of absorbing states to represent the original states. Let $Z = \{z_{s,a} \mid s \in \mathcal{S}, a \in \mathcal{A}\}$ be the transient states, and $X = \{x_s \mid s \in \mathcal{S}\}$ be the absorbing states. We formulate $\mathcal{M}_{\text{batch}}$ as follows:

- $\mathcal{S}_{\text{batch}} = Z \cup X$. The total number of states is $n \cdot m + n = \mathcal{O}(nm)$.
- $c_{\text{batch}}(z_{s,a}) = c(s)$ for all $z_{s,a} \in Z$, and for all $x_s \in X$, $c_{\text{batch}}(x_s) = (1-\gamma)\boldsymbol{v}^{t-1}(s)$.
- $\mathit{Succ}_{\text{batch}}(z_{s,a}) = \{x_{s'} \mid s' \in \mathit{Succ}(s,a)\}$ for all $z_{s,a} \in Z$, and for all $x_s \in X$, $\mathit{Succ}_{\text{batch}}(x_s) = \{x_s\}$ (making the original states absorbing).
- $\mathcal{P}_{\text{batch}}(z_{s,a}) = \mathcal{P}(s,a)$ for all $z_{s,a} \in Z$, and for all $x_s \in X$, $\mathcal{P}_{\text{batch}}(x_s)$ is restricted to the deterministic transition to x_s.

Because every state $x_s \in X$ deterministically transitions to itself with a cost of $(1-\gamma)\boldsymbol{v}^{t-1}(s)$, its value in $\mathcal{M}_{\text{batch}}$ is $\boldsymbol{v}^{t-1}(s)$.

Consequently, the value of any transient state $z_{s,a} \in Z$ satisfies the robust Bellman equation:

$$\boldsymbol{v}_{\text{batch}}(z_{s,a}) = c_{\text{batch}}(z_{s,a}) + \gamma \max_{\mathfrak{p} \in \mathcal{P}_{\text{batch}}(z_{s,a})} \mathfrak{p}^\top \boldsymbol{v}_{\text{batch}} = c(s) + \gamma \max_{\mathfrak{p} \in \mathcal{P}(s,a)} \mathfrak{p}^\top \boldsymbol{v}^{t-1}$$

Therefore, we construct $\mathcal{M}_{\text{batch}}$ and query the RMC oracle once to find the value of all states simultaneously. Then, for each state $s \in \mathcal{S}$, we extract the optimal action by evaluating $\sigma^t(s) \leftarrow \operatorname{argmin}_{a \in \mathcal{A}} \boldsymbol{v}_{\text{batch}}(z_{s,a})$. This shows that the policy improvement step for all states requires only a single RMC oracle call on an RMC of size $\mathcal{O}(nm)$. □

We now combine the iteration bound with the oracle-call bound to prove Theorem 1.

Proof of Theorem 1. By Lemma 7, the `RMDP-PI` algorithm converges to the optimal policy in $\mathcal{O}\left(n \cdot m \cdot \frac{\log(1-\gamma)}{\log \gamma}\right)$ iterations.

We count the number of RMC oracle calls required in each iteration. During the policy evaluation step (Line 6 of Algorithm 1), the algorithm solves the RMC for the current policy, which requires one call to the RMC oracle. Then, during the policy improvement step, Lemma 8 shows that the algorithm can process all states simultaneously with one additional RMC oracle call, whose size is linear according to the input size ($\mathcal{O}(nm)$). This results in two RMC oracle queries per iteration.

Because the total number of iterations is bounded by $\mathcal{O}\left(n \cdot m \cdot \frac{\log(1-\gamma)}{\log \gamma}\right)$, and each iteration requires two RMC oracle solutions, the algorithm finds the optimal policy with a polynomial number of calls to the RMC oracle. □

B Proof of Theorem 2: Complexity of RMCs with L_1, L_∞ Uncertainty Sets

The previous section established a polynomial bound on the number of policy iteration steps and RMC oracle calls for RMDPs with compact uncertainty sets and a constant discount factor. This section focuses on solving Robust Markov Chains (RMCs) with L_1, L_∞ uncertainty sets.

We demonstrate that RMCs under L_1 and L_∞ uncertainty sets can be solved in strongly polynomial time using the policy iteration procedure, `RMC-PI`, detailed in Algorithm 2. While the strongly

polynomial solvability of the L_∞ case was previously established by Asadi et al. [ACG+26a], our analysis yields a tighter bound by a factor of $|\mathcal{S}|$.

Before diving into the main proof, we briefly recall some standard results for Markov models that apply regardless of the specific uncertainty set [Put94, Iye05].

Robust Bellman Optimality Operator. Given an RMC $\mathcal{M}$ and a value vector $\boldsymbol{v}$, the robust Bellman operator $\mathcal{T}$ (for Algorithm 2) is defined as:

$$(\mathcal{T}\boldsymbol{v})_s = \mathbf{c}_s + \gamma \sup_{\mathfrak{p}\in\mathcal{P}_s} \mathfrak{p}^\top \boldsymbol{v}. \tag{3}$$

Note that because $\mathcal{P}_s$ is compact and the inner product $\mathfrak{p}^\top \boldsymbol{v}$ is a continuous function with respect to $\mathfrak{p}$, the supremum over this set is attainable. Consequently, we can use $\sup_{\mathfrak{p}\in\mathcal{P}_s}$ and $\max_{\mathfrak{p}\in\mathcal{P}_s}$ interchangeably.

It is a well-known result that policy iteration converges monotonically to the unique fixed point of $\mathcal{T}$ at an exponential rate. We summarize the core contractive and monotonic properties of this operator in the following lemma.

Lemma 9. *The following statements hold:*

- *For any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, $\|\mathcal{T}\boldsymbol{u} - \mathcal{T}\boldsymbol{v}\|_\infty \leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$.*
- *For any adversary policy τ with value vector $\boldsymbol{v}^\tau$, $\mathcal{T}\boldsymbol{v}^\tau \geqslant \boldsymbol{v}^\tau$.*
- *There is a unique optimal value vector $\boldsymbol{v}^\star \in \mathbb{R}^n$ such that $\mathcal{T}\boldsymbol{v}^\star = \boldsymbol{v}^\star$.*

Proof. • Fix an arbitrary state $s \in \mathcal{S}$. Without loss of generality, assume $(\mathcal{T}\boldsymbol{u})_s \geqslant (\mathcal{T}\boldsymbol{v})_s$. Let $\mathfrak{p}'$ and $\mathfrak{p}''$ be the respective maximizers in $\mathcal{P}_s$ for $\boldsymbol{u}$ and $\boldsymbol{v}$. We can bound the difference as follows:

$$\begin{aligned}
(\mathcal{T}\boldsymbol{u})_s - (\mathcal{T}\boldsymbol{v})_s &= \gamma({\mathfrak{p}'}^\top \boldsymbol{u} - {\mathfrak{p}''}^\top \boldsymbol{v}) && \text{(Definition of } \mathcal{T}) \\
&\leqslant \gamma({\mathfrak{p}'}^\top \boldsymbol{u} - {\mathfrak{p}'}^\top \boldsymbol{v}) && \text{(Optimality of } \mathfrak{p}'' \text{ for } \boldsymbol{v}) \\
&= \gamma {\mathfrak{p}'}^\top (\boldsymbol{u} - \boldsymbol{v}) && \text{(Factorization)} \\
&\leqslant \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_\infty . && (\mathfrak{p}' \text{ is a probability distribution})
\end{aligned}$$

Since this holds for any state s, the contraction property follows.

- For the second claim, observe that for any state $s \in \mathcal{S}$:

$$\begin{aligned}
(\mathcal{T}\boldsymbol{v}^\tau)_s &= \mathbf{c}_s + \gamma \max_{\mathfrak{p}\in\mathcal{P}_s} \mathfrak{p}^\top \boldsymbol{v}^\tau && \text{(Definition of } \mathcal{T}) \\
&\geqslant \mathbf{c}_s + \gamma(\mathfrak{p}_s^\tau)^\top \boldsymbol{v}^\tau && (\mathfrak{p}_s^\tau \text{ is feasible in } \mathcal{P}_s) \\
&= \boldsymbol{v}_s^\tau. && \text{(Definition of } \boldsymbol{v}^\tau)
\end{aligned}$$

Because this applies to all states simultaneously, we conclude $\mathcal{T}\boldsymbol{v}^\tau \geqslant \boldsymbol{v}^\tau$.

- Finally, existence and uniqueness follow immediately from applying the Banach fixed-point theorem to our contraction mapping.

□

Next, we show that the optimal value vector $\boldsymbol{v}^\star$ is an upper bound in each entry for the value vector induced by any arbitrary adversary policy.

Lemma 10. *For any arbitrary adversary policy τ with value vector $\boldsymbol{v}^\tau$, the optimal value vector $\boldsymbol{v}^\star$ satisfies $\boldsymbol{v}^\star \geqslant \boldsymbol{v}^\tau$.*

Proof. We first establish the monotonicity of the robust Bellman operator $\mathcal{T}$ for RMCs. Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}$ be two value vectors such that $\boldsymbol{u} \leqslant \boldsymbol{v}$. For any state $s \in \mathcal{S}$:

$$\begin{aligned}
(\mathcal{T}\boldsymbol{u})_s &= \mathbf{c}_s + \gamma \max_{\mathfrak{p}\in\mathcal{P}(s)} \mathfrak{p}^\top \boldsymbol{u} \\
&\leqslant \mathbf{c}_s + \gamma \max_{\mathfrak{p}\in\mathcal{P}(s)} \mathfrak{p}^\top \boldsymbol{v} && (\text{Since } \boldsymbol{u} \leqslant \boldsymbol{v} \text{ and } \mathfrak{p} \geqslant \mathbf{0}) \\
&= (\mathcal{T}\boldsymbol{v})_s
\end{aligned}$$

Thus, $\mathcal{T}$ is a monotonic operator, meaning $\boldsymbol{u} \preccurlyeq \boldsymbol{v} \implies \mathcal{T}\boldsymbol{u} \preccurlyeq \mathcal{T}\boldsymbol{v}$.

By Lemma 9, we know that evaluating the operator at the value vector of any adversary policy τ yields $\mathcal{T}\boldsymbol{v}^\tau \succcurlyeq \boldsymbol{v}^\tau$. By repeatedly applying the monotonic operator $\mathcal{T}$ to both sides of this inequality, we generate a chain of non-decreasing vectors:

$$\begin{aligned} \boldsymbol{v}^\tau &\preccurlyeq \mathcal{T}\boldsymbol{v}^\tau \preccurlyeq \mathcal{T}^2\boldsymbol{v}^\tau \\ \boldsymbol{v}^\tau &\preccurlyeq \mathcal{T}^k\boldsymbol{v}^\tau \quad \text{for any integer } k \geqslant 1 \end{aligned}$$

Because $\mathcal{T}$ is a contraction mapping, the sequence $\mathcal{T}^k\boldsymbol{v}^\tau$ converges to the unique optimal fixed point $\boldsymbol{v}^\star$ as $k \to \infty$. Taking the limit of both sides preserves the non-strict inequality:

$$\begin{aligned} \boldsymbol{v}^\tau &\preccurlyeq \lim_{k\to\infty} \mathcal{T}^k\boldsymbol{v}^\tau \\ \boldsymbol{v}^\tau &\preccurlyeq \boldsymbol{v}^\star \end{aligned}$$

This establishes that $\boldsymbol{v}^\star \succcurlyeq \boldsymbol{v}^\tau$, completing the proof. □

By the properties established in Lemmas 9, 10, we now show that the sequence of value vectors produced by the `RMC-PI` algorithm is non-decreasing and converges at an exponential rate.

Lemma 11. *Suppose $\tau^0, \tau^1, \ldots$ represents the sequence of policies obtained via `RMC-PI` (Algorithm 2), with $\boldsymbol{v}^t$ denoting the corresponding value vector for policy τ^t. For any iteration t, the following properties hold:*

- $\boldsymbol{v}^{t+1} \succcurlyeq \boldsymbol{v}^t$
- $\left\|\boldsymbol{v}^t - \boldsymbol{v}^\star\right\|_\infty \leqslant \gamma^t \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty$

Proof. • Let P^t denote the transition matrix corresponding to the policy chosen at iteration t. We analyze the difference between consecutive value vectors:

$$\begin{aligned} \boldsymbol{v}^{t+1} - \boldsymbol{v}^t &= (\mathbb{I} - \gamma P^{t+1})^{-1}\mathbf{c} - \boldsymbol{v}^t && \text{(Definition of policy value)} \\ &= (\mathbb{I} - \gamma P^{t+1})^{-1}\left[\mathbf{c} - (\mathbb{I} - \gamma P^{t+1})\boldsymbol{v}^t\right] && \text{(Factoring out the inverse matrix)} \\ &= (\mathbb{I} - \gamma P^{t+1})^{-1}\left[\mathbf{c} + \gamma P^{t+1}\boldsymbol{v}^t - \boldsymbol{v}^t\right] && \text{(Expanding the inner terms)} \\ &\succcurlyeq (\mathbb{I} - \gamma P^{t+1})^{-1}\left[\mathcal{T}\boldsymbol{v}^t - \boldsymbol{v}^t\right]. && \text{(Definition of } \mathcal{T}\text{)} \end{aligned}$$

The final inequality follows from the definition of $\mathcal{T}$ and the non-negativity of the matrix $(\mathbb{I} - \gamma P^{t+1})^{-1}$, which we establish below. By Lemma 9, we know $\mathcal{T}\boldsymbol{v}^t - \boldsymbol{v}^t \succcurlyeq \mathbf{0}$. We can express the inverse matrix using the Neumann series:

$$(\mathbb{I} - \gamma P^{t+1})^{-1} = \sum_{i=0}^{\infty} (\gamma P^{t+1})^i.$$

Because γ and all entries of P^{t+1} are non-negative, the resulting sum consists entirely of non-negative terms and converges. Consequently, the matrix $(\mathbb{I} - \gamma P^{t+1})^{-1}$ is component-wise non-negative, which implies $\boldsymbol{v}^{t+1} - \boldsymbol{v}^t \succcurlyeq \mathbf{0}$.

- We first establish the intermediate bound $\boldsymbol{v}^{t+1} \succcurlyeq \mathcal{T}\boldsymbol{v}^t$. We can show this by expanding the difference:

$$\begin{aligned} \boldsymbol{v}^{t+1} - \mathcal{T}\boldsymbol{v}^t &= \boldsymbol{v}^{t+1} - (\mathbf{c} + \gamma P^{t+1}\boldsymbol{v}^t) && \text{(Definition of } \mathcal{T} \text{ and policy update)} \\ &= \boldsymbol{v}^{t+1} - \left((\mathbb{I} - \gamma P^{t+1})\boldsymbol{v}^{t+1} + \gamma P^{t+1}\boldsymbol{v}^t\right) && \text{(Substituting } \mathbf{c} = (\mathbb{I} - \gamma P^{t+1})\boldsymbol{v}^{t+1}\text{)} \\ &= \boldsymbol{v}^{t+1} - (\boldsymbol{v}^{t+1} - \gamma P^{t+1}\boldsymbol{v}^{t+1} + \gamma P^{t+1}\boldsymbol{v}^t) && \text{(Expanding the expression)} \\ &= \gamma P^{t+1}(\boldsymbol{v}^{t+1} - \boldsymbol{v}^t). && \text{(Simplifying)} \end{aligned}$$

Given $\gamma \geqslant 0$, $P^{t+1} \geqslant 0$ entrywise, and our previous result that $\boldsymbol{v}^{t+1} - \boldsymbol{v}^t \succcurlyeq \mathbf{0}$, we conclude $\boldsymbol{v}^{t+1} \succcurlyeq \mathcal{T}\boldsymbol{v}^t$.

Now we prove the second statement of the lemma using induction on t. The base case $t = 0$ holds trivially. For $t > 0$:

$$\begin{aligned}
\left\|\boldsymbol{v}^t - \boldsymbol{v}^\star\right\|_\infty &\leqslant \left\|\mathcal{T}\boldsymbol{v}^{t-1} - \boldsymbol{v}^\star\right\|_\infty && \text{(Because } \boldsymbol{v}^\star \geqslant \boldsymbol{v}^t \geqslant \mathcal{T}\boldsymbol{v}^{t-1}\text{)} \\
&= \left\|\mathcal{T}\boldsymbol{v}^{t-1} - \mathcal{T}\boldsymbol{v}^\star\right\|_\infty && \text{(Fixed point property, Lemma 9)} \\
&\leqslant \gamma \left\|\boldsymbol{v}^{t-1} - \boldsymbol{v}^\star\right\|_\infty && \text{(Contraction of } \mathcal{T}\text{, Lemma 9)} \\
&\leqslant \gamma^t \left\|\boldsymbol{v}^0 - \boldsymbol{v}^\star\right\|_\infty . && \text{(By induction)}
\end{aligned}$$

□

B.1 Main Proof of Theorem 2

In this section, we will show that policy iteration is a strongly polynomial-time algorithm for RMCs with L_1 or L_∞ uncertainty set and a constant discount factor. This, combined with Theorem 1 shows that RMDPs with L_1 or L_∞ uncertainty sets and a constant discount factor are strongly polynomial.

We prove this by showing that policy iteration converges to the optimal policy in a polynomial number of iterations. To organize our analysis, we first introduce fast linear-time algorithms for the policy improvement step and identify the structural properties of the distributions they produce. Then, we use these properties to bound the convergence rate of our method.

Before we analyze the algorithms, we define the value of a policy in an RMC. For any adversary policy τ, let P^τ be its transition matrix and $\mathfrak{p}_s^\tau$ be its transition probabilities at state s. The value vector $\boldsymbol{v}^\tau$ satisfies the following Bellman equation for each state $s \in \mathcal{S}$:

$$\boldsymbol{v}_s^\tau = \mathbf{c}_s + \gamma {\mathfrak{p}_s^\tau}^\top \boldsymbol{v}^\tau \tag{4}$$

We can also write this in matrix form to calculate the value vector directly:

$$\boldsymbol{v}^\tau = (\mathbb{I} - \gamma P^\tau)^{-1}\mathbf{c} \tag{5}$$

B.1.1 Policy Improvement and Structural Property.

Line 9 of Algorithm 2 requires solving the following optimization problem:

$$\max_{\mathfrak{p} \in \mathcal{P}(s)} \mathfrak{p}^\top \boldsymbol{v}$$

where $\mathcal{P}(s) = \{\mathfrak{p} \in \Delta(\mathcal{S}) \mid \|\mathfrak{p} - \hat{\mathfrak{p}}_s\|_p \leqslant \delta_s\}$ for $p \in \{1, \infty\}$ for a given value vector $\boldsymbol{v}$. Note that for $p \in \{1, \infty\}$, the uncertainty set $\mathcal{P}(s)$ can be defined by linear constraints.

While one can solve this optimization problem using linear programming, a more efficient linear-time algorithm exists for L_1 and L_∞.

For L_1, [HPW21] presents a linear-time method, which they refer to as the homotopy algorithm. Given an RMC $\mathcal{M}$, a state $s \in \mathcal{S}$, and the current value vector $\boldsymbol{v}$, the algorithm first sorts the successor states in descending order of their values, such that $\boldsymbol{v}_{s_1} \geqslant \boldsymbol{v}_{s_2} \geqslant \cdots \geqslant \boldsymbol{v}_{s_{|Succ(s)|}}$. It then iteratively shifts transition probability mass from the states with the lowest values to the state with the highest value (s_1). Specifically, the algorithm maintains a pointer hi (initially pointing to the state with the lowest value $s_{|Succ(s)|}$) and transfers the maximum permissible probability mass from $\mathfrak{p}_{s_{hi}}$ to $\mathfrak{p}_{s_1}$ at each step. It decrements hi once the probability mass at s_{hi} is exhausted. Algorithm 3 details this procedure. We refer the reader to [HPW21] for the formal proof of correctness.

The policies generated by Algorithm 3 exhibit the following structural property that we use in our analysis.

Property 1 (Structural Characterization for L_1)**.** *For any policy $\mathfrak{p}$ generated by Algorithm 3, there exist four disjoint sets of states $R_\mathfrak{p}, N_\mathfrak{p}, Z_\mathfrak{p}$, and $I_\mathfrak{p}$ such that:*

- $|R_\mathfrak{p}| = 1$ *(Receiver), and for $s' \in R_\mathfrak{p}$ we have* $\mathfrak{p}_{s'} = \min(\hat{\mathfrak{p}}_{s,s'} + \frac{\delta_s}{2}, 1)$.
- *For $s' \in N_\mathfrak{p}$ (Not-changed),* $\mathfrak{p}_{s'} = \hat{\mathfrak{p}}_{s,s'}$.
- *For $s' \in Z_\mathfrak{p}$ (Zeros),* $\mathfrak{p}_{s'} = 0$.
- $|I_\mathfrak{p}| \leqslant 1$ *(Incomplete), and for $s' \in I_\mathfrak{p}$, the probability satisfies* $\mathfrak{p}_{s'} = \hat{\mathfrak{p}}_{s,s'} - \frac{\delta_s}{2} + \sum_{s'' \in Z_\mathfrak{p}} \hat{\mathfrak{p}}_{s,s''}$.

Algorithm 3 Best-case distribution under L_1 uncertainty sets

Input: State s, nominal distribution $\hat{\mathfrak{p}}_s$, state values $\boldsymbol{v}$, budget $\delta(s)$
Output: Distribution $\mathfrak{p}$ maximizing $\mathfrak{p}^\top \boldsymbol{v}$
$\mathfrak{p} \leftarrow \hat{\mathfrak{p}}_s$
Sort states $s_1, \ldots, s_{|Succ(s)|}$ so that $\boldsymbol{v}(s_1) \geqslant \boldsymbol{v}(s_2) \geqslant \cdots \geqslant \boldsymbol{v}(s_{|Succ(s)|})$ ▷ Break ties by increasing state index
$hi \leftarrow |Succ(s)|$ ▷ Index of lowest-value state
while $\delta(s) > 0$ and $hi > 1$ **do**
    $d \leftarrow \min\left(\mathfrak{p}_{s_{hi}}, \frac{\delta(s)}{2}\right)$
    $\mathfrak{p}_{s_{hi}} \leftarrow \mathfrak{p}_{s_{hi}} - d$
    $\mathfrak{p}_{s_1} \leftarrow \mathfrak{p}_{s_1} + d$
    $\delta(s) \leftarrow \delta(s) - 2d$
    $hi \leftarrow hi - 1$
return $\mathfrak{p}$

Property 1 follows directly from the execution of Algorithm 3. The set $R_\mathfrak{p}$ contains the highest-value state receiving the shifted probability. Conversely, $Z_\mathfrak{p}$ consists of states whose probability became zero, while $I_\mathfrak{p}$ contains at most one state that partially gives its mass before the budget finishes. Finally, $N_\mathfrak{p}$ consists of the not-changed states that the pointer hi never reaches.

Similarly, for L_∞, Behzadian et al. [BPH21] present a linear algorithm. It starts with the nominal distribution $\hat{\mathfrak{p}}_s$. Then, it sorts the successor states from highest value to lowest value. The algorithm uses a two-pointer approach. One pointer (hi) starts on the highest-value state, and the other pointer (lo) starts on the lowest-value state. It moves probability mass from the lowest-value states to the highest-value states. Each state can only give or receive up to δ_s mass, and the probability must stay within the $[0, 1]$ limits. The pointers move until they meet. Algorithm 4 details this procedure.

Algorithm 4 Best-case distribution under L_∞ uncertainty sets

Input: State s, nominal distribution $\hat{\mathfrak{p}}_s$, state values $\boldsymbol{v}$, budget δ_s
Output: Distribution $\mathfrak{p}$ maximizing $\mathfrak{p}^\top \boldsymbol{v}$
Sort states $s_1, \ldots, s_{|Succ(s)|}$ so that $\boldsymbol{v}(s_1) \geqslant \boldsymbol{v}(s_2) \geqslant \cdots \geqslant \boldsymbol{v}(s_{|Succ(s)|})$ ▷ Break ties by increasing state index
$\mathfrak{p} \leftarrow \hat{\mathfrak{p}}_s$
$hi \leftarrow 1$, $lo \leftarrow |Succ(s)|$
$b_{hi} \leftarrow \delta_s$, $b_{lo} \leftarrow \delta_s$
while $hi < lo$ **do**
    $d_{hi} \leftarrow \min(b_{hi}, 1 - \mathfrak{p}_{s_{hi}})$
    $d_{lo} \leftarrow \min(b_{lo}, \mathfrak{p}_{s_{lo}})$
    $t \leftarrow \min(d_{hi}, d_{lo})$
    $\mathfrak{p}_{s_{hi}} \leftarrow \mathfrak{p}_{s_{hi}} + t$
    $\mathfrak{p}_{s_{lo}} \leftarrow \mathfrak{p}_{s_{lo}} - t$
    $b_{hi} \leftarrow b_{hi} - t$
    $b_{lo} \leftarrow b_{lo} - t$
    if $b_{hi} = 0$ or $\mathfrak{p}_{s_{hi}} = 1$ **then**
        $hi \leftarrow hi + 1$
        $b_{hi} \leftarrow \delta_s$
    else
        $lo \leftarrow lo - 1$
        $b_{lo} \leftarrow \delta_s$
return $\mathfrak{p}$

Similar to the L_1 case, the policies generated by Algorithm 4 exhibit a clear structural property.
Property 2 (Structural Characterization for L_∞)**.** *For any policy* $\mathfrak{p}$ *generated by Algorithm 4, there exist four disjoint sets of states* $R_\mathfrak{p}, D_\mathfrak{p}, Z_\mathfrak{p}$, *and* $I_\mathfrak{p}$ *such that:*

- *For* $s' \in R_\mathfrak{p}$ *(Receivers),* $\mathfrak{p}_{s'} = \min(1, \hat{\mathfrak{p}}_{s,s'} + \delta_s)$.

- *For $s' \in D_{\mathfrak{p}}$ (Donors), $\mathfrak{p}_{s'} = \hat{\mathfrak{p}}_{s,s'} - \delta_s$.*
- *For $s' \in Z_{\mathfrak{p}}$ (Zeroed donors), $\hat{\mathfrak{p}}_{s,s'} \leqslant \delta_s$ and $\mathfrak{p}_{s'} = 0$.*
- *$|I_{\mathfrak{p}}| \leqslant 1$ (Incomplete), and for $s' \in I_{\mathfrak{p}}$, the probability satisfies $\hat{\mathfrak{p}}_{s,s'} - \delta_s < \mathfrak{p}_{s'} < \hat{\mathfrak{p}}_{s,s'} + \delta_s$.*

Property 2 shows how the probability mass moves. The set $R_{\mathfrak{p}}$ receives the mass. This mass is donated by the states in $D_{\mathfrak{p}}$ and $Z_{\mathfrak{p}}$. Finally, there is at most one state, $I_{\mathfrak{p}}$, that only partially gives or receives mass.

B.1.2 Potential Function and Sandwich Bounds

We now analyze the convergence rate of our policy iteration method. This analysis differs from the standard RMDP analysis because the action space of the adversary is infinite. Although it is easy to show that the optimal worst-case probabilities must lie on the boundary of the L_1 or L_∞ ball, there can still be exponentially many such extreme points.

First, we define the cumulative transition probability for an adversary policy τ as follows.

Definition 2. *Fix a state $s \in \mathcal{S}$, and let $s_1, \ldots, s_{|Succ(s)|}$ be an enumeration of the successor states such that $\boldsymbol{v}^\star_{s_1} \geqslant \boldsymbol{v}^\star_{s_2} \geqslant \cdots \geqslant \boldsymbol{v}^\star_{s_{|Succ(s)|}}$. We break ties by increasing state index. Let $\mathfrak{p}^\tau_s$ denote the transition probability vector at state s under the adversary policy τ. For any index $i \in [|Succ(s)|]$, we define the cumulative transition probability $F^\tau(s,i)$ as:*

$$F^\tau(s,i) := \sum_{j=1}^{i} \mathfrak{p}^\tau_{s,s_j}.$$

We choose $\tau^\star$ by applying Algorithm 3 for L_1, or Algorithm 4 for L_∞, at each state with value vector $\boldsymbol{v}^\star$ and this ordering. We write $F^\star(s,i) = F^{\tau^\star}(s,i)$ and $\mathfrak{p}^\star_s = \mathfrak{p}^{\tau^\star}_s$.

We now have the following lemma.

Lemma 12. *For any state $s \in \mathcal{S}$, any policy τ with $\mathfrak{p}^\tau_s \in \mathcal{P}(s)$, and any index $i \in [|Succ(s)|]$, we have $F^\star(s,i) \geqslant F^\tau(s,i)$.*

Proof. Algorithms 3, 4 operate on a given value permutation by greedily transferring probability mass from the lowest-value states at the end of the sequence to the highest-value states at the start until the budget saturates. Because the optimal policy evaluates states using the descending permutation of the $\boldsymbol{v}^\star$ values, $\mathfrak{p}^\star_s$ shifts the maximum allowable mass as early as possible in this specific order. For each prefix, the procedure maximizes the mass transferred into it from the remaining states, subject to the uncertainty constraints. Transfers within the prefix do not change its total probability. Thus, every prefix sum is maximal, which gives $F^\star(s,i) \geqslant F^\tau(s,i)$ for any feasible τ. □

Potential Function. For a given adversary policy τ, we define the potential function $f_\tau(s,i)$ for a state $s \in \mathcal{S}$ and index $i \in [|Succ(s)| - 1]$ as:

$$f_\tau(s,i) := (F^\star(s,i) - F^\tau(s,i))(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}}). \tag{6}$$

First, we show that the potential function is non-negative for any adversary policy τ.

Lemma 13. *For any adversary policy τ, state $s \in \mathcal{S}$, and index $i \in [|Succ(s)| - 1]$, the potential function is non-negative, i.e., $f_\tau(s,i) \geqslant 0$.*

Proof. By Definition 2, the successor states are enumerated in descending order of their optimal values, meaning $\boldsymbol{v}^\star_{s_1} \geqslant \boldsymbol{v}^\star_{s_2} \geqslant \cdots \geqslant \boldsymbol{v}^\star_{s_{|Succ(s)|}}$. Consequently, for any valid index i, the difference between adjacent state values is non-negative:

$$\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}} \geqslant 0$$

Furthermore, by Lemma 12, the chosen optimal policy maximizes every cumulative transition probability. Thus, for all indices i:

$$F^\star(s,i) - F^\tau(s,i) \geqslant 0$$

The potential function is defined as the product of these two terms:

$$f_\tau(s,i) = (F^\star(s,i) - F^\tau(s,i))(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}})$$

Thus, it follows that $f_\tau(s,i) \geqslant 0$. This completes the proof. □

Now we first prove the following lemma, which lower bounds the value difference in terms of the potential function.

Lemma 14 (Lower Bound)**.** *For every policy τ, state $s \in \mathcal{S}$, and index $i \in [|Succ(s)| - 1]$, we have $\boldsymbol{v}^\star_s - \boldsymbol{v}^\tau_s \geqslant \gamma f_\tau(s, i)$.*

Proof. We start by bounding the value difference using the Bellman equations:

$$\begin{aligned}
\boldsymbol{v}^\star_s - \boldsymbol{v}^\tau_s &= (\mathbf{c}_s + \gamma \mathfrak{p}^{\star\top}_s \boldsymbol{v}^\star) - (\mathbf{c}_s + \gamma \mathfrak{p}^{\tau\top}_s \boldsymbol{v}^\tau) && \text{(Equation 4)} \\
&\geqslant (\mathbf{c}_s + \gamma \mathfrak{p}^{\star\top}_s \boldsymbol{v}^\star) - (\mathbf{c}_s + \gamma \mathfrak{p}^{\tau\top}_s \boldsymbol{v}^\star) && \text{(Since } \boldsymbol{v}^\star \geqslant \boldsymbol{v}^\tau \text{ by Lemma 10)} \\
&= \gamma(\mathfrak{p}^\star_s - \mathfrak{p}^\tau_s)^\top \boldsymbol{v}^\star. && \text{(Canceling } \mathbf{c}_s \text{ and factoring)}
\end{aligned}$$

To evaluate this dot product, we use summation by parts. Because both $\mathfrak{p}^\star_s$ and $\mathfrak{p}^\tau_s$ are valid probability distributions over the successor states, their total cumulative probabilities at the final index are both equal to 1. Therefore, their cumulative difference at n is zero: $F^\star(s, |Succ(s)|) - F^\tau(s, |Succ(s)|) = 1 - 1 = 0$. Applying summation by parts yields:

$$\begin{aligned}
\gamma(\mathfrak{p}^\star_s - \mathfrak{p}^\tau_s)^\top \boldsymbol{v}^\star &= \gamma \sum_{j=1}^{|Succ(s)|} (\mathfrak{p}^\star_{s,s_j} - \mathfrak{p}^\tau_{s,s_j}) \boldsymbol{v}^\star_{s_j} && \text{(Expanding)} \\
&= \gamma \Bigg[\sum_{i=1}^{|Succ(s)|-1} \Big((F^\star(s,i) - F^\tau(s,i))(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}}) \Big) \\
&\qquad + (F^\star(s, |Succ(s)|) - F^\tau(s, |Succ(s)|)) \boldsymbol{v}^\star_{s_{|Succ(s)|}} \Bigg] && \text{(Summation by parts)} \\
&= \gamma \sum_{i=1}^{|Succ(s)|-1} (F^\star(s,i) - F^\tau(s,i))(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}}) && \text{(Last term is 0)} \\
&= \gamma \sum_{i=1}^{|Succ(s)|-1} f_\tau(s,i). && \text{(Definition of } f_\tau(s,i))
\end{aligned}$$

Because each $f_\tau(s, i)$ is non-negative by Lemma 13, the total sum is bounded below by any single term $\gamma f_\tau(s, i)$. □

Now we show the following lemma, which upper bounds the value difference based on the potential function.

Lemma 15 (Upper Bound)**.** *Let $\hat{f}_\tau = \max_{s \in \mathcal{S}, i} f_\tau(s, i)$. Then, $\|\boldsymbol{v}^\star - \boldsymbol{v}^\tau\|_\infty \leqslant \frac{\gamma n}{1-\gamma} \hat{f}_\tau$.*

Proof. First, we relate the value difference to the transition difference:

$$\begin{aligned}
\boldsymbol{v}^\star - \boldsymbol{v}^\tau &= (\mathbb{I} - \gamma P^\star)^{-1}\mathbf{c} - (\mathbb{I} - \gamma P^\tau)^{-1}\mathbf{c} && \text{(Equation 5)} \\
&= (\mathbb{I} - \gamma P^\tau)^{-1} \left[(\mathbb{I} - \gamma P^\tau) - (\mathbb{I} - \gamma P^\star) \right] \\
&\quad \cdot (\mathbb{I} - \gamma P^\star)^{-1}\mathbf{c} && (A^{-1} - B^{-1} = B^{-1}(B - A)A^{-1}) \\
&= \gamma(\mathbb{I} - \gamma P^\tau)^{-1}(P^\star - P^\tau)\boldsymbol{v}^\star. && \text{(Simplifying terms and Equation 5)}
\end{aligned}$$

We take the infinity norm of both sides and apply the Neumann series expansion:

$$\begin{aligned}
\|\boldsymbol{v}^\star - \boldsymbol{v}^\tau\|_\infty &\leqslant \gamma \left\|(\mathbb{I} - \gamma P^\tau)^{-1}\right\|_\infty \|(P^\star - P^\tau)\boldsymbol{v}^\star\|_\infty && \text{(Submultiplicativity of } L_\infty \text{ norm)} \\
&\leqslant \gamma \left(\sum_{k=0}^{\infty} \gamma^k \left\|(P^\tau)^k\right\|_\infty \right) \|(P^\star - P^\tau)\boldsymbol{v}^\star\|_\infty && \text{(Neumann series, Triangle inequality)} \\
&= \frac{\gamma}{1-\gamma} \|(P^\star - P^\tau)\boldsymbol{v}^\star\|_\infty . && \text{(Since } P^\tau \text{ is stochastic, } \left\|(P^\tau)^t\right\|_\infty = 1)
\end{aligned}$$

Let $s \in \mathcal{S}$ be the state that maximizes the row sum in the remaining infinity norm. Applying summation by parts to this specific row yields:

$$\begin{aligned}
\|(P^\star - P^\tau)\boldsymbol{v}^\star\|_\infty &= (\mathfrak{p}_s^\star - \mathfrak{p}_s^\tau)^\top \boldsymbol{v}^\star && \text{(Definition of the infinity norm)} \\
&= \sum_{i=1}^{|Succ(s)|-1} f_\tau(s,i) && \text{(Summation by parts as in Lemma 14)} \\
&\leqslant |Succ(s)|\hat{f}_\tau && \text{(Since } f_\tau(s,i) \leqslant \hat{f}_\tau\text{)} \\
&\leqslant n\hat{f}_\tau && \text{(Since } |Succ(s)| \leqslant n\text{)}
\end{aligned}$$

Substituting this bounded row sum back into our norm inequality completes the proof. □

B.1.3 Halving of the Cumulative Gap

We now chain the two previous lemmas to show the following Lemma.

Lemma 16. *Consider two policies τ and τ'. Let $s \in \mathcal{S}$ and $i \in [|Succ(s)| - 1]$ be the state and index that maximize the potential function $f_\tau(s,i)$. If τ' satisfies the condition:*

$$F^\star(s,i) - F^{\tau'}(s,i) \geqslant \frac{1}{2}(F^\star(s,i) - F^\tau(s,i)), \tag{7}$$

Then we have the following inequality:

$$\left\|\boldsymbol{v}^\star - \boldsymbol{v}^{\tau'}\right\|_\infty \geqslant \frac{1-\gamma}{2n}\left\|\boldsymbol{v}^\star - \boldsymbol{v}^\tau\right\|_\infty. \tag{8}$$

Proof. We first show how the lemma's assumption translates to the potential function. By rearranging the assumed inequality and multiplying both sides by the non-negative value $2(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}})$, we obtain:

$$\begin{aligned}
2(F^\star(s,i) - F^{\tau'}(s,i))(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}}) &\geqslant (F^\star(s,i) - F^\tau(s,i))(\boldsymbol{v}^\star_{s_i} - \boldsymbol{v}^\star_{s_{i+1}}) \\
&\Rightarrow 2f_{\tau'}(s,i) \geqslant f_\tau(s,i). && \text{(Definition of } f\text{)}
\end{aligned}$$

The main proof follows by chaining the upper bound for policy τ (Lemma 15) with the lower bound for policy τ' (Lemma 14). We can structure this as a direct sequence of inequalities:

$$\begin{aligned}
\|\boldsymbol{v}^\star - \boldsymbol{v}^\tau\|_\infty &\leqslant \frac{\gamma n}{1-\gamma} f_\tau(s,i) && \text{(Upper bound for } \tau\text{, Lemma 15)} \\
&\leqslant \frac{2\gamma n}{1-\gamma} f_{\tau'}(s,i) && \text{(Based on Lemma's assumption)} \\
&\leqslant \frac{2n}{1-\gamma}(\boldsymbol{v}^\star_s - \boldsymbol{v}^{\tau'}_s) && \text{(Lower bound for } \tau'\text{, Lemma 14)} \\
&\leqslant \frac{2n}{1-\gamma}\left\|\boldsymbol{v}^\star - \boldsymbol{v}^{\tau'}\right\|_\infty. && \text{(Definition of the } L_\infty \text{ norm)}
\end{aligned}$$

Rearranging the first and last terms of this chain directly yields the claimed bound. □

The following lemma establishes that the difference $F^\star(s,i) - F^\tau(s,i)$ must decrease by at least a factor of two for at least one state-index pair (s,i) within a bounded number of iterations, $L := \lceil \log_\gamma\left(\frac{1-\gamma}{2n}\right)\rceil$, of Algorithm 2.

Lemma 17. *Let $\tau^0, \tau^1, \ldots$ be the sequence of policies generated by Algorithm 2. For a specific policy τ^t, let $(s,i) = \operatorname{argmax}_{s' \in \mathcal{S}, j \in [|Succ(s')|-1]} f_{\tau^t}(s',j)$ be the state and index that maximize the potential in τ^t. Then, for any subsequent policy τ^l with $l > t + L$, we have:*

$$F^\star(s,i) - F^{\tau^l}(s,i) \leqslant \frac{1}{2}\left(F^\star(s,i) - F^{\tau^t}(s,i)\right).$$

Proof. Let (s,i) be the state and index pair defined in the lemma. Assume, for the sake of contradiction, that there exists some $l > t + L$ such that:

$$F^\star(s,i) - F^{\tau^l}(s,i) > \frac{1}{2}\left(F^\star(s,i) - F^{\tau^t}(s,i)\right).$$

This assumption allows us to invoke Lemma 16, giving us a lower bound on the value error at step l. Simultaneously, the standard convergence property of policy iteration gives us an upper bound. We combine these bounds as follows:

$$\left\|\boldsymbol{v}^\star - \boldsymbol{v}^{\tau^l}\right\|_\infty \leqslant \gamma^{l-t}\left\|\boldsymbol{v}^\star - \boldsymbol{v}^{\tau^t}\right\|_\infty \qquad \text{(Upper bound via Lemma 11)}$$

$$\left\|\boldsymbol{v}^\star - \boldsymbol{v}^{\tau^l}\right\|_\infty \geqslant \frac{1-\gamma}{2n}\left\|\boldsymbol{v}^\star - \boldsymbol{v}^{\tau^t}\right\|_\infty \qquad \text{(Lower bound via Lemma 16 and our assumption)}$$

If $\left\|\boldsymbol{v}^{\tau^t} - \boldsymbol{v}^\star\right\|_\infty = 0$, then τ^t is optimal, and the algorithm terminates at iteration t. Because the sequence continues to $l > t$, we must have $\left\|\boldsymbol{v}^{\tau^t} - \boldsymbol{v}^\star\right\|_\infty > 0$. Combining these two inequalities and dividing by this strictly positive term yields:

$$\gamma^{l-t} \geqslant \frac{1-\gamma}{2n} \qquad \text{(Canceling the positive value error norm)}$$

$$l - t \leqslant \log_\gamma\left(\frac{1-\gamma}{2n}\right) \qquad \text{(Taking } \log_\gamma\text{, flipping the inequality since } \gamma < 1\text{).}$$

This implies $l - t \leqslant L$, which directly contradicts our initial premise that $l > t + L$. Thus, the assumption must be false, meaning the gap $F^\star(s,i) - F^\tau(s,i)$ must have decreased by at least half. □

B.1.4 Combinatorial Bound on the Number of Halvings

The previous Lemma showed that after L steps of Algorithm 2, the difference $F^\star(s,i) - F^\tau(s,i)$ decreases by at least a factor of 2. Now we need to show that this decrease only happens at most a polynomial number of iterations. We can show that in fact this only happens $\mathcal{O}(n \log n)$ times. We thus have the following Lemma, which was shown in [ACG+26a], and we mention it for completeness.

Lemma 18. *Let c be a constant positive integer and let W be a finite set of non-negative real numbers. We define the set of all bounded-coefficient subset sums of W as:*

$$\mathcal{A}(W) = \left\{ \left| \sum_{w \in W} d_w w \right| \;\middle|\; d_w \in \{-c, \cdots, -1, 0, 1, \cdots, c\} \right\}.$$

Additionally, for any set of positive reals Y, let the degree $\mathrm{Deg}(Y)$ denote the number of unique Maximum Significant bits (MSBs) of the set Y where numbers are written in binary, formally given by:

$$\mathrm{Deg}(Y) = \left|\{\lfloor \log_2 y \rfloor \mid y \in Y\}\right|.$$

Then, for any such set W, the number of distinct MSBs in $\mathcal{A}(W)$ is:

$$\mathrm{Deg}(\mathcal{A}(W)\backslash\{0\}) = \mathcal{O}(|W| \log |W|).$$

Proof. We show this using a proof by [ACG+26a]; we present the full proof here for completeness. Throughout this proof, all logarithms are taken in base 2. The strategy proceeds in three steps: (i) we introduce a notion of bounded-coefficient linear independence and show that real numbers with sufficiently separated MSBs satisfy it; (ii) we use Siegel's lemma to extract a small-coefficient linear relation among any $\Theta(|W| \log |W|)$ elements of $\mathcal{A}(W)$; (iii) we combine these via a greedy selection on MSBs to derive a contradiction whenever $\mathrm{Deg}(\mathcal{A}(W)\backslash\{0\})$ exceeds the claimed bound.

Step 1: MSB separation implies linear independence. For a positive integer c', we call reals $y_1, \ldots, y_N$ *c'-linearly independent* if $\sum_{i=1}^N x_i y_i \neq 0$ for every non-zero $\underline{x} \in \mathbb{Z}^N$ with $\|\underline{x}\|_\infty \leqslant c'$.

Claim 1. *There exists a positive integer $D = D(c')$ such that for any $N \geqslant 1$, any non-zero reals $y_1, \ldots, y_N$ satisfying*

$$\lfloor \log_2 |y_{i+1}| \rfloor \geqslant \lfloor \log_2 |y_i| \rfloor + D \quad \textit{for } 1 \leqslant i < N$$

are c'-linearly independent.

Proof of Claim 1. Choose D large enough that $2^{D-1} > 2c'$ (e.g., $D = \lceil \log_2 c' \rceil + 3$). The MSB-separation hypothesis implies $|y_{i+1}|/|y_i| > 2^{D-1} > 2c'$ for every i, so in particular $|y_{i+1}| > 2|y_i|$. Iterating yields $|y_{N-1}| > \sum_{i=1}^{N-2} |y_i|$ via a geometric sum, hence

$$|y_N| > 2c'|y_{N-1}| > c' \sum_{i=1}^{N-1} |y_i|.$$

Suppose $\sum_{i=1}^{N} x_i y_i = 0$ with $\|\underline{x}\|_\infty \leqslant c'$. If $x_N \neq 0$, then $|x_N| \geqslant 1$ and

$$|x_N y_N| \geqslant |y_N| > c' \sum_{i=1}^{N-1} |y_i| \geqslant \left| \sum_{i=1}^{N-1} x_i y_i \right|,$$

contradicting $x_N y_N = -\sum_{i<N} x_i y_i$. Hence $x_N = 0$, and induction on N gives $\underline{x} = \underline{0}$. □

Step 2: Siegel's lemma. We use the following standard application of Siegel's lemma (see, e.g., [HS00]). There exist absolute constants $n_0 \geqslant 4$ and $C_0 > 0$ such that for every integer $n \geqslant n_0$, every real $c \geqslant 1$, and every integer matrix A with n columns, N rows satisfying $n \log_2 n - 1 \leqslant N \leqslant n \log_2 n$, and $\|A\|_\infty \leqslant c$, there exists a non-zero $\underline{x} \in \mathbb{Z}^N$ in the left-kernel of A with

$$\|\underline{x}\|_\infty \leqslant C_0 \cdot c.$$

This follows from the classical bound $\|\underline{x}\|_\infty \leqslant (\|A\|_\infty N)^{n/(N-n)}$. For $n \geqslant n_0$ sufficiently large, we have $n/(N-n) \leqslant 1/(\log_2 n - 1 - 1/n) \leqslant 1$, which justifies the inequality $(cN)^{n/(N-n)} \leqslant c \cdot N^{n/(N-n)}$ since $c \geqslant 1$. The remaining factor $N^{n/(N-n)}$ converges to 2 as $n \to \infty$ (an absolute constant independent of c), and is therefore uniformly bounded above by C_0 for all $n \geqslant n_0$. We may also enlarge n_0 to ensure $n_0 \log_2 n_0 - 1 \geqslant n_0$, so that the row count N strictly exceeds the column count n and the kernel is guaranteed to be non-trivial.

Step 3: Combining the steps. Set $c' := \lceil C_0 c \rceil$ and let $D := D(c')$ be the constant from Claim 1; both depend only on c. Let $K := \mathrm{Deg}(\mathcal{A}(W)\backslash\{0\})$.

Base case. If $|W| < n_0$, then $|\mathcal{A}(W)| \leqslant (2c+1)^{|W|} < (2c+1)^{n_0}$, which is a constant depending only on c. In particular, K is bounded by this constant, and the lemma holds trivially. Henceforth assume $|W| \geqslant n_0$.

Inductive case. We show $K \leqslant D \cdot |W| \log_2 |W|$. Assume for contradiction that $K > D \cdot |W| \log_2 |W|$. Sort the distinct MSBs of $\mathcal{A}(W)\backslash\{0\}$ as $m_1 < m_2 < \cdots < m_K$. Picking every D-th MSB yields $\lceil K/D \rceil > |W| \log_2 |W|$ MSBs with consecutive gaps at least D. Truncate to $N := \lfloor |W| \log_2 |W| \rfloor$ such MSBs, so $|W| \log_2 |W| - 1 \leqslant N \leqslant |W| \log_2 |W|$, and pick a representative $y_j \in \mathcal{A}(W)\backslash\{0\}$ with $\lfloor \log_2 y_j \rfloor$ equal to the j-th selected MSB. By Claim 1, $y_1, \ldots, y_N$ are c'-linearly independent.

On the other hand, by definition of $\mathcal{A}(W)$, each y_j admits a representation

$$y_j = \left| \sum_{w \in W} d_{j,w} w \right| \quad \text{with } d_{j,w} \in \{-c, \ldots, c\}.$$

Letting $\sigma_j \in \{-1, +1\}$ be such that $y_j = \sigma_j \sum_w d_{j,w} w$, and setting $\tilde{d}_{j,w} := \sigma_j d_{j,w}$, we still have $|\tilde{d}_{j,w}| \leqslant c$ and $y_j = \sum_{w \in W} \tilde{d}_{j,w} w$. Let $A \in \mathbb{Z}^{N \times |W|}$ be the matrix with entries $A_{j,w} = \tilde{d}_{j,w}$; then $\|A\|_\infty \leqslant c$. Since $|W| \geqslant n_0$ and $|W| \log_2 |W| - 1 \leqslant N \leqslant |W| \log_2 |W|$, Step 2 yields a non-zero $\underline{x} \in \mathbb{Z}^N$ with $\|\underline{x}\|_\infty \leqslant C_0 c \leqslant c'$ such that $A^\top \underline{x} = 0$, i.e., $\sum_{j=1}^{N} x_j \tilde{d}_{j,w} = 0$ for every $w \in W$. Consequently,

$$\sum_{j=1}^{N} x_j y_j = \sum_{j=1}^{N} x_j \sum_{w \in W} \tilde{d}_{j,w} w = \sum_{w \in W} w \sum_{j=1}^{N} x_j \tilde{d}_{j,w} = 0,$$

contradicting the c'-linear independence of $y_1, \ldots, y_N$. Therefore $K \leqslant D \cdot |W| \log_2 |W| = \mathcal{O}(|W| \log |W|)$. □

Finally, we will show that Algorithm 2 is strongly polynomial for L_1 and L_∞ uncertainty sets. We prove Theorem 2 by showing it separately for L_1 and L_∞ uncertainty sets.

Proof of Theorem 2 for L_1 uncertainty sets. For any state $s \in \mathcal{S}$, Property 1 describes the structure of the transition probabilities generated by Algorithm 3. Probability mass is shifted to the highest-value state by fully taking mass from a set of states $Z_{\mathfrak{p}}$ and partially taking mass from at most one state $I_{\mathfrak{p}}$. The mass taken from $I_{\mathfrak{p}}$ is the leftover budget: $\frac{\delta_s}{2} - \sum_{z \in Z_{\mathfrak{p}}} \hat{\mathfrak{p}}_{s,z}$.

When we calculate the cumulative probability $F^{\tau}(s,i)$ over any prefix set S_i, the sum simplifies algebraically. If $I_{\mathfrak{p}} \in S_i$, the $-\frac{\delta_s}{2}$ from the partial term cancels the receiver's $+\frac{\delta_s}{2}$, while the remaining $\sum_{z \in Z_{\mathfrak{p}}} \hat{\mathfrak{p}}_{s,z}$ adds each depleted state's nominal with coefficient $+1$ (regardless of whether $z \in S_i$). This ensures that $F^{\tau}(s,i)$ is always a linear combination of the nominal probabilities $\hat{\mathfrak{p}}_{s,j}$ and the budget $\frac{\delta_s}{2}$, using only the coefficients $\{-2,-1,0,1,2\}$.

Therefore, the difference $F^{\star}(s,i) - F^{\tau}(s,i)$ strictly belongs to the set of unitary signed subset sums $\mathcal{A}(W_s)$, which is defined over the base set $W_s = \{\hat{\mathfrak{p}}_{s,j} \mid j \in \mathcal{S}\} \cup \left\{\frac{\delta_s}{2}, 1\right\}$.

By Lemma 18, the number of distinct MSBs for the non-zero elements in $\mathcal{A}(W_s)$ is bounded by $\mathcal{O}(|W_s| \log |W_s|)$. Because the size of the base set is $|W_s| \leqslant n+2$, the total number of possible MSBs is bounded by $\mathcal{O}(n \log n)$.

From Lemma 17, we know that every $L = \lceil \log_{\gamma}\left(\frac{1-\gamma}{2n}\right) \rceil$ iterations, the maximum cumulative probability gap $F^{\star}(s,i) - F^{\tau}(s,i)$ must decrease by at least half. Since the optimal values are fixed, this halving directly corresponds to dropping to a lower MSB within the set $\mathcal{A}(W_s)$.

There are at most n^2 state-index pairs (s,i), and for any specific pair the gap $F^{\star}(s,i) - F^{\tau}(s,i)$ can drop its MSB at most $\mathcal{O}(n \log n)$ times. Combining with the L iterations per drop yields $n^2 \cdot \mathcal{O}(n \log n) \cdot L = \mathcal{O}(n^3 \log n \cdot L)$. □

Proof of Theorem 2 for L_∞ uncertainty sets. For any state $s \in \mathcal{S}$, Property 2 describes the structure of the transition probabilities generated by Algorithm 4. Probability mass is moved between states under strict L_∞ constraints. As a result, the final probability $\mathfrak{p}_{s,s'}$ for any successor state is always formed by adding or subtracting the budget δ_s from the nominal probabilities $\hat{\mathfrak{p}}_{s,s'}$, or by capping at the limits 0 or 1.

Therefore, the difference between the optimal probability and the current policy's probability, $\mathfrak{p}^{\star}_{s,s'} - \mathfrak{p}^{\tau}_{s,s'}$, strictly belongs to the set of signed subset sums $\mathcal{A}(X_s)$ with coefficients in $\{-2,-1,0,1,2\}$. This set is defined over the base set $X_s = \{\hat{\mathfrak{p}}_{s,j} \mid j \in \mathcal{S}\} \cup \{1\} \cup \{\delta_s, 2 \cdot \delta_s, \cdots, n \cdot \delta_s\}$.

The multiples $2\delta_s, \ldots, n\delta_s$ are included because the at-most-one incomplete coordinate, by conservation $\sum_{s'} \mathfrak{p}^{\tau}_{s,s'} = 1$, contributes a coefficient $|D_{\mathfrak{p}}| - |R_{\mathfrak{p}}| \in \{-n, \ldots, n\}$ on δ_s. Hence the cumulative difference $F^{\star}(s,i) - F^{\tau}(s,i)$ may carry a coefficient up to $\pm 2n$ on δ_s, which we encode using at most two base elements $k \cdot \delta_s$ with $k \leqslant n$, each with a coefficient in $\{-2, \ldots, 2\}$. This keeps the coefficient bound in $\mathcal{A}(X_s)$ a constant, as Lemma 18 requires.

By Lemma 18, the number of distinct MSBs for the non-zero elements in $\mathcal{A}(X_s)$ is bounded by $\mathcal{O}(|X_s| \log |X_s|)$. Because the size of the base set is $|X_s| \leqslant 2n+1$, the total number of possible MSBs is strictly bounded by $\mathcal{O}(n \log n)$.

From our convergence analysis, we know that every $L = \lceil \log_{\gamma}\left(\frac{1-\gamma}{2n}\right) \rceil$ iterations, the maximum cumulative probability gap $F^{\star}(s,i) - F^{\tau}(s,i)$ must decrease by at least half. Since the optimal values remain fixed, this halving directly corresponds to dropping to a lower MSB within the set $\mathcal{A}(X_s)$.

There are at most n^2 possible state-index pairs (s,i) that can dictate this maximum cumulative probability gap. For any specific pair, the associated probability discrepancy can drop its MSB at most $\mathcal{O}(n \log n)$ times. Because it takes at most L steps to force a drop, the total number of policy iteration steps before termination is bounded by $n^2 \cdot \mathcal{O}(n \log n) \cdot L = \mathcal{O}(n^3 \log n \cdot L)$. This represents a polynomial number of iterations, completing the proof. □

C Proof of Theorem 3: Complexity of RMC with L_p Uncertainty Sets

In this section, we analyze the complexity of the Problem DiscRMC with L_p uncertainty sets for any constant integer $p > 1$. Since the goal of this section is to show the complexity bounds for

this problem, we consider the Turing model of computation, where all the numbers in the input are assumed to be rational.

First, we analyze the p-ROOT-SUM complexity class. We prove that this class is polynomially equivalent to the class of $\frac{p}{q}$-ROOT-SUM where $\gcd(p, q) = 1$. Second, we construct a polynomial-time reduction from the $\frac{p}{p-1}$-ROOT-SUM problem to the decision version of the main problem, which shows that it is p-ROOT-SUM-hard. Finally, we establish an upper bound on the complexity of the problem. We show that it belongs to the class $\exists\mathbb{R}$. We achieve this by modeling the main problem (computing the exact values and optimal strategy) as an $\exists\mathbb{R}$ formula. The corresponding decision problem can then be easily solved by adding a threshold constraint to this formula.

C.1 Equivalence of ROOT-SUM Problems

Our main objective in this subsection is to prove that the p-ROOT-SUM problem and the $\frac{p}{q}$-ROOT-SUM problem (where $\gcd(p, q) = 1$) are polynomial-time equivalent.

First, we show that the p-ROOT-SUM problem is reducible to the kp-ROOT-SUM problem. Next, we introduce a polynomial-time greedy decomposition algorithm that expresses any integer as a sum of p-th powers using a logarithmically bounded number of terms. Then, we construct a formal reduction from the p-ROOT-SUM problem to the $\frac{p}{q}$-ROOT-SUM problem. Finally, Lemma 22 combines these steps to conclude the polynomial-time equivalence of these classes.

In the following lemma, we show that the kp-ROOT-SUM problem is harder than p-ROOT-SUM problem for any rational number p and any constant integer $k > 1$:

Lemma 19. *For any rational number p and any constant integer $k > 1$, the p-ROOT-SUM problem is polynomial-time reducible to the kp-ROOT-SUM problem.*

Proof. Let an instance of the p-ROOT-SUM problem be given by a sequence of positive integers $a_1, \ldots, a_n$ and an integer threshold α. We need to decide if the following holds:

$$\sum_{i=1}^{n} \sqrt[p]{a_i} \geqslant \alpha$$

We construct an instance of the kp-ROOT-SUM problem. For each index i, we define a new value $a'_i = a_i^k$. Because a_i and k are integers, a'_i is also an integer.

By the properties of exponents, we have:

$$\sqrt[kp]{a'_i} = (a'_i)^{\frac{1}{kp}} = (a_i^k)^{\frac{1}{kp}} = a_i^{\frac{k}{kp}} = a_i^{\frac{1}{p}} = \sqrt[p]{a_i} \Rightarrow \sum_{i=1}^{n} \sqrt[kp]{a'_i} = \sum_{i=1}^{n} \sqrt[p]{a_i}$$

Therefore, the condition $\sum_{i=1}^{n} \sqrt[kp]{a'_i} \geqslant \alpha$ is true if and only if $\sum_{i=1}^{n} \sqrt[p]{a_i} \geqslant \alpha$.

Because k is a constant, computing a_i^k for all n terms takes polynomial time and has a polynomial number of bits. Therefore, the reduction is polynomial. □

Now we need an efficient way to write integers as a sum of powers of p. The following lemma provides a polynomial-time algorithm to express any integer as a sum of p-th powers.

Lemma 20. *Let p be a constant positive integer greater than zero. Any integer n can be written as a sum $n = \sum_{i=1}^{k} a_i^p$ for some positive integers a_i, where the number of terms $k = O(\log \log n)$. Furthermore, the sequence of integers a_i can be computed in time polynomial in the bit-size of n.*

Proof. We construct the sequence a_i using the greedy approach by Algorithm 5.

Algorithm 5 Greedy p-th Power Decomposition

Input: Integer $n > 0$, Integer $p > 1$
Output: Sequence $(a_1, a_2, \ldots, a_k)$ such that $\sum_{i=1}^{k} a_i^p = n$
$n_0 \leftarrow n$
$j \leftarrow 0$
while $n_j > 0$ **do**
    $a_{j+1} \leftarrow \lfloor n_j^{1/p} \rfloor$ ▷ Find the largest integer root
    $n_{j+1} \leftarrow n_j - a_{j+1}^p$ ▷ Update the remainder
    $j \leftarrow j + 1$
$k \leftarrow j$
return $(a_1, a_2, \ldots, a_k)$

Let $n_0 = n$. For each step $j \geqslant 0$, if $n_j > 0$, the algorithm chooses the largest integer $a_{j+1} = \lfloor n_j^{1/p} \rfloor$ such that $a_{j+1}^p \leqslant n_j$, and updates the remainder to $n_{j+1} = n_j - a_{j+1}^p$. We bound n_{j+1} as follows:

$$\begin{aligned}
n_{j+1} &= n_j - a_{j+1}^p && \text{(Definition of the } n_{j+1}\text{)} \\
&< (a_{j+1}+1)^p - a_{j+1}^p && \text{(Since } (a_{j+1}+1)^p > n_j \geqslant a_{j+1}^p\text{)} \\
&= \sum_{\ell=0}^{p-1} \binom{p}{\ell} a_{j+1}^{\ell} && \text{(Binomial expansion)} \\
&\leqslant \left(\sum_{\ell=0}^{p-1} \binom{p}{\ell} \right) a_{j+1}^{p-1} && \text{(Since } a_{j+1} \geqslant 1\text{, we have } a_{j+1}^{\ell} \leqslant a_{j+1}^{p-1}\text{)} \\
&< 2^p a_{j+1}^{p-1} && \text{(Sum of binomial coefficients)} \\
&\leqslant 2^p n_j^{\frac{p-1}{p}} && \text{(Since } a_{j+1} \leqslant n_j^{\frac{1}{p}}\text{)} \\
\Rightarrow \log_2 n_{j+1} &\leqslant p + \frac{p-1}{p} \log_2 n_j. && \text{(Properties of logarithms)} \\
\Rightarrow \log_2 n_{j+1} - p^2 &\leqslant \frac{p-1}{p} (\log_2 n_j - p^2). && \text{(Subtracting } p^2 \text{ from both sides)}
\end{aligned}$$

This shows that $\log_2 n_j - p^2$ decreases geometrically by the constant factor $0 < \frac{p-1}{p} < 1$. Therefore, the number of steps required for $\log_2 n_j$ to drop below the constant $p^2 + 1$ is $O(\log \log n_j)$.

Once $\log_2 n_j \leqslant p^2 + 1$, we have $n_j \leqslant 2^{p^2+1}$. Since n_j is an integer that decreases by at least 1 in each step, the algorithm will reach 0 in at most 2^{p^2+1} steps. Since p is a constant, this takes $O(1)$ steps. Because $n_0 = n$, the total number of terms $k = O(\log \log n) + O(1) = O(\log \log n)$.

Finally, at each step j, computing the largest integer root $a_{j+1} = \lfloor n_j^{1/p} \rfloor$ takes polynomial time with respect to the bit-size of n_j. Because there are $O(\log \log n)$ steps, the entire sequence a_i is computed in polynomial time. □

Next, we show that we can change the root from p to p/q without changing the complexity of the problem. The following lemma provides a polynomial-time reduction for this step.

Lemma 21. *For any constant positive integers p and q such that $\gcd(p, q) = 1$, the p-ROOT-SUM problem is polynomial-time reducible to the $\frac{p}{q}$-ROOT-SUM problem.*

Proof. Because $\gcd(p, q) = 1$, modular arithmetic guarantees there exists a positive integer $k < q$ such that q divides $kp + 1$. Because $k < q$, we can treat k as a constant.

Let an instance of the p-ROOT-SUM problem be given by a sequence of positive integers $a_1, \ldots, a_n$ and an integer threshold α. We want to decide if:

$$\sum_{i=1}^{n} a_i^{\frac{1}{p}} \geqslant \alpha$$

We multiply both sides of the inequality by $w := \prod_{j=1}^{n} a_j^k$. We rewrite the new term $w \cdot a_i^{\frac{1}{p}}$ as follows:

$$\begin{aligned}
w \cdot a_i^{\frac{1}{p}} &= z_i \cdot a_i^k \cdot a_i^{\frac{1}{p}} && \text{(Factoring out } z_i := \prod_{j \neq i} a_j^k \text{ from } w) \\
&= z_i \cdot a_i^{\frac{kp+1}{p}} && \text{(Combining the exponents of } a_i) \\
&= z_i \left(a_i^{\frac{kp+1}{q}} \right)^{\frac{q}{p}} && \text{(Properties of exponents)} \\
&= z_i y_i^{\frac{q}{p}}, && \text{(Substituting integer } y_i := a_i^{\frac{kp+1}{q}})
\end{aligned}$$

where y_i is an integer because q divides $kp + 1$.

By Lemma 20, we can decompose each integer z_i into a sum of q-th powers of integers $c_{i,j}$. This gives $z_i = \sum_{j=1}^{m_i} c_{i,j}^q$, where the number of terms m_i is in $O(\log \log z_i)$. We define $b_{i,j} := c_{i,j}^p y_i$ and simplify the main inequality:

$$\begin{aligned}
\sum_{i=1}^{n} w \cdot a_i^{\frac{1}{p}} \geqslant w\alpha &\iff \sum_{i=1}^{n} z_i y_i^{\frac{q}{p}} \geqslant w\alpha && \text{(Substituting the term equivalence)} \\
&\iff \sum_{i=1}^{n} \left(\sum_{j=1}^{m_i} c_{i,j}^q \right) y_i^{\frac{q}{p}} \geqslant w\alpha && \text{(Applying Lemma 20 to decompose } z_i) \\
&\iff \sum_{i=1}^{n} \sum_{j=1}^{m_i} \left(c_{i,j}^p y_i \right)^{\frac{q}{p}} \geqslant w\alpha && \text{(Distributing } y_i^{\frac{q}{p}} \text{ inside the inner sum)} \\
&\iff \sum_{i=1}^{n} \sum_{j=1}^{m_i} b_{i,j}^{\frac{q}{p}} \geqslant w\alpha. && \text{(Substituting } b_{i,j})
\end{aligned}$$

This final inequality is an instance of the $\frac{p}{q}$-ROOT-SUM problem with the threshold $w\alpha$.

Because k is a constant, computing and storing w, z_i, y_i takes polynomial time and memory in the bit-size of a_i. Furthermore, the number of new terms is $O(n \log \log z_i)$. By Lemma 20 $c_{i,j}$ is computed in polynomial time. Finally, $b_{i,j}$ can be computed by the arithmetic operations. Therefore, this reduction is valid and runs in polynomial time. □

By combining the previous reductions, we can establish the equivalence result for this subsection. This shows that the denominator in the fractional root in the p-ROOT-SUM problem does not affect the complexity.

Lemma 22. *For any constant positive integers p and q such that $\gcd(p, q) = 1$, the p-ROOT-SUM problem and the $\frac{p}{q}$-ROOT-SUM problem are polynomial-time equivalent. In other words, the denominator does not matter in the complexity.*

Proof. By Lemma 19, the $\frac{p}{q}$-ROOT-SUM problem is polynomial-time reducible to the p-ROOT-SUM problem. Similarly, because $\gcd(p, q) = 1$, by Lemma 21 the p-ROOT-SUM problem is polynomial-time reducible to the $\frac{p}{q}$-ROOT-SUM problem. Therefore, they are polynomial-time equivalent. □

C.2 Proving Hardness via Reduction

By Lemma 22, the p-ROOT-SUM problem is polynomial-time equivalent to the $\frac{p}{p-1}$-ROOT-SUM problem. We now formalize the reduction from the $\frac{p}{p-1}$-ROOT-SUM problem to the decision version of the problem DiscRMC with L_p uncertainty sets. Formally, we do the following reduction:

Given an integer constant $p \geqslant 2$, positive integers $a_1, \ldots, a_n$, and an integer threshold α, the $\frac{p}{p-1}$-ROOT-SUM problem asks to decide if $\sum_{i=1}^{n} a_i^{\frac{p-1}{p}} \geqslant \alpha$. We reduce this problem to deciding if the

optimal value $\boldsymbol{v}^\star(s_0) \geqslant \lambda$ in an RMC $\mathcal{M} = (\mathcal{S}, c, \mathit{Succ}, \mathcal{P})$ with L_p uncertainty sets and a constant discount factor $\gamma \in (0, 1)$.

Reduction: First, we introduce the following scaled variables to construct the RMC:

- $\forall i \in [n] : b_i := 2^p a_i$.
- $\forall i \in [n] : x_i := b_i/2 = 2^{p-1} a_i$.
- $K := 2^{p-1}\alpha$.

By Lemma 20, we write x_i as the sum of p-th powers of positive integers: $x_i = \sum_{k=1}^{M_i} u_{i,k}^p$. Let $M^* = \max_i M_i$. We pad each sequence $u_{i,k}$ with zeros so each sequence has exactly M^* terms. This preserves the sum. By Lemma 20, we know that $M^* = O(\log\log a_i) = O(\log\log b_i)$.

We construct the RMC $\mathcal{M} = (\mathcal{S}, c, \mathit{Succ}, \mathcal{P})$ as follows:

- $\mathcal{S} := \{s_0\} \cup \{s_1, \ldots, s_n\} \cup \bigcup_{i=1}^n \{\bot_{i,1}^+, \ldots, \bot_{i,M^*}^+, \bot_{i,1}^-, \ldots, \bot_{i,M^*}^-\}$. The states consist of an initial state s_0, n transient states s_i, and $2M^*$ absorbing states for each transient state.
- $c\colon \mathcal{S} \to \mathbb{R}$:
$$c(s) := \begin{cases} (1-\gamma)u_{i,k}^{p-1} & \text{if } s = \bot_{i,k}^+, \\ -(1-\gamma)u_{i,k}^{p-1} & \text{if } s = \bot_{i,k}^-, \\ 0 & \text{if } s \in \{s_0, s_1, \ldots, s_n\}. \end{cases}$$
- $\mathit{Succ}\colon \mathcal{S} \to 2^{\mathcal{S}}$ and $\mathcal{P}\colon \mathcal{S} \to 2^{\Delta(\mathcal{S})}$.
The successor mapping and L_p uncertainty sets (defined by a nominal probability $\hat{\mathfrak{p}}_s$ and radius δ_s) are defined for each state class:
  - *Initial State:* $\mathit{Succ}(s_0) := \{s_1, \ldots, s_n\}$. The nominal distribution $\hat{\mathfrak{p}}_{s_0}$ is uniform over the transient states ($\hat{\mathfrak{p}}_{s_0}(s_i) = \frac{1}{n}$) with zero uncertainty ($\delta_{s_0} := 0$).
  - *Transient States:* $\mathit{Succ}(s_i) := \{\bot_{i,1}^+, \ldots, \bot_{i,M^*}^+, \bot_{i,1}^-, \ldots, \bot_{i,M^*}^-\}$. From s_i to its $2M^*$ absorbing states, the nominal probability $\hat{\mathfrak{p}}_{s_i}$ is uniform ($1/2M^*$). We define the fixed uncertainty radius for these states as $\delta := \frac{1}{2M^*}$, so $\delta_{s_i} := \delta$.
  - *Absorbing States:* $\mathit{Succ}(\bot_{i,k}^+) := \{\bot_{i,k}^+\}, \mathit{Succ}(\bot_{i,k}^-) := \{\bot_{i,k}^-\}$. Each absorbing state deterministically transitions to itself, with zero uncertainty ($\delta_{\bot_{i,k}^+} := 0, \delta_{\bot_{i,k}^-} := 0$).
- $\lambda := \frac{\gamma^2 \delta K}{n}$.

Now, we analyze the values of the states in RMC $\mathcal{M}$. We first calculate the values for the absorbing states.

Lemma 23. *For the absorbing states $\bot_{i,k}^+$ and $\bot_{i,k}^-$, their values are $u_{i,k}^{p-1}$ and $-u_{i,k}^{p-1}$, respectively.*

Proof. We first prove the claim for the positive absorbing state $\bot_{i,k}^+$. By construction, the absorbing state deterministically transitions to itself because its uncertainty radius is zero. Thus, the Bellman equation simplifies to:

$$\begin{aligned} & \boldsymbol{v}^\star(\bot_{i,k}^+) = c(\bot_{i,k}^+) + \gamma \boldsymbol{v}^\star(\bot_{i,k}^+) && \text{(Bellman equation for absorbing states)} \\ \implies & \boldsymbol{v}^\star(\bot_{i,k}^+) = \frac{c(\bot_{i,k}^+)}{1-\gamma} && \text{(Rearranging for } \boldsymbol{v}^\star\text{)} \\ & = \frac{(1-\gamma)u_{i,k}^{p-1}}{1-\gamma} && \text{(Substituting the defined cost)} \\ & = u_{i,k}^{p-1}. && \text{(Canceling the } 1-\gamma \text{ terms)} \end{aligned}$$

By symmetry, substituting the cost $c(\bot_{i,k}^-) = -(1-\gamma)u_{i,k}^{p-1}$ for the negative absorbing state into the same derivation yields $\boldsymbol{v}^\star(\bot_{i,k}^-) = -u_{i,k}^{p-1}$. □

Using the values of the absorbing states, we can determine the value of the transient states. The following lemma calculates this value.

Lemma 24. *For the transient state s_i, let $\boldsymbol{v}^{(i)}$ be the vector of values of its absorbing states. Then, $\boldsymbol{v}^\star(s_i) = \gamma\delta\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}$.*

Proof. The Bellman equation for s_i yields:

$$\begin{aligned}
\boldsymbol{v}^\star(s_i) &= c(s_i) + \gamma \max_{\tau\in\mathcal{P}(s_i)} \tau^\top \boldsymbol{v}^{(i)} && \text{(Definition of the Bellman equation)}\\
&= \gamma \max_{\tau\in\mathcal{P}(s_i)} \tau^\top \boldsymbol{v}^{(i)} && (c(s_i) = 0)\\
&= \gamma \max_{\substack{d\in\mathbb{R}^{2M*}\\ \|d\|_p\leqslant\delta\\ \mathbf{1}^\top d=0\\ \hat{\mathfrak{p}}_{s_i}+d\geqslant 0}} (\hat{\mathfrak{p}}_{s_i} + d)^\top \boldsymbol{v}^{(i)} && (\mathcal{P}(s_i) \text{ is an } L_p \text{ uncertainty set})\\
&= \gamma \max_{\substack{d\in\mathbb{R}^{2M*}\\ \|d\|_p\leqslant\delta\\ \mathbf{1}^\top d=0\\ \hat{\mathfrak{p}}_{s_i}+d\geqslant 0}} d^\top \boldsymbol{v}^{(i)} && (\hat{\mathfrak{p}}_{s_i}^\top \boldsymbol{v}^{(i)} = 0 \text{ as } \hat{\mathfrak{p}}_{s_i} \text{ is uniform and } \boldsymbol{v}^\star(\perp_{i,k}^+) + \boldsymbol{v}^\star(\perp_{i,k}^-) = 0)
\end{aligned}$$

Because $\frac{p-1}{p} + \frac{1}{p} = 1$, Hölder's inequality states that the inner product is bounded from above [Fol99]:

$$d^\top \boldsymbol{v}^{(i)} \leqslant \|d\|_p \|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}} \leqslant \delta\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}.$$

To prove this upper bound is achievable, we construct the optimal perturbation $d^\star$:

$$d_j^\star = \delta \frac{\operatorname{sign}(\boldsymbol{v}_j^{(i)})|\boldsymbol{v}_j^{(i)}|^{\frac{1}{p-1}}}{\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}^{\frac{1}{p-1}}}$$

This vector satisfies all constraints:

- ($\|d^\star\|_p = \delta$): By the definition of the p norm we have:

  $$\|d^\star\|_p^p = \sum_j |d_j^\star|^p = \frac{\delta^p}{\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}^{\frac{p}{p-1}}} \sum_j |\boldsymbol{v}_j^{(i)}|^{\frac{p}{p-1}}$$

  By the definition of the $L_{\frac{p}{p-1}}$ norm, the sum is equal to the denominator. Thus, $\|d^\star\|_p^p = \delta^p$. Therefore, $\|d^\star\|_p = \delta$,
- ($\mathbf{1}^\top d^\star = 0$): Since $|\boldsymbol{v}_j^{(i)}|^{\frac{1}{p-1}} = (u_{i,k}^{p-1})^{\frac{1}{p-1}} = u_{i,k}$, and $\boldsymbol{v}^{(i)}$ is symmetric, the components of $d^\star$ form canceling pairs $\pm u_{i,k}$.
- $\hat{\mathfrak{p}}_{s_i} + d^\star \geqslant 0$: Since $\|d^\star\|_p = \delta$, and $\|d^\star\|_\infty \leqslant \|d^\star\|_p$, the maximum negative perturbation is at least $-\delta$. Because $\hat{\mathfrak{p}}_{s_i,j} = \frac{1}{2M*} = \delta$, the minimum possible probability is 0.

Finally, we evaluate the inner product to confirm the maximum bound:

$$\begin{aligned}
d^{\star\top}\boldsymbol{v}^{(i)} &= \sum_{j=1}^{2M*} \left(\delta \frac{\operatorname{sign}(\boldsymbol{v}_j^{(i)})|\boldsymbol{v}_j^{(i)}|^{\frac{1}{p-1}}}{\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}^{\frac{1}{p-1}}}\right)\boldsymbol{v}_j^{(i)} && \text{(Expanding the dot product)}\\
&= \frac{\delta}{\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}^{\frac{1}{p-1}}} \sum_{j=1}^{2M*} |\boldsymbol{v}_j^{(i)}|^{\frac{p}{p-1}} && (\operatorname{sign}(x)\cdot x = |x| \text{ and } |x|^{\frac{1}{p-1}}|x| = |x|^{\frac{p}{p-1}})\\
&= \frac{\delta}{\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}^{\frac{1}{p-1}}} \|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}^{\frac{p}{p-1}} && \text{(Definition of the } L_{\frac{p}{p-1}} \text{ norm)}\\
&= \delta\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}. && \text{(Simplifying exponents)}
\end{aligned}$$

Substituting this back yields $\boldsymbol{v}^\star(s_i) = \gamma\delta\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}}$, completing the proof. □

To simplify the expression for the transient state values, we must evaluate the norm of the absorbing state values. The following lemma calculates this value.

Lemma 25. *For any transient state s_i, $L_{\frac{p}{p-1}}$ norm of its absorbing state values is $\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}} = b_i^{\frac{p-1}{p}}$.*

Proof. We evaluate the $L_{\frac{p}{p-1}}$ norm of the value vector $\boldsymbol{v}^{(i)}$ directly:

$$\begin{aligned}
\|\boldsymbol{v}^{(i)}\|_{\frac{p}{p-1}} &= \left(\sum_{j=1}^{2M^*} |\boldsymbol{v}_j^{(i)}|^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}} && \text{(Definition of the } L_{\frac{p}{p-1}} \text{ norm)} \\
&= \left(2\sum_{k=1}^{M^*} \left(u_{i,k}^{p-1}\right)^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}} && \text{(Grouping symmetric pairs } \perp_{i,k}^{+} \text{ and } \perp_{i,k}^{-}) \\
&= \left(2\sum_{k=1}^{M^*} u_{i,k}^{p}\right)^{\frac{p-1}{p}} && \text{(Simplifying the exponent: } (p-1)\cdot\frac{p}{p-1} = p) \\
&= (2x_i)^{\frac{p-1}{p}} && \text{(Substituting the construction } x_i = \sum_{k=1}^{M^*} u_{i,k}^{p}) \\
&= b_i^{\frac{p-1}{p}}. && \text{(Substituting the definition } b_i = 2x_i)
\end{aligned}$$

This completes the proof. □

Proof of Theorem 3-Item 1. By Lemma 22, and because $\gcd(p-1, p) = 1$, the $\frac{p}{p-1}$-ROOT-SUM problem is as hard as the p-ROOT-SUM problem. Therefore, establishing that the problem DiscRMC is $\frac{p}{p-1}$-ROOT-SUM-hard proves that it is also p-ROOT-SUM-hard.

We evaluate the Bellman equation at the initial state s_0:

$$\begin{aligned}
\boldsymbol{v}^\star(s_0) &= c(s_0) + \gamma\sum_{i=1}^{n} \hat{\mathfrak{p}}_{s_0}(s_i)\boldsymbol{v}^\star(s_i) && \text{(Because } \delta_{s_0} = 0) \\
&= 0 + \gamma\sum_{i=1}^{n} \frac{1}{n}\boldsymbol{v}^\star(s_i) && \text{(Substituting } c(s_0) = 0 \text{ and } \hat{\mathfrak{p}}_{s_0}(s_i) = \frac{1}{n}) \\
&= \frac{\gamma}{n}\sum_{i=1}^{n}\left(\gamma\delta b_i^{\frac{p-1}{p}}\right) && \text{(Substituting results from Lemmas 24 and 25)} \\
&= \frac{\gamma^2\delta}{n}\sum_{i=1}^{n} b_i^{\frac{p-1}{p}}.
\end{aligned}$$

We want to decide if $\boldsymbol{v}^\star(s_0) \geqslant \lambda$. Substituting our target threshold $\lambda = \frac{\gamma^2\delta K}{n}$ gives:

$$\begin{aligned}
\boldsymbol{v}^\star(s_0) \geqslant \lambda &\iff \frac{\gamma^2\delta}{n}\sum_{i=1}^{n} b_i^{\frac{p-1}{p}} \geqslant \frac{\gamma^2\delta K}{n} \\
&\iff \sum_{i=1}^{n} b_i^{\frac{p-1}{p}} \geqslant K. && \text{(Dividing both sides by the positive constant } \frac{\gamma^2\delta}{n})
\end{aligned}$$

Recall our scaling definitions $b_i := 2^p a_i$ and $K := 2^{p-1}\alpha$. We substitute these into the inequality:

$$\begin{aligned}
\sum_{i=1}^{n} (2^p a_i)^{\frac{p-1}{p}} \geqslant 2^{p-1}\alpha &\iff \sum_{i=1}^{n} 2^{p-1} a_i^{\frac{p-1}{p}} \geqslant 2^{p-1}\alpha && ((2^p)^{\frac{p-1}{p}} = 2^{p-1}) \\
&\iff 2^{p-1} \sum_{i=1}^{n} a_i^{\frac{p-1}{p}} \geqslant 2^{p-1}\alpha && \text{(Factoring out the constant } 2^{p-1}) \\
&\iff \sum_{i=1}^{n} a_i^{\frac{p-1}{p}} \geqslant \alpha. && \text{(Dividing by } 2^{p-1})
\end{aligned}$$

Because all numbers are bounded polynomially and Lemma 20 guarantees the gadget size M^* is bounded, the reduction is valid and runs in polynomial time. □

C.3 ETR Upper Bound

In this section, we establish that the decision version of the DiscRMC problem with L_p uncertainty sets belongs to the complexity class $\exists\mathbb{R}$ for any integer $p > 1$. Furthermore, we present an approach to solve the main computational problem by calling an $\exists\mathbb{R}$ solver.

The main DiscRMC problem asks to find the value vector $\boldsymbol{v}^\star_{\mathcal{M}} \in \mathbb{R}^{|\mathcal{S}|}$ and a best adversary policy $\tau^\star \colon \mathcal{S} \to \Delta(\mathcal{S})$. The decision version of this problem asks, for a given state s and a threshold λ, whether $\boldsymbol{v}^\star(s) \geqslant \lambda$. We will focus on formalizing the main problem as a polynomial-size $\exists\mathbb{R}$ formula, which can be solved via an $\exists\mathbb{R}$ solver. Once we find the exact answer using this approach, we can simply solve the decision problem by adding the threshold constraint to the formula.

Karush-Kuhn-Tucker (KKT). Before analyzing the adversary's policy, we recall the Karush-Kuhn-Tucker (KKT) technique [BV04]. This technique provides an equation to find the optimal solution in convex problems. Suppose we want to minimize a function $f(x)$, subject to inequality rules $g_i(x) \leqslant 0$ and equality rules $h_j(x) = 0$.

To apply KKT, we introduce a penalty variable for each rule. We use μ_i for each inequality rule and ν_j for each equality rule. We combine all conditions into a master equation called the Lagrangian:

$$L(x, \mu, \nu) = f(x) + \sum_i \mu_i g_i(x) + \sum_j \nu_j h_j(x)$$

The KKT conditions state that the optimal solution $x^\star$ must satisfy four conditions:

1. The solution $x^\star$ must satisfy all the original rules ($g_i(x^\star) \leqslant 0$ and $h_j(x^\star) = 0$).
2. The penalty variables for the inequalities cannot be negative ($\mu_i \geqslant 0$).
3. If $g_i(x^\star) < 0$, its penalty is ignored ($\mu_i = 0$). In other words, $\mu_i g_i(x^\star) = 0$ for every rule.
4. The derivative of Lagrangian must be zero ($\nabla f(x^\star) + \sum_i \mu_i \nabla g_i(x^\star) + \sum_j \nu_j \nabla h_j(x^\star) = 0$).

Slater's Condition. The KKT conditions are necessary and sufficient for global optimality if strong duality holds. A standard constraint to establish strong duality is *Slater's condition* [BV04]. Slater's condition requires the existence of a feasible point x in the interior of the problem domain. Formally, x must strictly satisfy all non-linear inequality constraints ($g_i(x) < 0$), but only needs to weakly satisfy linear inequality constraints ($g_i(x) \leqslant 0$) and equality constraints ($h_j(x) = 0$). Satisfying this condition guarantees that the KKT equations are sufficient to find the optimal solution.

To prove that our problem can be formalized via an $\exists\mathbb{R}$ formula, we need to express all conditions using polynomials. We first show that the L_p uncertainty set constraints can be written as a system of polynomial inequalities.

Lemma 26. *For any state $s \in \mathcal{S}$, the condition $\tau_s \in \mathcal{P}(s)$ can be expressed as a system of polynomial constraints.*

Proof. By definition, the uncertainty set $\mathcal{P}(s)$ contains the transition probabilities such that $\sum_{t \in Succ(s)} |\tau_{s,t} - \hat{\mathfrak{p}}_{s,t}|^p \leqslant \delta_s^p$.

Because the absolute value function is not a polynomial, we introduce auxiliary variables $y_{s,t} \in \mathbb{R}$ to bound the absolute differences. For each $t \in \mathcal{S}$, we add the following pair of linear inequalities:

$$\tau_{s,t} - \hat{\mathfrak{p}}_{s,t} - y_{s,t} \leqslant 0,$$
$$\hat{\mathfrak{p}}_{s,t} - \tau_{s,t} - y_{s,t} \leqslant 0.$$

These inequalities force $y_{s,t} \geqslant |\tau_{s,t} - \hat{\mathfrak{p}}_{s,t}|$. Note that the non-negativity of the auxiliary variables is logically implied. We replace the L_p constraint with the following inequality:

$$\sum_{t \in Succ(s)} y_{s,t}^p - \delta_s^p \leqslant 0.$$

Since $p > 1$ is a constant integer, $y_{s,t}^p$ is a valid polynomial term. Thus, the condition is expressed by a system of polynomial inequalities. □

Next, we formulate the adversary's optimization problem. By applying the KKT conditions, we can describe the optimal worst-case transition distribution using a polynomial system.

Lemma 27. *For any fixed value vector $\boldsymbol{v}$, the optimal transition distribution τ_s chosen by the adversary can be defined by a system of polynomial equations and inequalities using KKT conditions.*

Proof. The adversary wants to choose the worst-case transition probabilities τ_s to maximize the expected cost $\sum_{t \in \mathcal{S}} \tau_{s,t} \boldsymbol{v}_t$, which is equivalent to minimizing $-\sum_{t \in \mathcal{S}} \tau_{s,t} \boldsymbol{v}_t$.

Using Lemma 26, we can write the feasibility condition $\tau_s \in \mathcal{P}(s)$ using auxiliary variables $y_{s,t}$. Thus, the adversary's goal can be written as the following optimization problem:

Minimize:	$-\sum_{t \in Succ(s)} \tau_{s,t} \boldsymbol{v}_t$	
Subject to:	$\left(\sum_{t \in Succ(s)} \tau_{s,t}\right) - 1 = 0$	(Probabilities must sum to 1)
	$-\tau_{s,t} \leqslant 0$	(Probabilities cannot be negative)
	$\tau_{s,t} - \hat{\mathfrak{p}}_{s,t} - y_{s,t} \leqslant 0$	(Upper bound for distance)
	$\hat{\mathfrak{p}}_{s,t} - \tau_{s,t} - y_{s,t} \leqslant 0$	(Lower bound for distance)
	$\left(\sum_{t \in Succ(s)} y_{s,t}^p\right) - \delta_s^p \leqslant 0$	(Total L_p distance limit)

This is a convex optimization problem because the goal function is linear and all constraints are convex. To prove that the KKT technique finds the true optimum, we verify that the problem has Slater's condition. We evaluate this in two cases based on δ_s:

Case 1 ($\delta_s > 0$): We must find a strictly feasible point where the inequality constraints do not touch the exact edge (< 0). We choose the default probabilities ($\tau_{s,t} = \hat{\mathfrak{p}}_{s,t}$) and set the auxiliary variables to zero ($y_{s,t} = 0$).

- All of the linear constraints are satisfied by the definition as $\tau = \hat{\mathfrak{p}}_{s,t}$ is a valid probability, and all lower bounds and upper bounds are zero.
- The non-linear distance limit evaluates to $(\sum_{t \in Succ(s)} 0^p) - \delta_s^p = -\delta_s^p$. Because $\delta_s > 0$, this inequality strictly satisfies.

Because we found a valid choice safely inside the boundaries, Slater's condition is met.

Case 2 ($\delta_s = 0$): In this case, the uncertainty set $\mathcal{P}(s)$ is a single point with the nominal distribution. Therefore, the optimal transition distribution is the nominal probabilities ($\tau_{s,t} = \hat{\mathfrak{p}}_{s,t}$). Because there is no optimization here, we bypass the KKT formulation in this case. Instead, we express this deterministic optimal policy using linear equality constraints ($\tau_{s,t} - \hat{\mathfrak{p}}_{s,t} = 0$). Since linear equations are valid polynomial equations, the optimal policy is defined by a polynomial system.

Because Slater's condition holds, the optimum is characterized by the KKT conditions. By assigning penalty multipliers to the constraints, we construct the Lagrangian L. The fourth KKT condition requires the partial derivatives of L with respect to $\tau_{s,t}$ and $y_{s,t}$ to be zero.

Because the objective and all but the last constraint are linear, their partial derivatives evaluate to simple constants. The only non-linear term in the Lagrangian is the distance limit, which contains $\sum_{t \in Succ(s)} y_{s,t}^p$. Its derivative with respect to $y_{s,t}$ is $p y_{s,t}^{p-1}$.

Since $p > 1$ is an integer, the exponent $(p-1)$ is a non-negative integer, meaning no fractional powers are introduced. Therefore, the KKT system consists of polynomial equations and inequalities. □

Finally, we combine the Bellman equations and the KKT conditions into a single system. The following lemma proves that this combined polynomial system identifies the optimal solution.

Lemma 28. *Let Σ be the system of polynomial equations and inequalities formed by combining the Bellman equations and the KKT conditions for all states:*

1. *For every state $s \in \mathcal{S}$, the Bellman equation holds: $\boldsymbol{v}_s = c_s + \gamma \sum_{t \in \mathcal{S}} \tau_{s,t} \boldsymbol{v}_t$.*
2. *For every state $s \in \mathcal{S}$, the KKT polynomial system derived in Lemma 27 holds for τ_s and $\boldsymbol{v}$.*

A tuple $(\boldsymbol{v}, \tau, y, \mu, \nu)$ is a valid solution to Σ if and only if $\boldsymbol{v}$ is the optimal value vector $\boldsymbol{v}^\star$ and τ is an optimal adversary policy $\tau^\star$.

Proof. First, we show that the optimal value vector $\boldsymbol{v}^\star$ and the optimal worst-case policy $\tau^\star$ produce a valid solution to Σ. By Lemma 9, the robust Bellman operator has a unique fixed point. Thus, the true optimal value vector $\boldsymbol{v}^\star$ exists. For this $\boldsymbol{v}^\star$, there exists an optimal adversary policy $\tau^\star$ that achieves the worst-case transition, satisfying $\sum_{t \in \mathcal{S}} \tau_{s,t}^\star \boldsymbol{v}_t^\star = \max_{p \in \mathcal{P}(s)} p^\top \boldsymbol{v}^\star$. By substituting this into the first part of Σ, we get $\boldsymbol{v}_s^\star = c_s + \gamma \sum_{t \in \mathcal{S}} \tau_{s,t}^\star \boldsymbol{v}_t^\star$, which satisfies the Bellman equations.

Moreover, because $\tau_s^\star$ maximizes the expected cost for $\boldsymbol{v}^\star$, it is the optimal solution to the inner minimization problem in Lemma 27. Because this inner problem is convex and satisfies Slater's condition, strong duality holds. This guarantees that the KKT conditions are necessary for optimality. Therefore, there exist real auxiliary variables y and KKT multipliers μ, ν that satisfy the KKT system for $\boldsymbol{v}^\star$ and $\tau_s^\star$. Thus, the tuple $(\boldsymbol{v}^\star, \tau^\star, y, \mu, \nu)$ forms a valid solution to Σ.

Second, suppose we have a solution $(\boldsymbol{v}', \tau', y', \mu', \nu')$ for Σ. Because τ_s' satisfies the KKT conditions for $\boldsymbol{v}_s'$, and Slater's condition holds, τ_s' is the true worst-case transition distribution for $\boldsymbol{v}'$. This means the inner product reaches the maximum: $\sum_{t \in \mathcal{S}} \tau_{s,t}' \boldsymbol{v}_t' = \max_{p \in \mathcal{P}(s)} p^\top \boldsymbol{v}'$. Substituting this into the Bellman constraint of Σ gives:

$$\boldsymbol{v}_s' = c_s + \gamma \max_{p \in \mathcal{P}(s)} p^\top \boldsymbol{v}'$$

This means $\boldsymbol{v}'$ is a fixed point of the robust Bellman operator. By Lemma 9, this operator has a unique fixed point. Therefore, $\boldsymbol{v}'$ must be the optimal value vector $\boldsymbol{v}^\star$. Consequently, τ' is an optimal adversary policy. □

We now have a complete polynomial system Σ that finds the exact value vector and a best policy for the environment. Since the decision problem asks whether $\boldsymbol{v}^\star(s) \geqslant \lambda$, we can answer it by adding this linear constraint to the system Σ.

Proof of Theorem 3-Item 2. By Lemma 28, the polynomial system Σ identifies the optimal value vector $\boldsymbol{v}^\star$ and the optimal policy $\tau^\star$ as its real solutions.

The system Σ is constructed entirely using real variables, polynomial equations, and polynomial inequalities, where all coefficients are rational. By definition, finding a solution to a system of polynomial equations and inequalities over the real numbers is a problem in the existential theory of the reals. Thus, the main DiscRMC problem can be solved via an $\exists\mathbb{R}$ solver. Moreover, by adding the constraint $\boldsymbol{v}^\star(s) \geqslant \lambda$ to Σ, we can evaluate the threshold condition. Therefore, the decision problem belongs to the complexity class $\exists\mathbb{R}$. □

D Additional Experimental Details

Benchmarks. We evaluate our algorithm on five benchmarks. For each benchmark, we scale the state-space size up to $n = |\mathcal{S}| = 256$. Reward is represented as negated cost.

- **Gridworld** (Adapted from [SB18]). We consider a $k \times k$ grid navigation task containing $n = k^2$ states. The agent starts at the top-left $(0, 0)$ and navigates toward an absorbing goal at the bottom-right $(k-1, k-1)$ to receive a continuous +1 reward. Actions consist of the four cardinal directions. The environment dynamics are stochastic: intended movements succeed with a probability of 0.8, while slipping perpendicularly occurs with a 0.1 probability in either direction. Movements into boundaries result in self-loops. An absorbing trap is placed at $x = \lfloor (k-1)/2 \rfloor, y = k-1-x$, which imposes a continuous -1 penalty. All other transitions carry a standard step penalty of -0.01.
- **Inventory Management.** (Adapted from [Put94]). The state space $s \in \{0, \dots, n-1\}$ denotes the current inventory level. Let $d_{\max} = \max(1, \lfloor (n-1)/2 \rfloor)$ represent the maximum expected demand. At each step, the agent selects an order quantity from $\mathcal{A} = \{0, \mathrm{round}(d_{\max}/2), d_{\max}\}$, where $\mathrm{round}(x)$ rounds to the nearest integer, with half-integers rounded to the nearest even integer. Customer demand is modeled as a discrete triangular distribution supported on $\{0, \dots, d_{\max}\}$ with a peak at $m = \lfloor d_{\max}/2 \rfloor$ and unnormalized weights $w(d) = m - |d - m| + 1$. The reward function evaluates the expected profit under this distribution, assuming a unit revenue of 1, a unit holding cost of 0.1, and a unit ordering cost of 0.5.
- **Machine Replacement** (adapted by [WKR13]). The state space $s \in \{0, \dots, n-1\}$ tracks the machine's degradation level, starting pristine at $s = 0$ and fully broken at the absorbing state $s = n-1$. The action space consists of 3 options: `operate`, `repair`, and `replace`. The `operate` action causes the machine to degrade to $s + 1$ with a 1/3 probability or remain in s with a 2/3 probability. The `repair` action improves the condition to $s - 1$ with a 3/4 probability, failing and remaining in s with a 1/4 probability. The `replace` action deterministically resets the machine to $s = 0$. Operating provides a state-dependent reward defined as $\frac{n-1-s}{n-1}$, whereas the `repair` and `replace` actions impose fixed penalties of -0.25 and -0.5.
- **GARNET** (adapted from [AMT95]). Generalized Artificial Random Networks serve as unstructured random MDP baselines. We consider a state space of size n, where each state has 4 actions and a fixed transition branching factor of $b = 3$. The successor set $\mathit{Succ}(s, a) \subseteq \mathcal{S}$ of size b is drawn uniformly at random for every state-action pair. We assign normalized integer weights $w(s, a, s') \sim \mathcal{U}\{1, 1000\}$ to all $s' \in \mathit{Succ}(s, a)$. The nominal transition probabilities are calculated as $P(s' \mid s, a) = \frac{w(s,a,s')}{\sum_{s''} w(s,a,s'')}$. Immediate rewards are given by uniformly sampled integers $r(s, a) \sim \mathcal{U}\{0, 10\}$.
- **Long Chain**. A construction introduced in this paper to show a $\Theta(n)$ lower bound on policy iteration under a fixed discount factor γ. With chain length k (so $n = 2k + 1$), the state space consists of k *path* states $s_0, \dots, s_{k-1}$, k absorbing *leaf* states $l_0, \dots, l_{k-1}$, and an absorbing sink s_{end}. From each path state s_i, action `path` deterministically advances to s_{i+1} (or to s_{end} when $i = k - 1$), while action `leaf` jumps to l_i. Per-step rewards are 0 on path states, 1 on leaves, and $\gamma^{-(k+1)}$ on the sink; the inflated sink reward is calibrated so that `path` is optimal at every state by a unit margin. Initialized with `leaf` everywhere, policy iteration flips one path state per outer step, requiring $k = \Theta(n)$ iterations to converge. Although the sink reward grows in k, its bit-size remains $\mathcal{O}(k \log(1/\gamma))$, which is polynomial in the input size for any constant γ.

Results Figures 2 to 4 report the outer `RMDP-PI` and total inner `RMC-PI` iteration counts for different values of γ, δ under both L_1 and L_∞ uncertainty sets.

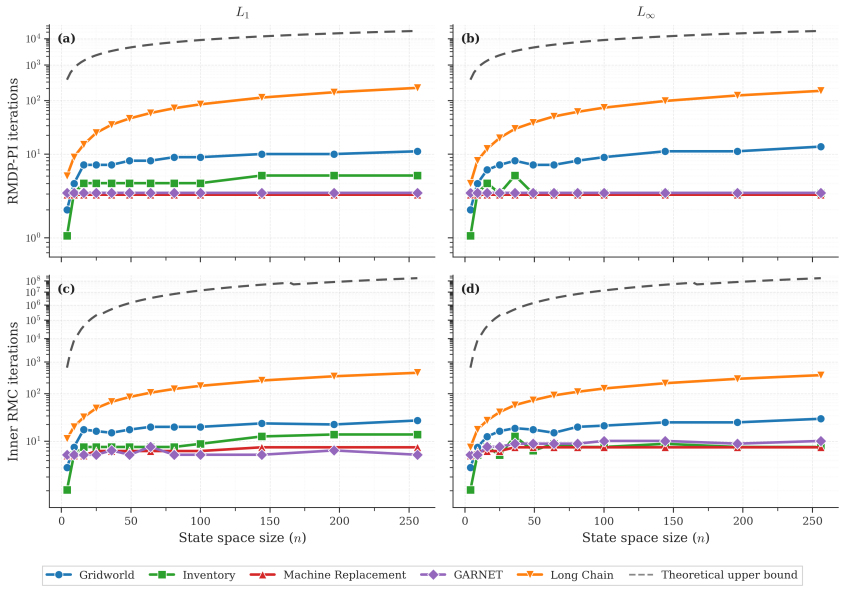


Figure 2: Outer `RMDP-PI` (a, b) and total inner `RMC-PI` (c, d) iterations versus state-space size n on five benchmarks, under L_1 (left) and L_∞ (right) uncertainty sets, with $\gamma = 0.9$, $\delta = 0.05$. Dashed line: theoretical upper bound.

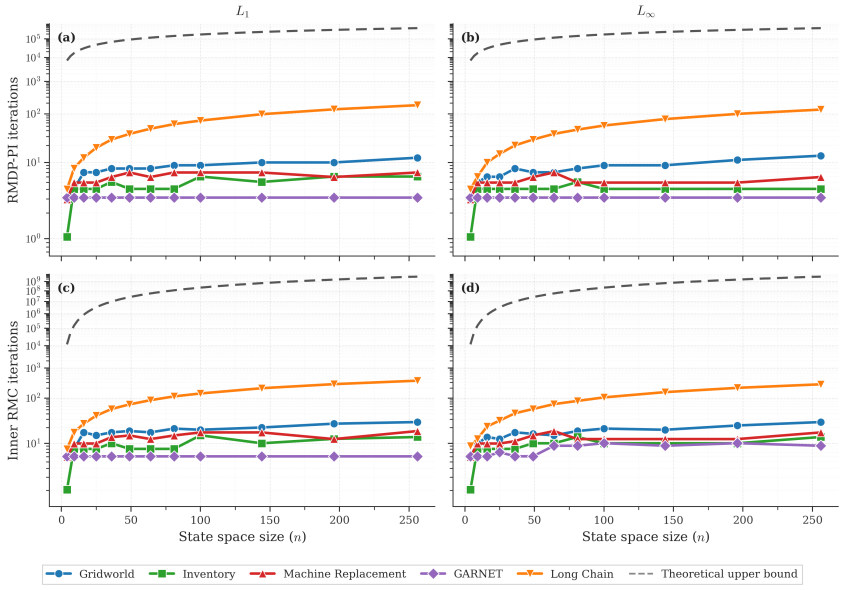


Figure 3: Outer `RMDP-PI` (a, b) and total inner `RMC-PI` (c, d) iterations versus state-space size n on five benchmarks, under L_1 (left) and L_∞ (right) uncertainty sets, with $\gamma = 0.99$, $\delta = 0.01$. Dashed line: theoretical upper bound.

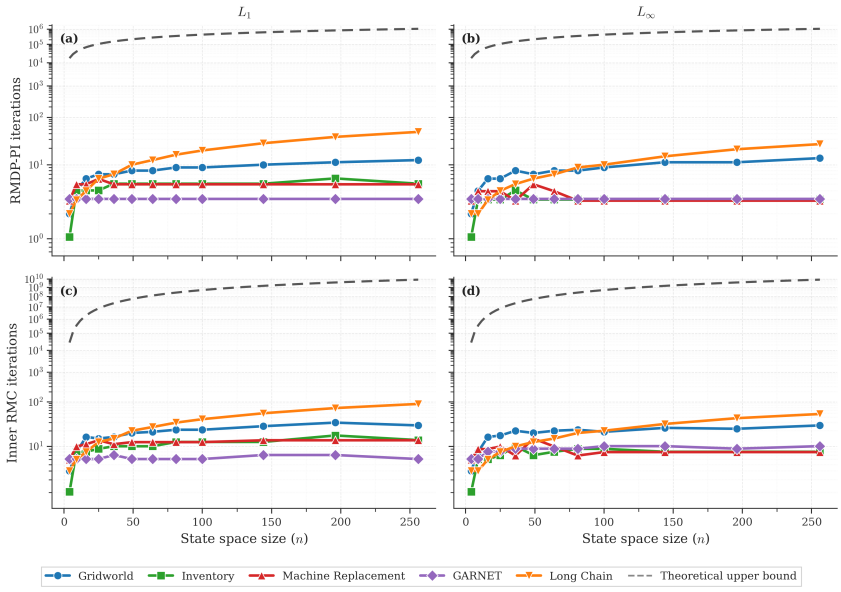


Figure 4: Outer `RMDP-PI` (a, b) and total inner `RMC-PI` (c, d) iterations versus state-space size n on five benchmarks, under L_1 (left) and L_∞ (right) uncertainty sets, with $\gamma = 0.995$, $\delta = 0.05$. Dashed line: theoretical upper bound.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize our main results, such as the complexity of RMDPs and RMCs, which match the proofs and experiments in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explain that we still need to find tighter bounds for L_p uncertainty sets and lower bounds for L_1 and L_∞ RMDPs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We state our assumptions, and we provide complete proofs for our theorems in the main sections and appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide enough details about our benchmarks and methods in the main text and the appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We describe the computational steps, libraries, and methods used in the readme file of the supplementary materials.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We list the parameters, such as state-space sizes and discount factors, in the experimental results Section and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Most of the benchmarks do not involve randomization in their construction. The policy iteration algorithm tested is deterministic and uses exact math to find the answers. Since the algorithm does not use random sampling to solve the problem, there is no statistical variance, so error bars are not needed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We state the specific hardware used for our tests, which was a machine with an Intel Core Ultra 5 processor and 16GB of RAM.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our research follows standard ethical guidelines for Neurips.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This is foundational theoretical research that does not have a direct societal impact or specific negative applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper presents theoretical mathematics and algorithms, which poses no safety risks or danger of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring

that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All benchmark sources are cited in the references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new software or datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: This study is purely theoretical and does not involve human subjects or crowdsourcing.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human participants were involved, so approval is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our methodology does not use LLMs as a core method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.