Fast Algorithms for L_∞-Constrained S-Rectangular Robust MDPs

Bahram Behzadian
University of New Hampshire
bahram@cs.unh.edu

Marek Petrik
University of New Hampshire
mpetrik@cs.unh.edu

Chin Pang Ho
City University of Hong Kong
clint.ho@cityu.edu.hk

Abstract

Robust Markov decision processes (RMDPs) are a useful building block of robust reinforcement learning algorithms but can be hard to solve. This paper proposes a fast, exact algorithm for computing the Bellman operator for S-rectangular robust Markov decision processes with L_∞-constrained rectangular ambiguity sets. The algorithm combines a novel homotopy continuation method with a bisection method to solve S-rectangular ambiguity in quasi-linear time in the number of states and actions. The algorithm improves on the cubic time required by leading general linear programming methods. Our experimental results confirm the practical viability of our method and show that it outperforms a leading commercial optimization package by several orders of magnitude.

1 Introduction

Markov decision processes (MDPs) are a powerful framework for dynamic decision-making problems and reinforcement learning (Bertsekas and Tsitsiklis, 1996; Puterman, 2005; Sutton and Barto, 2018). The MDP model assumes that the exact transition probabilities and rewards are available. However, these transition probabilities are typically unknown and must be estimated from sampled data. Such estimations are prone to errors, and the MDP's solution is sensitive to the introduced statistical errors. In particular, the quality of the optimal policy degrades significantly even with small errors in the transition probabilities (Le Tallec, 2007).

Robust MDPs (RMDPs) mitigate MDPs' sensitivity to estimation errors by computing an optimal policy for the worst plausible realization of the transition probabilities. This set of plausible transition probabilities is known as the *ambiguity set*. In this paper, we study RMDPs with S-rectangular ambiguity sets, which can be solved in polynomial time (Hansen, Miltersen, and Zwick, 2013). However, computing the worst-case realization of transition probabilities often requires solving a linear program (LP) or another convex optimization problem. Modern solvers are powerful and efficient, but as the problem size grows, solving an LP for every state becomes computationally prohibitive (Ho, Petrik, and Wiesemann, 2018).

Most prior work has focused on RMDPs with L_1-constrained ambiguity sets because both convenient concentration inequalities (Weissman et al., 2003; Petrik, Ghavamzadeh, and Chow, 2016; Russel, Gu, and Petrik, 2019) and fast algorithms (Iyengar, 2005; Petrik and Subramanian, 2014; Ho, Petrik, and Wiesemann, 2020) exist for this scenario. The concentration inequalities play an important role in the data-driven construction of high-confidence RMDPs. However, ambiguity sets defined by the L_∞-norm are more natural and interpretable by human modelers (Givan, Leach, and Dean,

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

2000; Delgado et al., 2016), and can significantly outperform L_1-based ambiguity sets in many circumstances (Behzadian et al., 2021). Unfortunately, RMDPs with S-rectangular ambiguity sets defined in terms of the L_∞ ball can currently be solved only using general-purpose LP solvers, which are complex and slow.

As our main contribution, we propose a new, fast algorithm for solving RMDPs with L_∞-constrained ambiguity sets. Our algorithm combines a new homotopy continuation method with a bisection method to achieve quasi-linear $\mathcal{O}(SA \log S)$ time complexity with respect to the number of states S and actions A. This computational complexity compares favorably with the cubic $\mathcal{O}((SA)^{3.5})$ time complexity of general interior-point LP algorithms. We identify new simplifying properties of the robust optimization problem defined over L_∞ balls to develop our algorithms.

Although bisection and homotopy methods have been used previously in the context of robust MDPs, their use and assumptions differ significantly from this work. A bisection method was used to solve SA-rectangular RMDPs (Nilim and El Ghaoui, 2005), but their approach does not generalize to S-rectangular RMDPs that we target. Homotopy and bisection methods have been used to solve L_1-constrained ambiguity sets (Ho, Petrik, and Wiesemann, 2018, 2020) but these methods are based on sparsity properties of the L_1-norm, which do not hold for the L_∞-norm. We elaborate on this important difference after we introduce our algorithm. The existing efficient algorithms are developed for the SA-rectangular RMDPs with L_∞ balls (Givan, Leach, and Dean, 2000), but they do not generalize to S-rectangular RMDPs. Developing fast optimization algorithms for S-rectangular RMDPs is especially challenging because optimal policies may need to be randomized.

Several new, fast methods have been proposed recently for solving RMDPs more efficiently. They propose replacing the standard value and policy iteration methods with more efficient algorithms, such as forms of modified policy iteration (Kaufman and Schaefer, 2013; Ho, Petrik, and Wiesemann, 2020) or gradient descent (Grand-Clément and Kroer, 2021). Most of these accelerated methods can further benefit from the fast Bellman operator algorithms that we propose in this work.

The remainder of the paper is organized as follows: Section 2 describes the basic robust MDP framework. Then, Section 3 proposes a new homotopy method for solving SA-rectangular ambiguity sets, which serves as a building block for our main contribution. In Section 4, we propose a bisection method that can solve, in combination with the homotopy method, RMDPs with S-rectangular ambiguity sets. Finally, Section 5 presents experimental results that show that our method is over 1000 times faster than using Gurobi, a leading commercial linear solver, when solving RMDPs with hundreds of states.

Notation: We reserve lower- and uppercase bold characters for vectors and matrices respectively. The symbol Δ^x denotes the probability simplex in $\mathbb{R}^x_+$. Finally, we use $\mathbf{I}$, $\mathbf{1}$, $\mathbf{0}$ to denote an identity matrix, a vector of ones, and a vector of zeros respectively.

2 Preliminaries: Robust MDPs

This section surveys the basic properties of RMDPs; for example, please see (Iyengar, 2005; Wiesemann, Kuhn, and Rustem, 2013; Ho, Petrik, and Wiesemann, 2020) for more details. We consider a finite RMDP model with states $\mathcal{S} = \{1, \ldots, S\}$ and actions $\mathcal{A} = \{1, \ldots, A\}$. The agent takes an action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, it receives a reward $r_{s,a} \in \mathbb{R}$, and it transitions to the next state $s' \in \mathcal{S}$ with a probability of $p_{s,a,s'}$. The transition probabilities $p_{s,a,s'}$ are unknown but are restricted to be in an ambiguity set $\mathcal{P} \subseteq (\Delta^S)^{S \times A}$. The initial state is distributed according to $\boldsymbol{p}_0 \in \Delta^S$.

We aim to compute a policy $\pi : \mathcal{S} \to \Delta^A$ from the set of stationary *randomized* policies Π that maximizes the expected γ-discounted return $\rho : \Pi \times \mathcal{P} \to \mathbb{R}$ for the worst-case transition probabilities:

$$\max_{\pi \in \Pi} \min_{\boldsymbol{p} \in \mathcal{P}} \ \rho(\pi, \boldsymbol{p}) \ . \tag{1}$$

Here, $\rho(\pi, \boldsymbol{p})$ is the standard discounted infinite-horizon return for a policy π defined as $\rho(\pi, \boldsymbol{p}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \cdot r(S_t, A_t) \mid A_t \sim \pi(S_t), S_{t+1} \sim \boldsymbol{p}_{S_t, A_t}, S_0 \sim \boldsymbol{p}_0\right]$. The optimization problem in (1) can be seen as a zero-sum game, where adversarial nature chooses transition probabilities from the ambiguity set in order to minimize the agent's return. Since solving the general optimization problem in (1) is NP-hard (e.g., (Wiesemann, Kuhn, and Rustem, 2013)), most research has focused on RMDPs with S-rectangular and SA-rectangular ambiguity sets that can be solved in polynomial time (Iyengar, 2005; Le Tallec, 2007; Wiesemann, Kuhn, and Rustem, 2013).

SA-rectangular ambiguity sets $\mathcal{P}$ are defined as Cartesian products of sets $\mathcal{P}_{s,a} \subseteq \Delta^S$ for each state s and action a as $\mathcal{P} = \big\{\boldsymbol{p} \in (\Delta^S)^{S\times A} \;\mid\; \boldsymbol{p}_{s,a} \in \mathcal{P}_{s,a},\, s \in \mathcal{S},\, a \in \mathcal{A}\big\}$. The intuitive interpretation of SA-rectangularity is that nature can choose the worst transition probabilities from sets $\mathcal{P}_{s,a}$ for each state s and action a *independently*. We focus on ambiguity sets bounded by L_∞-norm distance from nominal transition probabilities $\bar{\boldsymbol{p}}_{s,a} \in \Delta^S$ defined as

$$\mathcal{P}_{s,a} \;=\; \Big\{\boldsymbol{p}_{s,a} \in \Delta^S \;\mid\; \big\|\bar{\boldsymbol{p}}_{s,a} - \boldsymbol{p}_{s,a}\big\|_\infty \le \kappa_{s,a}\Big\}\;, \tag{2}$$

where $\kappa_{s,a} \ge 0$ is the robustness budget, and the nominal transition probability $\bar{\boldsymbol{p}}_{s,a}$ is typically estimated from samples of state transitions.

To streamline the definition of the robust Bellman operator, we follow the notation of Ho, Petrik, and Wiesemann (2018) and define a *nature response function* $q : \mathbb{R}_+ \times \mathbb{R}^S \to \mathbb{R}$ that represents the nature's response for a particular state s, action a, and budget ξ as

$$q_{s,a}(\xi, \boldsymbol{v}) \;=\; \min_{\boldsymbol{p}\in\Delta^S} \Big\{r_{s,a} + \gamma \cdot \boldsymbol{p}^\mathsf{T}\boldsymbol{v} \;\mid\; \big\|\bar{\boldsymbol{p}}_{s,a} - \boldsymbol{p}\big\|_\infty \le \xi\Big\}\;. \tag{3}$$

Then, the SA-rectangular robust Bellman operator $\mathfrak{T} : \mathbb{R}^S \to \mathbb{R}^S$ for a value function $\boldsymbol{v} \in \mathbb{R}^S$ is

$$(\mathfrak{T}\boldsymbol{v})_s \;=\; \max_{a\in\mathcal{A}} \min_{\xi\le\kappa_{s,a}} q_{s,a}(\xi, \boldsymbol{v})\;. \tag{4}$$

The optimal value function $\boldsymbol{v}^\star \in \mathbb{R}^S$ must satisfy the robust Bellman optimality equation $\boldsymbol{v}^\star = \mathfrak{T}\boldsymbol{v}^\star$ and can be computed using value iteration, policy iteration, or other methods (Iyengar, 2005; Ho, Petrik, and Wiesemann, 2020; Kaufman and Schaefer, 2013; Grand-Clément and Kroer, 2021).

S-rectangular ambiguity sets relax the assumptions of SA-rectangular sets and compute less conservative policies but with a higher computational complexity (Wiesemann, Kuhn, and Rustem, 2013). They are defined as Cartesian products of sets $\mathcal{P}_s \subseteq (\Delta^S)^A$ for each state s as $\mathcal{P} = \big\{\boldsymbol{p} \in (\Delta^S)^{S\times A} \;\mid\; (\boldsymbol{p}_{s,a})_{a\in\mathcal{A}} \in \mathcal{P}_s,\, \forall s \in \mathcal{S}\big\}$. As with SA-rectangular sets, we also consider marginal ambiguity sets $\mathcal{P}_s$ defined in terms of the L_∞-norm as

$$\mathcal{P}_s \;=\; \left\{(\boldsymbol{p}_{s,a})_{a\in\mathcal{A}} \in (\Delta^S)^A \;\mid\; \sum_{a\in\mathcal{A}} \big\|\bar{\boldsymbol{p}}_{s,a} - \boldsymbol{p}_{s,a}\big\|_\infty \le \kappa_s\right\}\;,$$

where $\kappa_s \ge 0$ is the robustness budget and $\bar{\boldsymbol{p}}_{s,a}$ is the nominal transition probability. The important distinction from the SA-rectangular setting is that κ_s depends only on the state and not the action. The S-rectangular Bellman operator is then defined as

$$(\mathfrak{T}\boldsymbol{v})_s \;=\; \max_{\boldsymbol{d}\in\Delta^A} \min_{\xi\le\kappa_s} \sum_{a\in\mathcal{A}} d_a \cdot q_{s,a}(\xi, \boldsymbol{v})\;. \tag{5}$$

Notice that the S-rectangular Bellman operator allows for randomizing actions through the probability distribution $\boldsymbol{d}$, which improves robustness but introduces additional significant computational complexity (Wiesemann, Kuhn, and Rustem, 2013; Ho, Petrik, and Wiesemann, 2020).

The vast majority of RMDP methods employ value iteration and policy iteration principles and require computing the robust Bellman operator many times during their run (Iyengar, 2005; Wiesemann, Kuhn, and Rustem, 2013; Ho, Petrik, and Wiesemann, 2020). Therefore, it is important that it can be computed efficiently. In the remainder of the paper, we develop new quasi-linear time algorithms for computing the robust Bellman operator.

3 Computing the SA-Rectangular Bellman Operator in Linear Time

In this section, we develop a new quasi-linear time algorithm for computing the SA-rectangular robust Bellman operator defined by the L_∞-norm. This entails solving the optimization in (4). The algorithm developed in this section also serves as the major building block of the S-rectangular algorithm described in Section 4. The remainder of the section is organized as follows: Section 3.1 first analyzes the LP formulation of the function q, and, then, Section 3.2 uses these properties to develop a new, fast homotopy continuation algorithm.

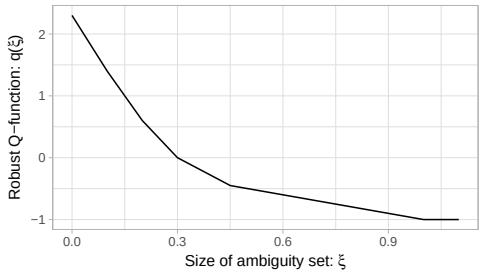

Figure 1: Function $q(\xi)$ in Example 3.1.

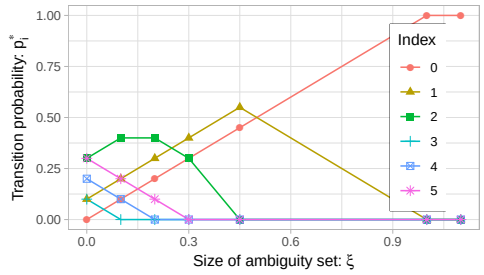

Figure 2: Probabilities $\boldsymbol{p}^\star(\xi)$ in Example 3.1.

Computing the SA-rectangular robust Bellman operator for a fixed state s, action a, and a value function $\boldsymbol{v}$ requires one to evaluate the nature response function $q_{s,a}(\xi, \boldsymbol{v})$ in (3). Because the symbols $s, a, \boldsymbol{v}$ are fixed throughout this section, we omit them in the notation. For example, we use $q(\xi)$ instead of $q_{s,a}(\xi, \boldsymbol{v})$ and $\bar{\boldsymbol{p}}$ in place of $\bar{\boldsymbol{p}}_{s,a}$. To further eliminate clutter, let $\boldsymbol{z} = r_{s,a} \cdot \mathbf{1} + \gamma \cdot \boldsymbol{v}$. Then, the optimization problem in (3) can be formulated as the following parametric LP:

$$\begin{aligned} q(\xi) &= \min_{\boldsymbol{p} \in \Delta^S} \left\{ \boldsymbol{p}^\mathsf{T} \boldsymbol{z} \;\middle|\; \|\bar{\boldsymbol{p}} - \boldsymbol{p}\|_\infty \leq \xi \right\} \\ &= \min_{\boldsymbol{p} \in \mathbb{R}^S} \left\{ \boldsymbol{z}^\mathsf{T} \boldsymbol{p} \;\middle|\; \mathbf{1}^\mathsf{T} \boldsymbol{p} = 1,\ -\xi \leq p_i - \bar{p}_i \leq \xi,\ p_i \geq 0,\ i = 1, \ldots, S \right\}. \end{aligned} \tag{6}$$

The remainder of this section develops fast algorithms for solving (6) for all values $\xi \geq 0$.

3.1 Properties of Nature Response Function q

The LP in (6) can be solved using generic solvers, like Gurobi or Mosek, but these are impractically slow for solving RMDPs. The optimization in (6) can also be solved in quasi-linear time for any *fixed* $\xi \geq 0$, as we summarize in Appendix C. The known quasi-linear algorithm is, unfortunately, insufficient for solving the S-rectangular robust Bellman operator in Section 4. In this section, we prove results that pave the way for solving (6) for *all* $\xi \geq 0$ simultaneously in quasi-linear time, which enables efficient algorithms for both S- and SA-rectangular RMDPs.

It will be convenient to use $\boldsymbol{p}^\star(\xi)$ to refer to an optimal solution in (6). To avoid unnecessary technicalities, we assume that all elements of $\boldsymbol{z}$ are distinct, which guarantees that the optimal solution $\boldsymbol{p}^\star(\xi)$ is unique. In practice, one may add an arbitrarily small value to the elements of $\boldsymbol{z}$ to ensure that they are all distinct.

To get some intuition into the form of the nature response function $q(\xi)$ and its optimal solution $\boldsymbol{p}^\star(\xi)$, consider the following simple example.

Example 3.1. *Consider an* RMDP *with six states, one action,* $\boldsymbol{z} = (-1, 0, 1, 2, 3, 4)^\mathsf{T}$*, and nominal transition probabilities* $\bar{\boldsymbol{p}} = (0.0, 0.1, 0.3, 0.1, 0.2, 0.3)^\mathsf{T}$*. The functions* $q(\xi)$ *and* $\boldsymbol{p}^\star(\xi)$ *are depicted in Figures 1 and 2, where Figure 2 shows the evolution of each* $p_i(\xi)$ *using a different color for each* i*.*

The following property of the function q is indispensable for our analysis and shows that $q(\xi)$ is always of the form depicted in Figure 1. It follows from standard LP properties and is proved in Appendix A.1.

Lemma 3.2. *The function* $q(\xi)$ *is continuous, piecewise linear, non-increasing, and convex in* ξ*.*

To develop an efficient algorithm, we now analyze the structure of the *bases* of the LP (6). Recall that a *basis* is a subset of S *linearly independent* constraints in the LP, which must hold with equality. There are S constraints included in each basis because S is the number of optimization variables. Note that constraints may be active (or violated) without being included in the basis.

To represent a basis in (6), we use sets $\mathcal{R}_B, \mathcal{D}_B, \mathcal{N}_B, \mathcal{T}_B \subseteq \{1, \ldots, S\}$ to indicate which constraints are included in the basis with their meanings summarized in Table 1. If $i \in \mathcal{D}_B$, we call it a *donor*, if $i \in \mathcal{R}_B$, we call it a *receiver*, and if $i \in \mathcal{N}_B$, we call it a *none*. The set $\mathcal{T}_B = \{1, \ldots, S\} \setminus \mathcal{R}_B \setminus \mathcal{D}_B \setminus \mathcal{N}_B$ represents the remaining indexes, and $i \in \mathcal{T}_B$ is called a *trader*. Lemma 3.4 below justifies the names for these sets.

Index $i \in$	Constraints in B
$\mathcal{R}_B$ (receiver)	$p_i - \bar{p}_i \leq \xi$ in B
$\mathcal{D}_B$ (donor)	$\bar{p}_i - p_i \leq \xi$ in B
$\mathcal{N}_B$ (none)	$p_i \geq 0$ in B

Figure 3: Composition of B for $i \in \mathcal{S}$.

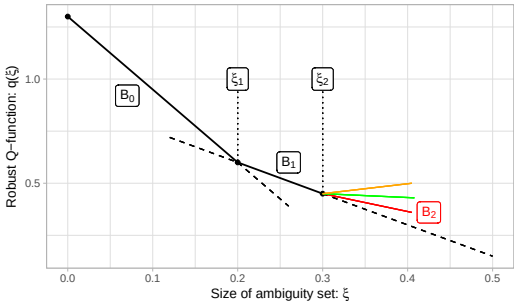

Figure 4: An illustration of Algorithm 1.

Our homotopy algorithm will leverage the specific behavior of the optimal solution $\boldsymbol{p}^\star(\xi)$ as a function of ξ. Because each basis B represents a set of S linearly independent inequalities with S variables, there exists a unique solution $\boldsymbol{p}_B(\xi)$ for any value ξ. Note that $\boldsymbol{p}_B(\xi)$ need not be optimal or feasible.

The following lemma establishes the properties of the bases in (6) that we need to consider in our optimization; its proof can be found in Appendix A.1.

Lemma 3.3. *Suppose that $\boldsymbol{p}^\star$ is optimal in* (6) *for some $\xi \geq 0$. Then, there exists a basis B such that* (i) $\boldsymbol{p}^\star = \boldsymbol{p}_B(\xi)$, (ii) *sets $\mathcal{R}_B, \mathcal{D}_B, \mathcal{N}_B, \mathcal{T}_B$ do not intersect,* (iii) $|\mathcal{R}_B| + |\mathcal{D}_B| + |\mathcal{N}_B| + |\mathcal{T}_B| = S$, (iv) $|\mathcal{T}_B| = 1$, *and* (v) $z_i < z_j < z_k$ *for each $i \in \mathcal{R}_B, j \in \mathcal{T}_B, k \in \mathcal{D}_B \cup \mathcal{N}_B$.*

Lemma 3.3 is important because it limits the bases relevant to the optimization, which is crucial for building fast algorithms. In particular, it shows that the sets $\mathcal{R}, \mathcal{D}, \mathcal{N}, \mathcal{T}$ *partition* the set $\mathcal{S}$, and there is always exactly one trader. The lemma also shows that z coefficients for receivers are smaller than the coefficient for the trader, which is smaller than the coefficients for donors and nones.

The following lemma establishes the rate of change of the linear function $\boldsymbol{p}_B(\xi)$, which is the last necessary component for our homotopy algorithm. The lemma's proof is in Appendix A.1.

Lemma 3.4. *The derivatives $\dot{\boldsymbol{p}} = \nabla_\xi \boldsymbol{p}_B(\xi)$ for any basis B that satisfies the properties in Lemma 3.3 are equal to*

$$\dot{p}_i = 1 \text{ if } i \in \mathcal{R}_B, \quad \dot{p}_i = -1 \text{ if } i \in \mathcal{D}_B, \quad \dot{p}_i = 0 \text{ if } i \in \mathcal{N}_B, \quad \dot{p}_i = |\mathcal{D}_B| - |\mathcal{R}_B| \text{ if } i \in \mathcal{T}_B \,.$$

for $i \in \mathcal{S}$. Moreover, the slope is $\dot{q} = {}^{d}/_{d\xi}\, q_B(\xi) = \sum_{i \in \mathcal{R}_B} z_i - \sum_{j \in \mathcal{D}_B} z_j + \sum_{\tau \in \mathcal{T}_B} \dot{p}_\tau z_\tau$.

Note that Lemma 3.3 shows that each $i \in \mathcal{S}$ is either a receiver, a donor, a trader, or none. Lemma 3.4 then shows that with an increasing ξ, a *donor* donates its probability mass, a *receiver* receives probability mass, a *trader* either donates or receives at a variable rate, and a *none* remains unchanged.

3.2 Homotopy Algorithm

We are now ready to describe the proposed homotopy method and prove its correctness and complexity. Algorithm 1 summarizes a conceptual version of the homotopy algorithm. As we discuss below, one needs to avoid computing the full gradient $\nabla_\xi \boldsymbol{p}_B(\xi)$ to achieve quasi-linear time complexity. The complete algorithm with quasi-linear runtime is described in Algorithm 3 in Appendix A.2.

The main idea of Algorithm 1 is simple: it iteratively computes the linear segments of $q(\xi)$ for all $\xi \geq 0$. The algorithm starts with $\xi = 0$, where the optimal solution is $\boldsymbol{p}_0 = \bar{\boldsymbol{p}}$ with objective value $q_0 = \boldsymbol{p}_0^\top \boldsymbol{z}$. Then, the algorithm tracks the optimal bases in $q(\xi)$ as ξ increases. When the $\boldsymbol{p}_{B_t}(\xi)$ becomes infeasible with the increasing ξ, the algorithm finds a new optimal basis B_{t+1} and continues until it arrives at a basis with ${}^{d}/_{d\xi}\, q(\xi') = 0$; the function q is constant for all $\xi \geq \xi'$. Since $q(\xi)$ is piecewise linear in ξ (see Lemma 3.2), we obtain its full description from all optimal bases.

The following theorem proves the correctness of Algorithm 1. Informally, the theorem shows that the function q is piecewise linear with *breakpoints* (points of non-linearity) only at $\xi_t, t = 1, \ldots, T+1$. Note that $\xi_{T+1} = 1$ because this is the upper bound on the L_∞-norm of a difference of two discrete probability distributions, and, as a result, the function $q(\xi)$ is constant for $\xi > 1$. The proof can be found in Appendix A.1.

Algorithm 1 Homotopy method to compute $q(\xi)$

1: **input:** Objective $\boldsymbol{z}$, and nominal probabilities $\bar{\boldsymbol{p}}$;
2: **output:** Breakpoints $(\xi_t)_{t=0,\dots,T+1}$ and $(q_t)_{t=0,\dots,T+1}$ such that $q_t = q(\xi_t)$;
3: Initialize $\xi_0 \leftarrow 0, t \leftarrow 0, \boldsymbol{p}_0 \leftarrow \bar{\boldsymbol{p}}$ and $q_0 \leftarrow q(\xi_0) = \boldsymbol{p}_0^\mathsf{T}\boldsymbol{z}$, $\tau_0 = \lceil S/2 \rceil$ and basis B_0 such that:
4: $\mathcal{T}_{B_0} = \{\tau_0\}$, $\mathcal{R}_{B_0} = \{i \,|\, i < \tau_0\}$, $\mathcal{D}_{B_0} = \{j \,|\, j > \tau_0\}$, $\mathcal{N}_{B_0} = \{\}$;
5: **while** $\dot{q}_t < 0$ **do**
6: Compute maximum step size for B_t to remain feasible ($\mathcal{T}_{B_t} = \{\tau_t\}$):
7: $\Delta\xi_t \leftarrow \max\{\xi \geq 0 \;|\; \boldsymbol{p}_t + \xi \cdot \nabla_\xi \boldsymbol{p}_{B_t}(\xi_t) \geq \boldsymbol{0},\, |(\boldsymbol{p}_t + \xi \cdot \nabla_\xi \boldsymbol{p}_{B_t}(\xi_t) - \bar{\boldsymbol{p}})_{\tau_t}| \leq \xi_t + \xi\}$;
8: Update breakpoints:
9: $\boldsymbol{p}_{t+1} \leftarrow \boldsymbol{p}_t + \Delta\xi_t \cdot \nabla_\xi \boldsymbol{p}_{B_t}(\xi_t)$; $q_{t+1} \leftarrow \boldsymbol{p}_{t+1}^\mathsf{T}\boldsymbol{z}$; $\xi_{t+1} \leftarrow \xi_t + \Delta\xi_t$;
10: Let $B_{t+1} \leftarrow$ next basis with the steepest slope (see Lemma 3.7 and Table 1);
11: Let $t \leftarrow t + 1$;
12: **end while**
13: Let $\xi_{T+1} \leftarrow 1$ and $q_{T+1} \leftarrow q_T$;
14: **return:** $(\xi_t)_{t=0,\dots,T+1}$, and $(q_t)_{t=0,\dots,T+1}$

Type	B_{t+1}	$\mathcal{D}_{B_{t+1}}$	$\mathcal{R}_{B_{t+1}}$	$\mathcal{T}_{B_{t+1}}$	$\mathcal{N}_{B_{t+1}}$
1: $\mathcal{D} \to \mathcal{N}$	$\hat{B}^1$	$\mathcal{D}_{B_t} \setminus \{l\}$	$\mathcal{R}_{B_t}$	$\mathcal{T}_{B_t}$	$\mathcal{N}_{B_t} \cup \{l\}$
2: $\mathcal{T} \to \mathcal{N}$	$\hat{B}^2$	$\mathcal{D}_{B_t}$	$\mathcal{R}_{B_t} \setminus \{m\}$	$\{m\}$	$\mathcal{N}_{B_t} \cup \mathcal{T}_{B_t}$
3: $\mathcal{T} \to \mathcal{D}$	$\hat{B}^3$	$\mathcal{D}_{B_t} \cup \mathcal{T}_{B_t}$	$\mathcal{R}_{B_t} \setminus \{m\}$	$\{m\}$	$\mathcal{N}_{B_t}$

Table 1: Possible types of basis change at a breakpoint ξ_{t+1} described in Lemma 3.7.

Theorem 3.5. *Suppose that Algorithm 1 returns $(\xi_t)_{t=0,\dots,T+1}$ and $(q_t)_{t=0,\dots,T+1}$. Then $q(\alpha \cdot \xi_t + (1-\alpha) \cdot \xi_{t+1}) = \alpha \cdot q(\xi_t) + (1-\alpha) \cdot q(\xi_{t+1})$ for $\alpha \in [0,1]$ and $t = 0, \dots, T+1$.*

We will refer to Figure 4 in order to provide the intuition that underlies the construction of Algorithm 1 and its correctness. The figure depicts an example state of Algorithm 1 at $t = 2$ and Line 10. The solid lines show the values q_{B_1} and q_{B_2} when they are feasible and optimal. The dashed lines indicate when the bases are infeasible or suboptimal at each one of the breakpoints ξ_1, ξ_2. The colored lines at ξ_2 indicate the slopes for the possible candidates for B_2. The algorithm chooses a basis with the minimal slope.

The correctness of Algorithm 1 follows from the following three lemmas. The first lemma shows that the algorithm chooses the initial basis with the minimum possible slope.

Lemma 3.6. *The basis B_0 constructed in Line 3 of Algorithm 1 is feasible at $\xi = 0$ and has a steeper slope than any other basis B that satisfies the conditions of Lemma 3.3:*

$$d/d\xi \, q_{B_0}(0) \;\leq\; d/d\xi \, q_B(0) \; .$$

The second lemma shows that the next basis will be selected according to one of the rules in Table 1.

Lemma 3.7. *Let a basis B_t be optimal for ξ_{t+1} in Algorithm 1, such that $\boldsymbol{p}^\star(\xi_{t+1}) = \boldsymbol{p}_{B_t}(\xi_{t+1})$ and $q(\xi_{t+1}) = q_{B_t}(\xi_{t+1})$. Assume that $\boldsymbol{p}_{B_t}(\xi)$ is infeasible for $\xi > \xi_{t+1}$. If $\mathcal{B}$ are all bases feasible for some $\xi > \xi_t$, then one with the steepest slope can be constructed as*

$$\operatorname*{argmin}_{B \in \mathcal{B}} \; \frac{d}{d\xi} q(\xi_{t+1}) \;\ni\; \begin{cases} \hat{B}^1 & \text{if} \quad (\boldsymbol{p}_{B_t}(\xi_{t+1}))_l = 0, \text{ for some } l \in \mathcal{D}_{B_t} \\ \hat{B}^2 & \text{if} \quad (\boldsymbol{p}_{B_t}(\xi_{t+1}))_\tau = 0, \text{ and } \mathcal{T}_{B_t} = \{\tau\} \\ \hat{B}^3 & \text{if} \quad (\bar{\boldsymbol{p}} - \boldsymbol{p}_{B_t}(\xi_{t+1}))_\tau = \xi_{t+1}, \text{ and } \mathcal{T}_{B_t} = \{\tau\} \end{cases} ,$$

where $\hat{B}^1, \hat{B}^2, \hat{B}^3$ are defined in Table 1 and $m \in \operatorname{argmax}_{i \in \mathcal{R}_{B_t}} z_i$.

Lemma 3.7 shows that there are three possible types of basis change; any other possible choice of the basis would contradict the continuity of $q(\xi)$ (Lemma 3.2). Recall also that Lemma 3.3 shows that there is always exactly one trader. The *first* type of basis change occurs when p_l for a donor $l \in \mathcal{D}$ reaches zero; the donor turns into a none in the new basis. The *second* type of basis change occurs

when the trader probability mass becomes zero; the trader then turns into a none, and the receiver with the largest z value becomes the new trader. The *third* type of basis change happens when the trader's gradient satisfies $d/d\xi\, p_\tau(\xi) < -1$ and its probability mass reaches to its lower bound for a given ξ, making the basis infeasible for greater values of ξ. The trader then becomes a donor, and, again, the receiver with the largest z value becomes the new trader.

Finally, the third lemma shows that the optimal basis B_t identified at ξ_t remains feasible until ξ_{t+1}. Note that the convexity of $q(\xi)$ implies that the feasible basis also remains optimal.

Lemma 3.8. *If B_t is feasible and optimal at ξ_t in Algorithm 1, then it is also optimal on the interval $[\xi_t, \xi_t + \Delta\xi_t]$ computed in Line 6 of Algorithm 1.*

We now turn to the computational complexity of Algorithm 3. As the following theorem shows, the number of iterations T in Algorithm 1 is at most $\mathcal{O}(S)$. Unfortunately, keeping track of $\boldsymbol{p}_t$ in each iteration of Algorithm 1 requires also $\mathcal{O}(S)$ time leading to the overall time complexity of $\mathcal{O}(S^2)$. To adapt Algorithm 1 to run in quasi-linear time, Algorithm 3, in Appendix A.2, generates the necessary values $\xi_t,\ q_t$ without tracking the complete $\boldsymbol{p}_t$ values. Its runtime is quasi-linear because it needs to sort the values of $\boldsymbol{z}$ to perform the optimization in Line 10 in constant time.

Theorem 3.9. *Algorithm 1 terminates in at most $\mathcal{O}(S)$ iterations and can be adapted to run in $\mathcal{O}(S \log S)$ time (see Algorithm 3 in Appendix A.2).*

We conclude by discussing the relationship with the homotopy method proposed for solving RMDPs with the L_1 ambiguity sets (Ho, Petrik, and Wiesemann, 2018). Although our algorithm is also a homotopy method, it is based on analysis that departs significantly from earlier work. The simplifying properties for the L_∞ ambiguity sets differ significantly from the L_1-norm. When the ambiguity sets are defined as L_1 balls, only two components of $\boldsymbol{p}$ change at the time. Figure 2 illustrates that when the ambiguity sets are L_∞ balls, all components of $\boldsymbol{p}$ may change with the increasing ξ. The fast algorithm for the L_∞-constrained RMDP relies on the more subtle structure of the optimal bases described in Lemma 3.3, which leads to a more complex algorithm.

4 Computing the S-Rectangular Bellman Operator in Linear Time

In this section, we propose a fast algorithm for computing the robust Bellman operator (5) for S-rectangular RMDPs. We assume a fixed state $s \in \mathcal{S}$ and omit the subscripts throughout the section. For instance, the nominal probabilities for state s and action a are denoted by $\bar{\boldsymbol{p}}_a \in \Delta^A$. We also assume a fixed value function $\boldsymbol{v} \in \mathbb{R}^S$ and let $\boldsymbol{z}_a = r_{s,a} \cdot \mathbf{1} + \gamma \cdot \boldsymbol{v}$ for $a \in \mathcal{A}$.

The fast algorithm for computing the S-rectangular robust Bellman operator builds on Algorithm 1. As Theorem 3.9 shows, the function q_a defined in (3) is piecewise linear with $\mathcal{O}(S)$ linear segments that can be computed efficiently by Algorithm 3. Since q_a is piecewise linear, it is easy to construct its inverse just by swapping ξ_t and q_t to get the following function:

$$q_a^{-1}(u) \;=\; \min_{\boldsymbol{p} \in \Delta^S} \left\{ \|\boldsymbol{p} - \bar{\boldsymbol{p}}_a\|_\infty \;\mid\; \boldsymbol{p}^\mathsf{T} \boldsymbol{z}_a \le u \right\}, \quad \forall a \in \mathcal{A}. \tag{7}$$

The function q_a^{-1} returns the budget that nature needs to achieve a response u. Using the function q_a^{-1}, we can reformulate the S-rectangular robust Bellman operator as:

$$(\mathfrak{T}\boldsymbol{v})_s = \max_{\boldsymbol{d} \in \Delta^A} \min_{\boldsymbol{\xi} \in \mathbb{R}_+^A} \left\{ \sum_{a \in \mathcal{A}} d_a \cdot q_a(\xi_a) \;\mid\; \sum_{a \in \mathcal{A}} \xi_a \le \kappa \right\} = \min_{u \in \mathbb{R}} \left\{ u \;\mid\; \sum_{a \in \mathcal{A}} q_a^{-1}(u) \le \kappa \right\}. \tag{8}$$

The correctness of this formulation follows by standard duality arguments and is proved in Lemma A.3 in Appendix A.3.

The optimization in (8) is remarkable because its objective is a one-dimensional function with one constraint. A natural algorithm to use with such an optimization problem is the bisection method outlined in Algorithm 2 (see Algorithm 4 in Appendix A.3 for a more detailed algorithm). Algorithm 2 keeps an interval $[u_{\min}, u_{\max}]$ such that the optimal $u^\star$ satisfies that $u^\star \in [u_{\min}, u_{\max}]$. In every time step, the algorithm bisects the interval $[u_{\min}, u_{\max}]$ in half and updates $u_{\min}, u_{\max}$ in order to preserve that $u^\star \in [u_{\min}, u_{\max}]$. One may think of $u_{\min}$ as the maximal known infeasible u in (8) and of $u_{\max}$ as the minimal known feasible u in (8).

Algorithm 2 Bisection method for solving (7).

input: Desired precision ϵ, functions $q_a^{-1}, \forall a \in \mathcal{A}$
output: $\hat{u}$ such that $|u^\star - \hat{u}| \leq \epsilon$, where $u^\star$ is optimal in Equation (7)
Initialize bounds $u_{\min} \leftarrow \min_{a \in \mathcal{A}, s \in \mathcal{S}} (\boldsymbol{z}_a)_s$; $u_{\max} \leftarrow \max_{a \in \mathcal{A}, s \in \mathcal{S}} (\boldsymbol{z}_a)_s$;
while $u_{\max} - u_{\min} > 2\,\epsilon$ **do**
    Let $u \leftarrow (u_{\min} + u_{\max})/2$;
    if $\sum_{a \in \mathcal{A}} q_a^{-1}(u) \leq \kappa$ **then** $u_{\max} \leftarrow u$ **else** $u_{\min} \leftarrow u$ **end if**
end while
return: $(u_{\min} + u_{\max})/2$

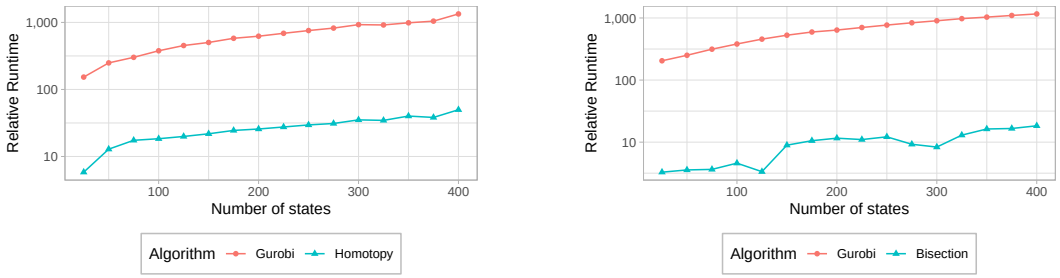

Figure 5: Relative computation time (unitless) of our algorithms and an LP solver over nominal MDP in SA-rectangular (left) and S-rectangular (right) inventory management RMDPs.

The time complexity of Algorithm 2 depends on the desired precision ϵ. To remove this dependence on ϵ, it is sufficient to replace the bisection by binary search over the breakpoints; we give the details of this method in Algorithm 4 in Appendix A.3. The following theorem, proved in Appendix A.3, summarizes the correctness and complexity of the proposed algorithms.

Theorem 4.1. *The combined Algorithms 1 and 2 compute the S-rectangular robust Bellman operator for any state $s \in \mathcal{S}$ and can be adapted (see Algorithms 3 and 4) to run in time $\mathcal{O}(SA\log(SA))$.*

5 Numerical Results

This section compares the empirical runtime of Algorithms 1 and 2 with the runtime of Gurobi 9.1, a leading LP solver. The results were generated on a computer with an Intel i7-9700 CPU with 32 GB RAM; the algorithms are implemented in C++.

As the main benchmark problem, we use the classic *Inventory Management (IM)* problem (Zipkin, 2000). In this problem, the decision-maker must decide at every time step how much inventory to order. The number of states and actions in this problem corresponds to the holding capacity and order size respectively. This makes it easy to scale the number of states and actions and evaluate how the algorithms scale with problem size. To evaluate the performance of our methods on small problems, we also consider the *RiverSwim (RS)* domain (Strehl and Littman, 2008) and the *Machine Replacement (MR)* domain (Delage and Ye, 2010). Please see Appendix B for the detailed description of these domains.

Figure 5 shows the time to compute the robust Bellman operator for a single state in the inventory management domain. The x-axis represents the number of states (maximum holding capacity) in the domain. The number of actions is the same as the number of states. The y-axis represents the time to compute the robust Bellman operator divided by time to compute the standard (non-robust) Bellman operator. The results show that even in MDPs with a few hundred states, the algorithms we propose are about 100 times faster than the leading LP solver. Interestingly, our algorithm is an order of magnitude faster even for small problems. We use the robustness budget $\kappa = 1.2$, but the computation time is insensitive to the particular choice of κ.

Table 2 compares the time to compute the robust policy for Machine Replacement (MR), RiverSwim (RS), and Inventory Management (IM) problems. The IM problem has 30 states. It is worth emphasizing that MR and RS are very small problems with less than 30 states, yet our algorithms

Rect.	Algorithm	MR	RS	IM
SA	Algorithm 1	$<$ **1**	**3**	**10**
SA	Gurobi LP	2960	2240	9770
S	Algorithm 2	**40**	**52**	**67**
S	Gurobi LP	129	217	2740

Table 2: Time (ms) to compute the robust policy for S- and SA-rectangular RMDPs with L_∞ sets.

Rect.	Algorithm	MR	RS	IM
SA	[Ho]-Alg.1	<1	2	1
SA	Gurobi LP	92	363	1140
S	[Ho]-Alg.2	1	2	5
S	Gurobi LP	79	317	2260

Table 3: Time (ms) to compute the robust policy for S- and SA-rectangular RMDPs with L_1 sets (Ho, Petrik, and Wiesemann, 2020).

are up to 800 times faster than using an LP solver. This indicates not only that our methods scale well with the number of states but also that the constant overhead is quite small. For the sake of completeness, we include in Table 3 the timing results obtained for the RMDP with L_1 ambiguity sets. These results show that solving the L_∞-constrained RMDP is more difficult than the L_1-constrained RMDP, but also that we can achieve similar dramatic speedups in L_∞-constrained RMDPs as (Ho, Petrik, and Wiesemann, 2020).

6 Conclusion

We introduced a new homotopy method for calculating robust Bellman operators for S- and SA-rectangular ambiguity sets constructed with L_∞-norm ball. Theoretically, we show that the worst-case time complexity of our algorithms is quasi-linear: $\mathcal{O}(SA\log(S))$. The algorithms also perform well in practice, outperforming a leading LP solver by several orders of magnitude.

In addition to being faster than a general-purpose LP solver, our algorithms are also much simpler. They make it possible to solve L_∞-constrained RMDPs without the cost and complexity of involving a general LP solver. Although free and open-source LP solvers are available, their performance falls significantly short of commercial ones. The algorithms we propose are also easy to combine with value function approximation methods in RMDPs (Tamar, Mannor, and Xu, 2014).

In terms of future work, we believe that it is important to understand whether similar algorithms can be developed for RMDPs with more complex ambiguity sets, such as ones defined using Wasserstein distance, L_2-norm, or KL-divergence.

Acknowledgments

The authors would like to thank Bence Cserna for discussions on this topic and the reviewer for the comments that improved this paper. Partial support for the work was provided by the National Science Foundation (Grants IIS-1717368 and IIS-1815275), the CityU Start-Up Grant (Project No. 9610481), the CityU Strategic Research Grant (Project No. 7005534), the National Natural Science Foundation of China (Project No. 72032005), and Chow Sang Sang Group Research Fund sponsored by Chow Sang Sang Holdings International Limited (Project No. 9229076). Any opinion, finding, conclusion, or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and the National Natural Science Foundation of China.

References

Behzadian, B.; Russel, R. H.; Petrik, M.; and Ho, C. P. 2021. Optimizing Percentile Criterion using Robust MDPs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1009–1017.

Bertsekas, D. P.; and Tsitsiklis, J. N. 1996. *Neuro-dynamic programming*. Athena Scientific, Belmont, MA.

Delage, E.; and Mannor, S. 2010. Percentile Optimization for Markov Decision Processes with Parameter Uncertainty. *Operations Research*, 58(1): 203–213.

Delage, E.; and Ye, Y. 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3): 595–612.

Delgado, K. V.; De Barros, L. N.; Dias, D. B.; and Sanner, S. 2016. Real-time dynamic programming for Markov decision processes with imprecise probabilities. *Artificial Intelligence*, 230: 192–223.

Faísca, N. P.; Dua, V.; and Pistikopoulos, E. N. 2007. *Multiparametric linear and quadratic programming*, 1–23. John Wiley & Sons, Ltd.

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

Grand-Clément, J.; and Kroer, C. 2021. First-Order Methods for Wasserstein Distributionally Robust MDP. *arXiv:2009.06790*.

Hansen, T. D.; Miltersen, P. B.; and Zwick, U. 2013. 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.

Ho, C. P.; Petrik, M.; and Wiesemann, W. 2018. Fast Bellman Updates for Robust MDPs. In *International Conference on Machine Learning (ICML)*, 1979–1988.

Ho, C. P.; Petrik, M.; and Wiesemann, W. 2020. Partial Policy Iteration for L1-Robust Markov Decision Processes. *arXiv:2006.09484*.

Ibaraki, T.; and Katoh, N. 1988. *Resource allocation problems: Algorithmic approaches*. The MIT Press.

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

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

Le Tallec, Y. 2007. *Robust, risk-sensitive, and data-driven control of Markov decision processes*. Ph.D. thesis, MIT.

Nilim, A.; and El Ghaoui, L. 2005. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research*, 53(5): 780–798.

Petrik, M.; Ghavamzadeh, M.; and Chow, Y. 2016. Safe policy improvement by minimizing robust baseline regret. *Advances in Neural Information Processing Systems*, 29: 2298–2306.

Petrik, M.; and Subramanian, D. 2014. RAAM : The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Neural Information Processing Systems (NIPS)*.

Puterman, M. L. 2005. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc.

Rockafellar, R. T. 1996. *Convex analysis*. Princeton University Press.

Russel, R. H.; Gu, T.; and Petrik, M. 2019. Robust Exploration with Tight Bayesian Plausibility Sets. *Multi-Disciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*.

Strehl, A. L.; and Littman, M. L. 2008. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8): 1309–1331.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT Press.

Tamar, A.; Mannor, S.; and Xu, H. 2014. Scaling Up Robust MDPs using Function Approximation. In *International Conference of Machine Learning (ICML)*.

Weissman, T.; Ordentlich, E.; Seroussi, G.; Verdú, S.; and Weinberger, M. J. 2003. Inequalities for the L1 Deviation of the Empirical Distribution.

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

Zipkin, P. H. 2000. *Foundations of inventory management*. McGraw-Hill/Irwin. ISBN 0256113793.

A Technical Results and Proofs

A.1 Proofs of Results in Section 3

Proof of Lemma 3.2. The functions $q(\xi)$ is convex due to the LP formulation of Equation (6); see (Faísca, Dua, and Pistikopoulos, 2007). □

Proof of Lemma 3.3. (i) The statement follows from the results in sections (ii)-(v) of this lemma.

(ii) If the intersection of any pair of $\mathcal{R}_B$, $\mathcal{D}_B$, and $\mathcal{N}_B$ is not an empty set, there exist a component i that satisfies two or more constraints in Table 1. In such a scenario, the basis B contains linearly dependent constraints that violate the definition of a basis. $\mathcal{T}_B = \{1, \ldots, S\} \setminus \mathcal{R}_B \setminus \mathcal{D}_B \setminus \mathcal{N}_B$ by definition does not intersect with other sets.

(iii) and (iv) By definition, B in $\mathcal{B}$ implies that the constraint $\mathbf{1}^\mathsf{T}\boldsymbol{p} = 1$ is in B; thus, one needs $S-1$ additional constraints selected from Figure 3 to form a basis. However, for every $i \in [S]$, at most one of the three constraints in Figure 3 should be selected, otherwise the constraints would not be linearly independent. Therefore, it implies that there exists exactly one $j \in [S]$ such that none of the three constraints in Figure 3 is selected in B, and so $j \in \mathcal{T}_B$. For every $i \in [S]\backslash\{j\}$, $i \in \mathcal{R}_B \cup \mathcal{D}_B \cup \mathcal{N}_B$.

(v) We prove this results via contradiction with the following cases. Firstly, suppose there exist a basis B', in which $l < \tau \in \mathcal{T}_{B'}$ where $l \in \mathcal{D}_{B'}$, then we construct another basis B, where $\mathcal{R}_B = \mathcal{R}_{B'} \cup \{l\}$, $\mathcal{D}_B = \mathcal{D}_{B'} \backslash \{l\}$, $\mathcal{N}_B = \mathcal{N}_{B'}$, and $\mathcal{T}_B = \mathcal{T}_{B'}$. By Lemma 3.4, we have:

$$\begin{aligned}
\dot{q}_{B'} &= \sum_{i \in \mathcal{R}_{B'}} z_i - \sum_{j \in \mathcal{D}_{B'}} z_j + (|\mathcal{D}_{B'}| - |\mathcal{R}_{B'}|)\, z_\tau, \\
\dot{q}_B &= \sum_{i \in \mathcal{R}_{B'}} z_i - \sum_{j \in \mathcal{D}_{B'}} z_j + 2z_l + (|\mathcal{D}_{B'}| - |\mathcal{R}_{B'}| - 2)\, z_\tau
\end{aligned}$$

and thus $\dot{q}_B - \dot{q}_{B'} = 2(z_l - z_\tau) \leq 0$ as $z_l \leq z_\tau$. The above construction of B also ensure that $p_B(\xi)$ is feasible in a neighborhood of ξ, as long as $p_{B'}(\xi)$ is feasible in a neighborhood of ξ.

Furthermore, suppose there exist a basis B', in which $l < \tau \in \mathcal{T}_{B'}$ where $l \in \mathcal{N}_{B'}$, then we construct another basis B, where $\mathcal{R}_B = \mathcal{R}_{B'} \cup \{l\}$, $\mathcal{D}_B = \mathcal{D}_{B'}$, $\mathcal{N}_B = \mathcal{N}_{B'} \backslash \{l\}$, and $\mathcal{T}_B = \mathcal{T}_{B'}$. By Lemma 3.4, we have:

$$\begin{aligned}
\dot{q}_{B'} &= \sum_{i \in \mathcal{R}_{B'}} z_i - \sum_{j \in \mathcal{D}_{B'}} z_j + (|\mathcal{D}_{B'}| - |\mathcal{R}_{B'}|)\, z_\tau, \\
\dot{q}_B &= \sum_{i \in \mathcal{R}_{B'}} z_i - \sum_{j \in \mathcal{D}_{B'}} z_j + z_l + (|\mathcal{D}_{B'}| - |\mathcal{R}_{B'}| - 1)\, z_\tau
\end{aligned}$$

and thus $\dot{q}_B - \dot{q}_{B'} = z_l - z_\tau \leq 0$ as $z_l \leq z_\tau$. The above construction of B also ensure that $p_B(\xi)$ is feasible in a neighborhood of ξ, as long as $p_{B'}(\xi)$ is feasible in a neighborhood of ξ.

Now we prove the second part of this result.

Suppose there exist a basis B', in which $m > \tau \in \mathcal{T}_{B'}$ where $m \in \mathcal{R}_{B'}$, then we construct another basis B, where $\mathcal{R}_B = \mathcal{R}_{B'} \backslash \{m\}$, $\mathcal{D}_B = \mathcal{D}_{B'} \cup \{m\}$, $\mathcal{N}_B = \mathcal{N}_{B'}$, and $\mathcal{T}_B = \mathcal{T}_{B'}$. By Lemma 3.4, we have:

$$\begin{aligned}
\dot{q}_{B'} &= \sum_{i \in \mathcal{R}_{B'}} z_i - \sum_{j \in \mathcal{D}_{B'}} z_j + (|\mathcal{D}_{B'}| - |\mathcal{R}_{B'}|)\, z_\tau, \\
\dot{q}_B &= \sum_{i \in \mathcal{R}_{B'}} z_i - \sum_{j \in \mathcal{D}_{B'}} z_j - 2z_m + (|\mathcal{D}_{B'}| - |\mathcal{R}_{B'}| + 2)\, z_\tau
\end{aligned}$$

and thus $\dot{q}_B - \dot{q}_{B'} = 2(z_\tau - z_m) \leq 0$ as $z_m \geq z_\tau$. The above construction of B also ensure that $p_B(\xi)$ is feasible in a neighborhood of ξ, as long as $p_{B'}(\xi)$ is feasible in a neighborhood of ξ.

□

Proof of Lemma 3.4. Note that if $k \in \mathcal{N}_B$ implies $(\boldsymbol{p}_B(\xi))_k = 0$ for every ξ therefore $\dot{p}_k = 0$. For all components $i \in \mathcal{R}_B$ we have $p_i - \bar{p}_i = \xi$. By taking the derivative with respect to ξ we have

$\dot{p}_i = 1$. Similarly, for all $j \in \mathcal{D}_B$ we have $\bar{p}_j - p_j = \xi$. Taking the derivative leads to $\dot{p}_j = -1$. We denote by $\boldsymbol{x}_{\mathcal{G}}$ the subvector of $\boldsymbol{x} \in \mathbb{R}^S$ formed by the elements x_i, $i \in \mathcal{G}$, where indices are contained in the set $\mathcal{G} \subseteq \mathcal{S}$. We consider a fixed basis B and thus drop the subscript B for the rest of this proof.

Figure 3 implies the following useful equality that any $\boldsymbol{p}$ must satisfy.

$$\begin{aligned} 1 = \mathbf{1}^\mathsf{T}\boldsymbol{p} &= \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{R}} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{D}} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{N}} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{T}} \\ &= \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{R}} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{D}} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{T}} \\ &= \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{R}} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{\mathcal{D}} + p_\tau \end{aligned}$$

where the second identity follows from the fact that $\forall k \in \mathcal{N}$ implies $p_k = 0$. By taking the derivative $\frac{d}{d\xi}$ from both sides we have:

$$\begin{aligned} 0 &= \mathbf{1}^\mathsf{T}\dot{\boldsymbol{p}}_{\mathcal{R}} + \mathbf{1}^\mathsf{T}\dot{\boldsymbol{p}}_{\mathcal{D}} + \dot{p}_\tau \\ &= |\mathcal{R}| - |\mathcal{D}| + \dot{p}_\tau . \end{aligned}$$

And finally we have:

$$\begin{aligned} \dot{q} &= \boldsymbol{z}^\mathsf{T}\dot{\boldsymbol{p}} \\ &= \boldsymbol{z}^\mathsf{T}\dot{\boldsymbol{p}}_{\mathcal{R}} + \boldsymbol{z}^\mathsf{T}\dot{\boldsymbol{p}}_{\mathcal{D}} + \boldsymbol{z}^\mathsf{T}\dot{\boldsymbol{p}}_{\mathcal{N}} + \boldsymbol{z}^\mathsf{T}\dot{\boldsymbol{p}}_{\mathcal{T}} \\ &= \sum_{i\in\mathcal{R}} z_i - \sum_{j\in\mathcal{D}} z_j + \dot{p}_\tau z_\tau \ . \end{aligned}$$

□

Proof of Theorem 3.5. The statement is true due to linearity of $q(\xi)$ on the interval $[\xi_t, \xi_{t+1}]$ shown in Lemma 3.2, as well as the results in Lemma 3.6, Lemma 3.7, and Lemma 3.8. □

Proof of Lemma 3.6. At $\xi = 0$, we can assume the none set is empty $\mathcal{N}_B = \emptyset$ because one can replace all non-negativity constraints $p_i \geq 0$ with $p_i - \bar{p}_i \leq \xi$ or $\bar{p}_i - p_i \leq \xi$. In Lemma 3.3, Section (v), we show for every $B \in \mathcal{B}$, $\forall\, i \in \mathcal{R}_B$, $\forall\, j \in \mathcal{D}_B$, and $\tau \in \mathcal{T}_B$ we have $i < \tau < j$. So $\dot{q}_B$ can be written as:

$$\begin{aligned} \dot{q}_B &= \sum_{i\in\mathcal{R}_B} z_i - \sum_{j\in\mathcal{D}_B} z_j + (|\mathcal{D}_B| - |\mathcal{R}_B|)\, z_\tau \\ &= \sum_{i=1}^{\tau-1} z_i - \sum_{j=\tau+1}^{S} z_j + ((S-\tau) - (\tau-1))z_\tau \\ &= \sum_{i=1}^{\tau-1} z_i - \sum_{j=\tau+1}^{S} z_j + (S - 2\tau + 1)z_\tau \\ &= \sum_{k=1}^{S} \operatorname{sign}(k-\tau) z_k + (S - 2\tau + 1)z_\tau \end{aligned} \tag{9}$$

Equation (9) shows at $\xi = 0$, the trader's rate $\dot{p}_\tau = S - 2\tau + 1$. We can also show that at $\xi = 0$, for all component $i \in \{1, \ldots, S\}$ we have $-1 \leq \dot{p}_i \leq 1$ because the constraints $p_i - \bar{p}_i \leq \xi$ and $\bar{p}_i - p_i \leq \xi$ are both active in equality. Thus we have

$$\begin{aligned} \min_{B\in\mathcal{B}} \quad & \frac{d}{d\xi} q_B(\xi_o) = \boldsymbol{z}^\mathsf{T}\dot{\boldsymbol{p}} \\ & \text{s. t.} \quad \mathbf{1}^\mathsf{T}\dot{\boldsymbol{p}} = 0 \ , \\ & \qquad -\mathbf{1} \leq \dot{\boldsymbol{p}} \leq \mathbf{1} \ . \end{aligned} \tag{10}$$

Since we previously showed the trader's exchange rate follows from $\dot{p}_\tau = |\mathcal{D}_B| - |\mathcal{R}_B|$ we can conclude $\dot{p}_\tau$ is an integer. Given the constraints in (10) at $\xi = 0$, we conclude $\dot{p}_\tau \in \{-1, 0, 1\}$. The

index of the trader is obtained from one of the following scenarios:

$$S - 2\tau + 1 = 0 \implies \tau = \frac{S+1}{2}, \tag{11}$$

$$S - 2\tau + 1 = 1 \implies \tau = \frac{S}{2}, \tag{12}$$

$$S - 2\tau + 1 = -1 \implies \tau = \frac{S+2}{2}, \tag{13}$$

When S is an odd number, τ can be only $\frac{S+1}{2}$ because S is also an integer and τ cannot be fractional. And when S is an even number, τ can be either $\frac{S}{2}$ or $\frac{S+2}{2}$. Algorithm 3 returns the exact solution in both cases.

Given the index of trader for B_0, the index of all donors and receivers can be achieved form Lemma 3.2 section (v). We initialize the sets: $\mathcal{T}_{B_0} = \{\lceil S/2 \rceil\}$, $\mathcal{R}_{B_0} = \{i \mid i < \tau\}$, $\mathcal{D}_{B_0} = \{j \mid j > \tau\}$, $\mathcal{N}_{B_0} = \{\}$;

□

Proof of Lemma 3.7. Suppose $z_1 \leq z_2 \leq \cdots \leq z_S$. Consider a base B that is feasible in the neighborhood of $\xi_t > 0$, and satisfies $B = \text{argmin}_{B \in \mathcal{B}} \frac{d}{d\xi} q(\xi_t)$. In Lemma 3.4, we show $\forall\, i \in \mathcal{R}_B$ and $\forall\, j \in \mathcal{D}_B \cup \mathcal{N}_B$ and $\tau \in \mathcal{T}_B$ we have $i < \tau < j$, and $\dot{q}_B$ can be written as:

$$\frac{d}{d\xi} q(\xi_t) = \dot{q}_B = \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k + (|\mathcal{D}_B| - |\mathcal{R}_B|)\, z_\tau \tag{14}$$

The adjacent basis $B' \in \mathcal{B}$ can be chosen from one of the following cases:

$$B' = \begin{cases} 1 & \mathcal{D}_{B'} = \mathcal{D}_B \backslash \{l\}, \quad \mathcal{N}_{B'} = \mathcal{N}_B \cup \{l\}, \quad \mathcal{T}_{B'} = \mathcal{T}_B, \quad \mathcal{R}_{B'} = \mathcal{R}_B \\ 2 & \mathcal{N}_{B'} = \mathcal{N}_B \cup \{\tau\}, \quad \mathcal{R}_{B'} = \mathcal{R}_B \backslash \{m\}, \quad \mathcal{T}_{B'} = \{m\}, \quad \mathcal{D}_{B'} = \mathcal{D}_B \\ 3 & \mathcal{D}_{B'} = \mathcal{D}_B \cup \{\tau\}, \quad \mathcal{R}_{B'} = \mathcal{R}_B \backslash \{n\} \quad \mathcal{T}_{B'} = \{n\}, \quad \mathcal{N}_{B'} = \mathcal{N}_B \\ 4 & \mathcal{R}_{B'} = \mathcal{R}_B \cup \{\tau\}, \quad \mathcal{D}_{B'} = \mathcal{D}_B \backslash \{o\}, \quad \mathcal{T}_{B'} = \{o\}, \quad \mathcal{N}_{B'} = \mathcal{N}_B \\ 5 & \mathcal{R}_{B'} = \mathcal{R}_B \cup \{\tau\}, \quad \mathcal{N}_{B'} = \mathcal{N}_B \backslash \{p\}, \quad \mathcal{T}_{B'} = \{p\}, \quad \mathcal{D}_{B'} = \mathcal{D}_B \\ 6 & \mathcal{N}_{B'} = \mathcal{N}_B \backslash \{q\}, \quad \mathcal{D}_{B'} = \mathcal{D}_B \cup \{q\}, \quad \mathcal{T}_{B'} = \mathcal{T}_B, \quad \mathcal{R}_{B'} = \mathcal{R}_B \end{cases} \tag{15}$$

Case 1 occurs when a donor becomes a none by donating all of its probability mass to a receiver. In this basis change, the index of the trader remains unchanged. B' is an adjacent basis for B since we only remove one active constraint ($\bar{p}_l - p_l \leq \xi$), and add another one ($p_l \geq 0$). In case 2, the trader becomes a none by losing all of its probability mass. The trader's index shifts from τ to m, one of the receivers in B. Note that in case 2 also, B' is an adjacent basis to B. We removed one active constraint ($p_m - \bar{p}_m \leq \xi$), and add another one ($p_\tau \geq 0$). Case 3 is similar to case 2, however in this case the trader reaches its lower bound, and as a result the new active constraint in B' is ($\bar{p}_\tau - p_\tau \leq \xi$). Case 4 occurs when a trader becomes a receiver. In this scenario, the trader's index shifts from τ to o, which was a member of $\mathcal{D}_B$. Case 5 and case 4 are similar. However, the trader in B' belongs to $\mathcal{N}_B$. In the last case, one of the components in $\mathcal{N}_B$ gain probability mass and moves to the donor's set. In the following, we show that cases 4-6 are not a feasible choice for B'.

Any other case violates Lemma 3.3, Section (v). The corresponding $\dot{q}_{B'}$ obtain as follows:

$$\dot{q}_{B'} = \begin{cases} 1 & \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k + z_l + (|\mathcal{D}_B| - |\mathcal{R}_B| - 1)\, z_\tau \\ 2 & \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k - z_m + (|\mathcal{D}_B| - |\mathcal{R}_B| + 1)\, z_m \\ 3 & \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k - z_\tau - z_n + (|\mathcal{D}_B| - |\mathcal{R}_B| + 2)\, z_n \\ 4 & \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k + z_\tau + z_o + (|\mathcal{D}_B| - |\mathcal{R}_B| - 2)\, z_o \\ 5 & \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k + z_\tau + (|\mathcal{D}_B| - |\mathcal{R}_B| - 1)\, z_p \\ 6 & \sum_{i \in \mathcal{R}_B} z_i - \sum_{k \in \mathcal{D}_B} z_k - z_q + (|\mathcal{D}_B| - |\mathcal{R}_B| + 1)\, z_\tau \end{cases} \tag{16}$$

And hence we have:

$$\dot{q}_{B'} = \begin{cases} 1 & \dot{q}_B + (z_l - z_\tau) \\ 2 & \dot{q}_B + (z_m - z_\tau)(|\mathcal{D}_B| - |\mathcal{R}_B|) \\ 3 & \dot{q}_B + (z_n - z_\tau)(|\mathcal{D}_B| - |\mathcal{R}_B| + 1) \\ 4 & \dot{q}_B + (z_o - z_\tau)(|\mathcal{D}_B| - |\mathcal{R}_B| - 1) \\ 5 & \dot{q}_B + (z_p - z_\tau)(|\mathcal{D}_B| - |\mathcal{R}_B| - 1) \\ 6 & \dot{q}_B - (z_q - z_\tau) \end{cases} \tag{17}$$

Given Lemmas A.1 and A.2, B'_4, B'_5, and B'_6 are not a suitable choice for B' since $\dot{q}_{B'_4} \leq \dot{q}_B$, $\dot{q}_{B'_5} \leq \dot{q}_B$ and $\dot{q}_{B'_6} \leq \dot{q}_B$.

The choice over B'_1, B'_2, and B'_3 depend on the probability mass of the components at each breakpoint.

In order to minimize the decent rate in the case of $B' = B'_2$, we can show that:

$$\dot{q}_{B'} = \min_{m \in \mathcal{R}_B} \dot{q}_B + (z_m - z_\tau)(|\mathcal{D}_B| - |\mathcal{R}_B|) \tag{18}$$

We know $z_m - z_\tau \leq 0$. And $0 \leq (z_m - z_\tau)(|\mathcal{D}_B| - |\mathcal{R}_B|)$ otherwise Lemma A.2 will be violated. As a result we conclude in this particular case $(|\mathcal{D}_B| - |\mathcal{R}_B|) \leq 0$.

In order to minimize Equation (18) the term $z_m - z_\tau$ should be minimized. Since $z_1 \leq \cdots \leq z_m \leq \cdots \leq z_\tau$, therefore $m^\star = \tau - 1$. With the same reasoning we can show in the case of $B' = B'_3$ we have $n^\star = \tau - 1$.

Our results follows the continuity assumption of the solution $\boldsymbol{p}^\star = \boldsymbol{p}_B(\xi)$ for all $\xi > 0$, in which a receiver can only become a trader, not a donor nor empty, at each breakpoints. Also, a donor cannot become a receiver unless it becomes a trader first. Otherwise, the continuity assumption will be violated.

□

Lemma A.1. *For all $B \in \mathcal{B}$ we have $|\mathcal{D}_B| - |\mathcal{R}_B| \leq 1$.*

Proof. Consider the problem with fixed ξ,

$$q(\xi) \;=\; \min_{\boldsymbol{p} \in \Delta^S} \left\{ \boldsymbol{p}^\mathsf{T} \boldsymbol{z} : \|\bar{\boldsymbol{p}} - \boldsymbol{p}\|_\infty \leq \xi \right\}, \tag{19}$$

For any fix $B \in \mathcal{B}$, we know:

$$\begin{aligned} &\text{if} \quad i \in \mathcal{R}_B \implies p_i = \bar{p}_i + \xi, \\ &\text{if} \quad j \in \mathcal{D}_B \implies p_j = \bar{p}_j - \xi, \\ &\text{if} \quad k \in \mathcal{N}_B \implies p_k = 0, \\ &\text{if} \quad \tau \in \mathcal{T}_B, \quad \exists \Delta \in \mathbb{R} \quad \text{that} \quad p_\tau = \bar{p}_\tau + \Delta. \end{aligned}$$

We also know

$$\begin{aligned} \mathbf{1}^\mathsf{T} \boldsymbol{p} = 1 \iff & \sum_{i \in \mathcal{R}_B} (\bar{p}_i + \xi) + \sum_{j \in \mathcal{D}_B} (\bar{p}_j - \xi) + \bar{p}_\tau + \Delta = 1 \\ \iff & (1 - \sum_{k \in \mathcal{N}_B} \bar{p}_k) + (|\mathcal{R}_B| - |\mathcal{D}_B|)\xi + \Delta = 1 \\ \iff & \Delta = \sum_{k \in \mathcal{N}_B} \bar{p}_k + (|\mathcal{D}_B| - |\mathcal{R}_B|)\xi \end{aligned}$$

We know for feasibility, $\Delta \leq \xi$ so we have:

$$\begin{aligned} &\sum_{k \in \mathcal{N}_B} \bar{p}_k + (|\mathcal{D}_B| - |\mathcal{R}_B|)\xi \leq \xi \\ &\sum_{k \in \mathcal{N}_B} \bar{p}_k \leq (|\mathcal{R}_B| - |\mathcal{D}_B| + 1)\xi \end{aligned}$$

Since $\sum_{k\in\mathcal{N}_B} \bar{p}_k \geq 0$, and $\xi > 0$, we conclude $(|\mathcal{R}_B| - |\mathcal{D}_B| + 1) \geq 0$. As a result:

$$|\mathcal{D}_B| - |\mathcal{R}_B| \leq 1 \ .$$

□

Lemma A.2. *let $(\xi_t)_{t=0,\ldots,T+1}$, and $q(\xi)$ is a piecewise-affine convex function with breakpoints ξ_l. Under the assumption of $\xi_t < \xi_{t+1}$ for all $t = 0, \ldots, T+1$, we have $\dot{q}_0 \leq \dot{q}_1 \leq \ldots \leq \dot{q}_{T+1}$.*

Proof. The results follows from Theorem 24.1 in Rockafellar (1996).

□

Proof of Lemma 3.8. The optimization problem in (3) can be formulated as the following parametric LP:

$$q(\xi) \ = \ \min_{\boldsymbol{p}\in\mathbb{R}^S} \ \left\{\boldsymbol{z}^\mathsf{T}\boldsymbol{p} \ \mid \ \mathbf{1}^\mathsf{T}\boldsymbol{p} = 1, \ -\xi \leq p_i - \bar{p}_i \leq \xi, \ p_i \geq 0, \ i = 1, \ldots, S\right\} \ . \tag{20}$$

At each basis B_t, there are S constraints that are active and satisfied in equity. In order to maintain the feasibility the basis B_t on the interval $[\xi_t, \xi_t + \Delta\xi_t]$, one needs to keep track of constrains that will be violated first by increasing $\xi \in [\xi_t, \xi_t + \Delta\xi_t]$, and relax all other constraint. Since the donation rate is equal among all donors $\dot{p}_i = -1 \ \forall i \in \mathcal{D}_{B_t}$, the non-negativity constraints could be watched by following the donors with minimal probability mass $\Delta\xi_t \leftarrow \max\{\xi \geq 0 \ \mid \ \boldsymbol{p}_t + \xi \cdot \nabla_\xi \boldsymbol{p}_{B_t}(\xi_t) \geq \mathbf{0}\}$. The rate of exchange for the trader varies at each basis, as a result, the trader could violate its lower and upper bound $-\xi \leq p_\tau - \bar{p}_\tau \leq \xi$. The algorithm trace the trader's rate so one can check the constrain via $\Delta\xi_t \leftarrow \max\{\xi \geq 0 \ \mid \ |(\boldsymbol{p}_t + \xi \cdot \nabla_\xi \boldsymbol{p}_{B_t}(\xi_t) - \bar{\boldsymbol{p}})_{\tau_t}| \leq \xi_t + \xi\}$. Line 6 of Algorithm 1 combines these constraints and relaxes others.

□

Proof of Theorem 3.9. A naive implementation of the homotopy method in Algorithm 1 has a computational complexity of $\mathcal{O}(S^2)$. The algorithm obtains the $\boldsymbol{p}^\star$ at each breakpoint. The number of iteration depends on the number of breakpoints in $q(\xi)$, which is at most $\frac{3}{2}S$. We observed numerically that the naive implementation performs on par with LP solvers and sometimes even slower. In Algorithm 3, we take advantage of the structural property of the slope of the q-function presented in Lemma 3.4, and only trace the optimal probability mass of the *trader* to speed up the method dramatically. Algorithm 3 compute q-function for each state-action pair in $\mathcal{O}(S \log S)$ for sorting the values of $\boldsymbol{z}$.

□

A.2 Detailed Homotopy Algorithm

This section provides the detailed procedure of our homotopy algorithm for computing the exact solution for robust Bellman operator with L_∞ constrained ambiguity sets. Algorithm 3 starts with the initialization of the doner, receiver, and trader sets according to Lemma 3.6, and then iterates through all breakpoints. Each breakpoint has been obtained concerning the conditions that are described in Lemma 3.7. The type of each basis is change is indicated according to Table 1. We use a priority queue to keep track of the donor with the smallest probability mass. The algorithm follows the value of q-function at each iteration, however ignores the probability mass values for all components except the trader. The iteration stops as soon as ξ exceeds the budget κ, which is given as an input.

A.3 Proofs of Results in Section 4

Proof of Theorem 4.1. The result follows from the complexity analysis of the bisection algorithm with quasi-linear time complexity in (Ho, Petrik, and Wiesemann, 2020), appendix B. □

Lemma A.3. *The optimal objective values of Equations* (7) *and* (8) *are equivalent.*

Algorithm 3 Homotopy method for $q(\kappa)$ with L_∞ constrained ambiguity set.

Input: LP parameters $\boldsymbol{z}$, κ and $\bar{\boldsymbol{p}}$.
Output: Breakpoints $(\xi_t)_{t=0,\ldots,T+1}$ and values $(q_t)_{t=0,\ldots,T+1}$;
Initialize $\xi_0 \leftarrow 0$, $t \leftarrow 0$, $\boldsymbol{p}_0 \leftarrow \bar{\boldsymbol{p}}$ and $q_0 \leftarrow q(\xi_0) = \boldsymbol{p}_0^\intercal \boldsymbol{z}$;
Sort $\boldsymbol{z}$ in ascending order and rearrange $\bar{\boldsymbol{p}}$ accordingly
Initialize the sets: $\mathcal{T} = \{\lceil S/2 \rceil\}$, $\mathcal{R} = \{i \mid i < \tau\}$, $\mathcal{D} = \{j \mid j > \tau\}$, $\mathcal{N} = \{\}$;
$z_\mathcal{R} = \sum_{i \in \mathcal{R}} z_i$; $\quad z_\mathcal{D} = \sum_{j \in \mathcal{D}} z_j$
Push all elements of $\mathcal{D}$ into a min-heap $\mathcal{H}$ according to their probability mass
$\xi \leftarrow \xi_0$
while $\xi < \kappa$ **do**
  $\dot{p}_\tau \leftarrow |\mathcal{D}| - |\mathcal{R}|$; # The trader's rate of exchange
  $j \leftarrow \mathcal{H}.top$
  $\Delta\xi_D \leftarrow p_j - \xi$
  $\Delta\xi_\tau \leftarrow$ Calculate largest feasible Δp_τ given $\dot{p}_\tau$
  if $\Delta\xi_\tau > \Delta\xi_D$ **then**
    Basis Change $\leftarrow \mathcal{D}$ *to* $\mathcal{N}$
    $\Delta\xi \leftarrow \Delta\xi_D$;
  else
    $\Delta\xi \leftarrow \Delta\xi_\tau$; $\;p'_\tau \leftarrow p_\tau + \dot{p}_\tau \cdot \Delta\xi$;
    if $p'_\tau = 0$ **then**
      Basis Change $\leftarrow \mathcal{T}$ *to* $\mathcal{N}$
    else
      Basis Change $\leftarrow \mathcal{T}$ *to* $\mathcal{D}$
    end if
  end if
  $\Delta\xi \leftarrow \max\{\Delta\xi, \kappa - \xi\}$;
  $p_\tau \leftarrow p_\tau + \dot{p}_\tau \cdot \Delta\xi$;
  $q_t = q_{t-1} + (z_\mathcal{R} - z_\mathcal{D} + \dot{p}_\tau z_\tau) \cdot \Delta\xi$
  $\xi \leftarrow \xi + \Delta\xi$; $\xi_t \leftarrow \xi$; $t \leftarrow t + 1$
  if Basis Change is $\mathcal{D}$ *to* $\mathcal{N}$ **then**
    $z_\mathcal{D} \leftarrow z_\mathcal{D} - z_j$;
    $\mathcal{D} = \mathcal{D} \backslash \{j\}$;
    $\mathcal{N} = \mathcal{N} \cup \{j\}$
    $\mathcal{H}.pop$
  else
    if Basis Change is $\mathcal{T}$ *to* $\mathcal{D}$ **then**
      $\mathcal{H}.push(\tau)$ # $p = p_\tau + \xi$
      $\mathcal{D} = \mathcal{D} \cup \{\tau\}$
      $z_\mathcal{D} \leftarrow z_\mathcal{D} + z_\tau$
    else if Basis Change is $\mathcal{T}$ *to* $\mathcal{N}$ **then**
      $\mathcal{N} = \mathcal{N} \cup \{\tau\}$
    end if
    $\tau \leftarrow \tau - 1$;
    $\mathcal{T} = \{\tau\}$
    $\mathcal{R} = \mathcal{R} \backslash \{\tau\}$
    $p_\tau \leftarrow \bar{p}_\tau + \xi$
    $z_\mathcal{R} \leftarrow z_\mathcal{R} - z_\tau$
  end if
end while
The remainder of the function $q(\xi)$ will be constant: $q_{T+1} \leftarrow q_t$
$\xi_{T+1} \leftarrow \infty$
Return: $(\xi_t)_{t=0,\ldots,T+1}$, and $(q_t)_{t=0,\ldots,T+1}$

Proof of Lemma A.3. Since the functions q_a, for all $a \in \mathcal{A}$ in Equation (8) are convex due to the LP formulation of Equation (6). We can exchange the maximization and minimization operators in Equation (8) to obtain

$$\min_{\boldsymbol{\xi} \in \mathbb{R}_+^A} \left\{ \max_{\boldsymbol{\pi} \in \Delta^A} \left(\sum_{a \in \mathcal{A}} \pi_a \cdot q_a(\xi_a) \right) \mid \sum_{a \in \mathcal{A}} \xi_a \leq \kappa \right\}, \tag{21}$$

Since the inner maximization is linear in $\boldsymbol{\pi}$, it is optimized at an extreme point of Δ^A. This allows us to re-express the optimization problem as

$$\min_{\boldsymbol{\xi} \in \mathbb{R}_+^A} \left\{ \max_{a \in \mathcal{A}} q_a(\xi_a) \mid \sum_{a \in \mathcal{A}} \xi_a \leq \kappa \right\}. \tag{22}$$

We can linearize the objective function in this problem by introducing the epigraphical variable $u \in \mathbb{R}$

$$\min_{u \in \mathbb{R}} \min_{\boldsymbol{\xi} \in \mathbb{R}_+^A} \left\{ u \mid \sum_{a \in \mathcal{A}} \xi_a \leq \kappa, u \geq \max_{a \in \mathcal{A}} [q_a(\xi_a)] \right\} \tag{23}$$

It can be readily seen that for a fixed u in the outer minimization, there is an optimal $\boldsymbol{\xi}$ in the inner minimization that minimizes each ξ_a a individually while satisfying $q_a(\xi_a) \leq u$ for all $a \in \mathcal{A}$. Define g_q as the a-th component of this optimal $\boldsymbol{\xi}$:

$$g_a(u) = \min_{\xi_a \in \mathbb{R}_+^A} \{\xi_a \mid q_a(\xi_a) \leq u\}. \tag{24}$$

We show that $g_a(u) = q_a^{-1}$. To see this, we substitude q_a in Equation (24) to get:

$$g_a(u) = \min_{\xi_a \in \mathbb{R}_+^A} \min_{\boldsymbol{p}_a \in \Delta^S} \left\{ \xi_a \mid \boldsymbol{p}_a^\mathsf{T} \boldsymbol{z}_a \leq u, \|\boldsymbol{p}_a - \bar{\boldsymbol{p}}_a\|_\infty \leq \xi_a \right\}. \tag{25}$$

The identity $g_a = q_a^{-1}$ then follows by realizing that the optimal $\xi_a^\star$ in the equation above must satisfy $\xi_a^\star = \|\boldsymbol{p}_a - \bar{\boldsymbol{p}}_a\|_\infty$. Finally, substituting the definition of g_a in Equation (24) into the problem (23) show that the optimization problem (8) is equivalent to Equation (7). □

Algorithm 4 Bisection method for the robust Bellman optimality operator (Ho, Petrik, and Wiesemann, 2020).

Input: Precision ϵ, functions $q_a^{-1}, \forall a \in \mathcal{A}$
$u_{\min}$: maximum known u for which Equation (7) is infeasible
$u_{\max}$: minimum known u for which Equation (7) is feasible
Output: $\hat{u}$ such that $|u^\star - \hat{u}| \leq \epsilon$, where $u^\star$ is optimal in Equation (7)
Return: $(u_{\min} + u_{\max})/2)$
while $u_{\max} - u_{\min} > 2\,\epsilon$ **do**
    Split interval $[u_{\min}, u_{\max}]$ in half: $u \leftarrow (u_{\min} + u_{\max})/2$
    Calculate the budget required to achieve the mid-point u: $s \leftarrow \sum_{a \in \mathcal{A}} q_a^{-1}(u)$
    if $s \leq \kappa$ **then**
        u is feasible: update the feasible upper bound: $u_{\max} \leftarrow u$
    else
        u is infeasible: update the infeasible lower bound: $u_{\min} \leftarrow u$
    end if
end while

B Detailed Description of Domains

In this section, we provide a detailed description of five standard reinforcement domains that have been previously used to evaluate robustness.

As the primary metric, we compare the running time of our homotopy and bisection algorithm with Gurobi 9.1.2—a standard LP solver. In order to enable the comparison of the results among different domains, we also compare our results with the homotopy and bisection algorithm for L_1-constrained ambiguity sets in (Ho, Petrik, and Wiesemann, 2020).

As the first benchmark, we employ Inventory Management (IM), a classic inventory management problem (Zipkin, 2000), with discrete inventory levels $0, \ldots, S = 30$. The purchase cost, sale price, and holding cost are $2.49, 3.99$, and 0.03, respectively. The demand is sampled from a normal distribution with a mean $S/4$ and a standard deviation of $S/6$. The initial state is 0 (empty stock). It also uses a Dirichlet prior. Table 2 summarizes the run-time for computed guaranteed returns of different methods at 0.95 confidence levels.

The second domain is RiverSwim (RS) which is a standard benchmark (Strehl and Littman, 2008), which is an MDP consisting of six states and two actions. The process follows by sampling synthetic datasets from the true model and then computing the guaranteed robust returns for different methods. The prior is a uniform Dirichlet distribution over reachable states.

Moreover, Machine Replacement (MR) is a small benchmark MDP problem with $S = 10$ states that models progressive deterioration of a mechanical device (Delage and Mannor, 2010). Two repair actions $A = 2$ are available and restore the machine's state.

C Fast Algorithm for Nature Response with Fixed ξ

Let us consider the optimization problem (3) with fixed $\xi > 0$:

$$\min_{\boldsymbol{p} \in \Delta^S} \{\boldsymbol{p}^\mathsf{T} \boldsymbol{z} : \|\bar{\boldsymbol{p}} - \boldsymbol{p}\|_\infty \leq \xi\}, \tag{26}$$

This problem was studied by Ibaraki and Katoh (1988). For the sake of completeness, in this section, we provide the computational procedure of solving this problem. As expressed earlier, the problem can be formulated as the following LP problem:

$$\begin{aligned} q(\xi) = \quad & \min_{\boldsymbol{p} \in \mathbb{R}^S} \quad \boldsymbol{z}^\mathsf{T} \boldsymbol{p} \\ & \text{s. t.} \quad -\boldsymbol{\xi} \leq \boldsymbol{p} - \bar{\boldsymbol{p}} \leq \boldsymbol{\xi} \\ & \qquad\quad \mathbf{1}^\mathsf{T} \boldsymbol{p} = 1,\ \boldsymbol{p} \geq \mathbf{0} \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} & \min_{\boldsymbol{p} \in \mathbb{R}^S} \quad \boldsymbol{z}^\mathsf{T} \boldsymbol{p} \\ & \text{s. t.} \quad \boldsymbol{l}' \leq \boldsymbol{p} \leq \boldsymbol{u}' \\ & \qquad\quad \mathbf{1}^\mathsf{T} \boldsymbol{p} = 1\ . \end{aligned} \tag{27}$$

Here, $\boldsymbol{l}' = \max\{\mathbf{0}, \boldsymbol{l}\}$ and $\boldsymbol{u}' = \min\{\mathbf{1}, \boldsymbol{u}\}$ where $\boldsymbol{l} = -\boldsymbol{\xi} + \bar{\boldsymbol{p}}$ and $\boldsymbol{u} = \boldsymbol{\xi} + \bar{\boldsymbol{p}}$. The problem (27) is a bounded resource allocation problem with continuous variables, where the objective function is convex and continuously differentiable. Without loss of generality we add the following restrictions:

First, $\boldsymbol{l}' < \boldsymbol{u}'$, since if $l'_j = u'_j$ for any $j \in \{1, \ldots, S\}$ implies that p_j is fixed and can be dropped from (27). Second, $\mathbf{1}^\mathsf{T} \boldsymbol{l}' < 1 < \mathbf{1}^\mathsf{T} \boldsymbol{u}'$. Otherwise the problems is either infeasible or trivially solvable. We consider the following equivalent problem, which obtained by change in variables $\boldsymbol{x} = \boldsymbol{p} - \boldsymbol{l}'$, and the modified upper bound $\boldsymbol{u} = \boldsymbol{u}' - \boldsymbol{l}'$. Let $\alpha = 1 - \mathbf{1}^\mathsf{T} \boldsymbol{l}'$:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^S} \quad & \boldsymbol{z}^\mathsf{T} \boldsymbol{x} \\ \text{s. t.} \quad & \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u} \\ & \mathbf{1}^\mathsf{T} \boldsymbol{x} = \alpha\ . \end{aligned} \tag{28}$$

To solve (28), we rely on the following relaxed problem

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^S} \quad & \boldsymbol{z}^\mathsf{T} \boldsymbol{x} \\ \text{s. t.} \quad & \mathbf{0} \leq \boldsymbol{x} \\ & \mathbf{1}^\mathsf{T} \boldsymbol{x} = \alpha\ . \end{aligned} \tag{29}$$

The above problem has a trivial solution; for example, one optimal solution is $x_i = \alpha$ for any one of $i \in \arg\min_j z_j$ and $x_j = 0$ otherwise. Therefore, one can efficiently solve the this relaxed problem (29) and check if the solution is feasible in (28). If it is feasible, then this solution is optimal in (28); otherwise, we can eliminate the associate variable x_i using the following lemma.

Lemma C.1. *Let $\hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_n)$ be the optimal solution of* (29)*. Then $\hat{x}_j \geq u_j$ implies that $x_j^\star = u_j$ holds in an optimal solution $\boldsymbol{x}^\star$ of* (28)*.*

The proof is provided by Ibaraki and Katoh (1988). This lemma allows us to fix the optimal $x_j^\star = u_j$ and remove it from (28) and (29), which α should be updated and be subtracted by u_j. We can apply the same strategy until the optimal solution of the (29) (after removing the known optimal x_j's) is also optimal in (28).