Convergence of Fast Policy Iteration in Markov Games and Robust MDPs

Keith Badger[1], **Jefferson Huang**[2], **Marek Petrik**[1]

[1]University of New Hampshire
[2]Naval Postgraduate School
keith.badger@unh.edu, jefferson.huang@nps.edu, mpetrik@cs.unh.edu

Abstract

Markov games and robust MDPs are closely related models that involve computing a pair of saddle point policies. As part of the long-standing effort to develop efficient algorithms for these models, the Filar-Tolwinski (FT) algorithm has shown considerable promise. As our first contribution, we demonstrate that FT may fail to converge to a saddle point and may loop indefinitely, even in small games. This observation contradicts the proof of FT's convergence to a saddle point in the original paper. As our second contribution, we propose Residual Conditioned Policy Iteration (RCPI). RCPI builds on FT, but is guaranteed to converge to a saddle point. Our numerical results show that RCPI outperforms other convergent algorithms by several orders of magnitude.

1 Introduction

Markov Games (MG) (Kallenberg 2022) and Robust MDPs (RMDPs) (Iyengar 2005; Wiesemann, Kuhn, and Rustem 2013; Ho, Petrik, and Wiesemann 2022) are two important models that generalize Markov Decision Processes (MDPs) (Puterman 2005). Markov games can model strategic adversaries that can act to minimize the agent's returns and are a common model in multi-agent reinforcement learning (Shou et al. 2022; Littman 1994). Similarly, RMDPs can model an adversarial nature that can perturb transition probabilities and rewards to minimize the agent's returns and are useful when making decisions with imperfect data-driven models (Lobo et al. 2023; Behzadian et al. 2021). In recent years, MGs and RMDPs have seen an increasing number of applications in machine and reinforcement learning, which has motivated the study of efficient algorithms for solving them (Pérolat et al. 2016; Ho, Petrik, and Wiesemann 2021, 2022; Behzadian, Petrik, and Ho 2021; Kaufman and Schaefer 2013; Winnicki and Srikant 2023).

Although basic algorithms, like value and policy iteration, adapt readily from MDPs to MGs and RMDPs, developing more efficient algorithms has been challenging. The efforts to adapt efficient optimistic policy iteration (OPI) algorithms, such as modified or fitted policy iteration, have been difficult. Many natural OPI algorithms proposed for MGs and RMDPs cycle among suboptimal policies, alternatively improving the minimization or maximization sides of the saddle point equilibrium. The lack of convergence is often counter-intuitive and has led to several incorrect convergence proofs in the literature (Condon 1993; Pérolat et al. 2016; Filar and Tolwinski 1991). The overarching reason is that OPI in MG and RMDPs do not monotonically improve the policy and its value function as in MDPs.

We make two main contributions in this paper. First, we show that a fast OPI method proposed in Filar and Tolwinski (1991) can terminate with an arbitrarily suboptimal policy. Pérolat et al. (2016) first identified a gap in the proof of correctness in Filar and Tolwinski (1991) but hypothesized the algorithm works nevertheless. In contrast, we show that the algorithm is inherently suboptimal.

Second, we propose and analyze *Residual Conditioned Policy Iteration* (RCPI). RCPI is a new, simple approximate policy iteration algorithm for solving MGs and RMDPs that is guaranteed to converge to optimal policies. It builds on earlier efficient OPI algorithms (Filar and Tolwinski 1991; Pérolat et al. 2016; Ho, Petrik, and Wiesemann 2021; Winnicki and Srikant 2023) and combines them with an adaptive correction step. Our theoretical analysis shows that RCPI matches the worst-case computational complexity of value iteration. Our numerical results show that on a wide range of problems, RCPI outperforms other convergent algorithms by several orders of magnitude, even in moderately sized problems.

In this paper, we restrict our focus to model-based algorithms for MGs and RMDPs. It is important to note that this setting differs from online algorithms for solving games and multi-agent reinforcement learning problems, such as in Zhang et al. (2022). Although some of the issues that need to be overcome in online and model-based solvers are similar, we leave the study of the exact relationship between online and model-based algorithms for future work.

The remainder of the paper is organized as follows. Section 2 positions our work in the context of prior algorithmic developments for MGs and RMDPs. Then, Section 3 describes the formal framework for MGs and RMDPs. Section 4 describes our first contribution, which is to show that an existing OPI algorithm (Filar and Tolwinski 1991) may fail with an arbitrarily suboptimal policy. Section 5 describes our second and main contribution, the RCPI algorithm, along with its convergence rate and computational complexity analysis. Finally, our numerical results in Section 6 compare RCPI

with existings algorithms for solving MGs and RMDPs.

2 Prior Work: Solving MGs and RMDPs

In this section, we summarize prior efforts on developing OPI algorithms for MGs and RMDPs. We note that MG and RMDP communities have been largely separate, though the similarities between them have been noted and exploited previously (Iyengar 2005; Grand-Clément and Petrik 2024; Grand-Clément, Petrik, and Vieille 2025).

Value iteration is a simple convergent algorithm for solving MDPs, RMDPs, and MGs, but can be very slow in many practical settings (Puterman 2005). Many faster convergent algorithms for MDPs exist, such as policy iteration or modified policy iteration. Since the early days of MG (Condon 1993) and RMDPs (Iyengar 2005; Kaufman and Schaefer 2013), researchers have sought to generalize the ideas of modified policy iteration from MDPs to MGs and RMDPs. However, the attempts to speed up policy iteration while guaranteeing convergence have been largely unsuccessful (Pérolat et al. 2016). Existing algorithms are either too slow for larger problems or lack optimality guarantees.

Policy iteration (Puterman 2005), another basic MDP algorithm, can dramatically reduce the number of Bellman operator evaluations and compute the optimal policy in strongly polynomial time in MDPs (Ye 2011). Hoffman-Karp algorithm, also known as robust policy iteration (Iyengar 2005), for MGs and RMDPs adapts policy iteration to MGs and RMDPs. Although Hoffman-Karp has polynomial worst-case time complexity (Hansen, Miltersen, and Zwick 2013), it can be slower than value iteration in practice. Each Hoffman-Karp policy evaluation requires computing the adversarial agent's optimal policy. That is a significant increase in the complexity of the policy evaluation step in MDPs, which entails solving a system of linear equations.

Optimistic policy iteration (OPI) methods, such as modified policy iteration, accelerate policy iteration by performing the evaluation step approximately (Puterman 2005). In MDPs, OPI algorithms dramatically improve empirical performance while preserving the worst-case convergence rate of value iteration. In MGs and RMDPs, many natural OPI algorithms attain good empirical performance but fail to compute optimal policies (Condon 1993). For instance, Pollatschek Avi-Itzhak (PAI) algorithm holds the adversarial policy constant in the policy evaluation step, which is quicker than Hoffman-Karp, but may lead to infinitely looping over suboptimal policies (Van der Wal 1978).

One well-known attempt to fix PAI's non-convergence is the Filar-Tolwinski (FT) algorithm (Filar and Tolwinski 1991). It leverages the observation that PAI can be seen as Newton's method on the L_2 norm of the Bellman residual. FT replaces the pure Newton's method of PAI with the modified Newton's method, which uses Armijo's rule when deciding the step size in the value function update. While (Filar and Tolwinski 1991) claims that this resolves the cycling issues found with PAI, we show that FT may not converge. We discuss this issue in more detail in Section 4.

Recent years have seen several notable attempts to develop algorithms that match the empirical performance of PAI while guaranteeing convergence to an optimal policy. Robust Modified Policy Iteration (RMPI) (Kaufman and Schaefer 2013) and Partial Policy Iteration (PPI) (Ho, Petrik, and Wiesemann 2021) modify Hoffman-Karp to evaluate the adversarial policy approximately. RMPI uses a fixed-precision approximation, while PPI adapts the evaluation throughout the algorithm's execution. Numerical evidence suggests that PPI outperforms RMPI (Ho, Petrik, and Wiesemann 2021). The Winnicki-Srikant (WS) algorithm combines value iteration steps with policy backup steps and proposes ratios that guarantee the algorithm's convergence (Winnicki and Srikant 2023).

3 Preliminaries: MGs and Robust MDPs

In this section, we define Markov games and robust Markov Decision Processes formally and describe the properties we use to derive our main results.

3.1 Notation

The symbols $\mathbb{R}$ and $\mathbb{N}$ denote the sets of real and natural (including 0) numbers. Vectors are denoted with a lower-case bold font, such as $\boldsymbol{x} \in \mathbb{R}^n$, and $x_i, i = 1, \ldots, n$ is the i-th element of the vector. Matrices are denoted in uppercase bold font. Sets are denoted with calligraphic letters. We use the notation $\mathbb{R}^{\mathcal{Z}}$ to denote the set of all functions $f\colon \mathbb{R} \mapsto \mathcal{Z}$, and interpret each f equivalently as a vector $\boldsymbol{f}$ such that $f_z = f(z)$. The notation $\Delta^{\mathcal{Z}} := \left\{\boldsymbol{x} \in \mathbb{R}^{\mathcal{Z}} \mid \mathbf{1}^\mathsf{T}\boldsymbol{x} = 1, \boldsymbol{x} \geq \mathbf{0}\right\}$ refers to the set of probability distributions over the finite non-empty set $\mathcal{Z}$.

To streamline our notation, we define the *ϵ-saddle-point operator* $\mathfrak{S}_\epsilon\colon \mathbb{R}^{\mathcal{X}\times\mathcal{Y}} \to 2^{\mathcal{X}\times\mathcal{Y}}$ for any tolerance $\epsilon \geq 0$ and an objective function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ as

$$\begin{aligned}\mathfrak{S}_\epsilon(f) \;:=\; \Big\{&(x^\star, y^\star) \in \mathcal{X} \times \mathcal{Y} \mid \\ &f(x, y^\star) - \epsilon \leq f(x^\star, y^\star) \leq f(x^\star, y) + \epsilon, \\ &\forall x \in \mathcal{X}, y \in \mathcal{Y}\Big\},\end{aligned} \tag{1}$$

where $\mathcal{X}, \mathcal{Y}$ are arbitrary sets. Note that the first parameter of f is maximized, and the second one is minimized. The intuitive explanation of this definition is that $x^\star$ and $y^\star$ are ϵ-optimal responses to each other. In the remainder of the paper, we shorten $\mathfrak{S} := \mathfrak{S}_0$. Note that if $\epsilon_1 \leq \epsilon_2$ then $\mathfrak{S}_{\epsilon_1}(f) \subseteq \mathfrak{S}_{\epsilon_2}(f)$. We also allow $\mathfrak{S}_0$ to be used with objective function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ for some partially ordered set $\mathcal{Z}$.

If the function f is real-valued and bi-linear, then an element of $\mathfrak{S}(f)$ can be computed using the standard linear problem formulation of matrix games, see for example (Kallenberg 2022, section 10.1.3).

3.2 Markov Games

Markov games extend Markov decision processes to a zero-sum game-theoretic setting (Kallenberg 2022; Filar and Vrieze 1997) and can model multi-agent reinforcement learning. An imperfect-information *Markov game* is defined as $(\mathcal{S}, \mathcal{A}, \mathcal{B}, r, P, s_0)$ where $\mathcal{S} = \{1, \ldots, S\}$ is the finite non-empty set of states that the agents share, $\mathcal{A} = \{1, \ldots, A\}$ is the finite non-empty set of actions for the primary agent, $\mathcal{B} =$

$\{1, \ldots, B\}$ is the finite non-empty set of actions for the adversarial agent. The function $r\colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to [-r_{\max}, r_{\max}]$ for $r_{\max} \in \mathbb{R}$ represents the rewards the primary agent seeks to maximize and the adversarial agent seeks to minimize. The function $P\colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \Delta^{\mathcal{S}}$ is the transition probability function, where $p(s, a, b, s')$ is the probability of transitioning from state s to state s' after the agents take actions action a and b, respectively. Finally, $s_0 \in \mathcal{S}$ is the initial state.

We consider *infinite-horizon discounted rewards* for a discount factor $\gamma \in (0, 1)$, and restrict attention to randomized stationary policies $\Pi := (\Delta^{\mathcal{A}})^{\mathcal{S}}$ and $\Sigma := (\Delta^{\mathcal{B}})^{\mathcal{S}}$ for the maximizing and minimizing agents, respectively. Note that the restriction to randomized stationary policies is not limiting, because neither of the players can benefit from using Markov or history-dependent policies (Kallenberg 2022; Filar and Vrieze 1997). The value function $\boldsymbol{v}^{\boldsymbol{\pi},\boldsymbol{\sigma}} \in \mathbb{R}^S$ associated with each $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\sigma} \in \Sigma$ as (Filar and Vrieze 1997):

$$v_s^{\boldsymbol{\pi},\boldsymbol{\sigma}} \;:=\; \mathbb{E}^s_{\boldsymbol{\pi},\boldsymbol{\sigma}}\left[\sum_{t=0}^{\infty} \gamma^t r(\tilde{s}_t, \tilde{a}_t, \tilde{b}_t)\right], \quad \forall s \in \mathcal{S}. \tag{2}$$

The superscripts and subscripts of $\mathbb{E}^s_{\boldsymbol{\pi},\boldsymbol{\sigma}}$ indicate that the probability measure is chosen such that $\tilde{s}_0 = s$ and that $\tilde{a}_t \sim \boldsymbol{\pi}(\tilde{s}_t)$, $\tilde{b}_t \sim \boldsymbol{\sigma}(\tilde{s}_t)$, and $\tilde{s}_{t+1} \sim P(\tilde{s}_t, \tilde{a}_t, \tilde{b}_t, \cdot)$ for all $t \in \mathbb{N}$. In general, we adorn random variables with a tilde. The equilibrium value function $\boldsymbol{v}^\star \in \mathbb{R}^S$ is defined as the saddle point over policy pairs:

$$v_s^\star \;:=\; \max_{\boldsymbol{\pi}\in\Pi} \min_{\boldsymbol{\sigma}\in\Sigma} v_s^{\boldsymbol{\pi},\boldsymbol{\sigma}}, \qquad \forall s \in \mathcal{S}. \tag{3}$$

That is, the agents seek to compute the saddle point of the infinite-horizon discounted objective function $\rho_{\mathrm{G}}\colon \mathcal{S} \times \Pi \times \Sigma \to \mathbb{R}$ for some tolerance $\epsilon \geq 0$:

$$(\boldsymbol{\pi}^\star, \boldsymbol{\sigma}^\star) \in \mathfrak{S}_\epsilon(\rho), \text{ where } \rho_{\mathrm{G}}(s_0, \boldsymbol{\pi}, \boldsymbol{\sigma}) \;:=\; v_{s_0}^{\boldsymbol{\pi},\boldsymbol{\sigma}}. \tag{4}$$

Given any $\epsilon \geq 0$, the existence of an equilibrium pair $(\boldsymbol{\pi}^\star, \boldsymbol{\sigma}^\star)$ is guaranteed for discounted Markov games with finite state and action sets (Kallenberg 2022, corollary 10.1).

Next, we describe the *Bellman operator* for Markov games. For each $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\sigma} \in \Sigma$, we define the reward vector $\boldsymbol{r}^{\boldsymbol{\pi},\boldsymbol{\sigma}} \in \mathbb{R}^S$ and a transition matrix $\boldsymbol{P}^{\boldsymbol{\pi},\boldsymbol{\sigma}} \in \mathbb{R}_+^{S\times S}$ as

$$\boldsymbol{r}_s^{\boldsymbol{\pi},\boldsymbol{\sigma}} := \sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} \pi_a(s) \cdot \sigma_b(s) \cdot r(s, a, b),$$
$$\boldsymbol{P}_{s,s'}^{\boldsymbol{\pi},\boldsymbol{\sigma}} := \sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} \pi_a(s) \cdot \sigma_b(s) \cdot p(s, a, b, s').$$

Then, the Bellman evaluation operator $\mathfrak{T}^{\boldsymbol{\pi},\boldsymbol{\sigma}}\colon \mathbb{R}^S \to \mathbb{R}^S$ is defined for each $\boldsymbol{v} \in \mathbb{R}^n$ and $s \in \mathcal{S}$ as

$$\mathfrak{T}_s^{\boldsymbol{\pi},\boldsymbol{\sigma}}\boldsymbol{v} \;:=\; r_s^{\boldsymbol{\pi},\boldsymbol{\sigma}} + \gamma \cdot \boldsymbol{P}_s^{\boldsymbol{\pi},\boldsymbol{\sigma}}\boldsymbol{v}\,. \tag{5}$$

For all operators, we use the shorthand $\mathfrak{T}\boldsymbol{v}_s := (\mathfrak{T}\boldsymbol{v})_s$. The Bellman equilibrium operator $\mathfrak{T}^\star\colon \mathbb{R}^S \to \mathbb{R}^S$ is defined as $\mathfrak{T}_s^\star\boldsymbol{v} := \max_{\boldsymbol{\pi}\in\Pi} \min_{\sigma\in\Sigma} \mathfrak{T}_s^{\boldsymbol{\pi},\boldsymbol{\sigma}}\boldsymbol{v}$. The Bellman policy operator $\mathfrak{B}^\star\colon \mathbb{R}^S \to 2^{\Pi\times\Sigma}$ computes the saddle point policies and is defined as

$$\mathfrak{B}^\star\boldsymbol{v} \;:=\; \mathfrak{S}((\boldsymbol{\pi}, \boldsymbol{\sigma}) \mapsto \mathfrak{T}^{\boldsymbol{\pi},\boldsymbol{\sigma}}\boldsymbol{v}), \tag{6}$$

where the partial order on the value functions is defined as $\boldsymbol{u} \leq \boldsymbol{v} \Leftrightarrow u_s \leq v_s, \forall s \in \mathcal{S}$.

Bellman operators can be used to compute both $\boldsymbol{v}^{\boldsymbol{\pi},\boldsymbol{\sigma}}$ for any $(\boldsymbol{\pi}, \boldsymbol{\sigma}) \in \Pi \times \Sigma$, as well as $\boldsymbol{v}^\star$. These value functions defined in (2) and (3) are the *unique* solutions for each $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\sigma} \in \Sigma$ to, respectively (Kallenberg 2022, corollary 10.1),

$$\boldsymbol{v}^{\boldsymbol{\pi},\boldsymbol{\sigma}} = \mathfrak{T}^{\boldsymbol{\pi},\boldsymbol{\sigma}}\boldsymbol{v}^{\boldsymbol{\pi},\boldsymbol{\sigma}}, \qquad \boldsymbol{v}^\star = \mathfrak{T}^\star\boldsymbol{v}^\star.$$

The Bellman operators $\mathfrak{T}^{\boldsymbol{\pi},\boldsymbol{\sigma}}$ and $\mathfrak{T}^\star$ are monotone and γ-contractive in the L_∞ norm (Kallenberg 2022, theorem 10.5). Because solutions to saddle points can be computed only approximately in polynomial time, we also define *approximate Bellman equilibrium operator* $\mathfrak{T}^\delta\colon \mathbb{R}^S \to \mathbb{R}^S$ which satisfies that

$$\|\mathfrak{T}^\delta\boldsymbol{v} - \mathfrak{T}^\star\boldsymbol{v}\|_\infty \leq \delta, \qquad \forall \boldsymbol{v} \in \mathbb{R}^S, \tag{7}$$

and the *approximate Bellman policy operator* $\mathfrak{B}^\delta\colon \mathbb{R}^S \to 2^{\Pi\times\Sigma}$ which satisfies

$$\mathfrak{B}^\delta\boldsymbol{v} \subseteq \mathfrak{S}_\delta((\boldsymbol{\pi}, \boldsymbol{\sigma}) \mapsto \mathfrak{T}^{\boldsymbol{\pi},\boldsymbol{\sigma}}\boldsymbol{v}).$$

The well-known *value iteration* is the simplest method for computing $\boldsymbol{v}^\star$ iteratively as $\boldsymbol{v}^{k+1} = \mathfrak{T}^\star\boldsymbol{v}^k$, where it is well-known that $\lim_{k\to\infty} \boldsymbol{v}^k = \boldsymbol{v}^\star$. It's worth noting that $\mathfrak{T}^\star$ is typically replaced with $\mathfrak{T}^\delta$ which has similar convergence properties.

Computing the exact equilibrium is often unnecessary. Instead, it may be sufficient to compute an ϵ-equilibrium for a sufficiently small ϵ. To evaluate how close the value function is to the equilibrium, it is convenient to define the *Bellman residual* $\psi_p\colon \mathbb{R}^S \to \mathbb{R}$ as

$$\psi_p(\boldsymbol{v}) \;:=\; \|\mathfrak{T}^\star\boldsymbol{v} - \boldsymbol{v}\|_p, \qquad p \in \{1, 2, \infty\},$$

and the *approximate Bellman residual* as

$$\psi_p^\delta(\boldsymbol{v}) \;:=\; \|\mathfrak{T}^\delta\boldsymbol{v} - \boldsymbol{v}\|_p.$$

The following proposition shows that we can obtain ϵ-equilibrium policies from a value function that approximates the equilibrium value function.

Proposition 3.1. *For each $\boldsymbol{v} \in \mathbb{R}^S$:*

$$\emptyset \neq \mathfrak{B}^\star\boldsymbol{v} \;\subseteq\; \mathfrak{S}_\epsilon(\rho_{\mathrm{G}}), \quad \textit{where} \quad \epsilon = \frac{2\gamma}{1-\gamma}\psi_\infty(\boldsymbol{v}).$$

The proof, which we include in the appendix of Badger, Huang, and Petrik (2025) for the sake of completeness, follows standard arguments; see, for example, (Kallenberg 2022, theorem 10.11). We note that the bound in Proposition 3.1 is tighter than the bounds given, for example, in Ho, Petrik, and Wiesemann (2021, corollary A.4) and Williams and Baird (1993, theorems 3.1, 3.2).

3.3 Robust MDPs

RMDPs generalize MDPs to allow for adversarial perturbations to the transition probabilities. We consider s-rectangular RMDPs $(\mathcal{S}, \mathcal{A}, r, \mathcal{P}, s_0)$ where $\mathcal{S}$ and $\mathcal{A}$ are the finite non-empty sets of states and actions, respectively (Wiesemann, Kuhn, and Rustem 2013; Ho, Petrik, and Wiesemann 2021), and $r\colon \mathcal{S} \times \mathcal{A} \to [-r_{\max}, r_{\max}]$ is the reward function. The ambiguity set $\mathcal{P} := (\mathcal{P}_s)_{s\in\mathcal{S}}$, where $\mathcal{P}_s \subseteq \Delta^{\mathcal{S}}$ is compact

and non-empty for each $s \in \mathcal{S}$, and determines the range of possible adversarial transition probability functions. Finally, the initial state is s_0.

It is common to define the ambiguity sets in RMDPs as bounded norm-balls around a given nominal transition function $\bar{\boldsymbol{p}}\colon \mathcal{S} \times \mathcal{A} \to \Delta^S$, such as (Ho, Petrik, and Wiesemann 2021, 2022; Behzadian et al. 2021; Behzadian, Petrik, and Ho 2021)

$$\mathcal{P}_s := \left\{ \boldsymbol{p} \in (\Delta^{\mathcal{S}})^{\mathcal{A}} \mid \sum_{a \in \mathcal{A}} \|\boldsymbol{p}(a) - \bar{\boldsymbol{p}}(s,a)\| \leq \xi_s \right\},$$

for some norm $\|\cdot\|$ and $\xi_s \geq 0$, $s \in \mathcal{S}$. In this work, we focus on ambiguity sets defined by the L_1-norm.

As with Markov games as defined above, we seek to compute a stationary policy $\pi \in \Pi$ that maximizes the expected γ-discounted infinite-horizon robust return:

$$\begin{aligned} &\max_{\boldsymbol{\pi} \in \Pi} \min_{\boldsymbol{p} \in \mathcal{P}} \rho_{\mathrm{R}}(\boldsymbol{\pi}, \boldsymbol{p}), \\ &\rho_{\mathrm{R}}(\boldsymbol{\pi}, \boldsymbol{p}) := \mathbb{E}^{s_0}_{\boldsymbol{\pi}, \boldsymbol{p}} \left[\sum_{t=0}^{\infty} \gamma^t r(\tilde{s}_t, \tilde{a}_t) \right], \end{aligned} \tag{8}$$

where $\gamma \in (0,1)$. We emphasize that for each $\boldsymbol{\pi} \in \Pi$, the domain of $\rho_{\mathrm{R}}(\boldsymbol{\pi}, \cdot)$ is the set of feasible transition probabilities $\mathcal{P}$, rather than the set of all transition probabilities. An optimal policy $\boldsymbol{\pi}^\star$ in (8) exists and can be computed by robust value or policy iteration (Wiesemann, Kuhn, and Rustem 2013; Iyengar 2005).

The following proposition shows that the concept of approximate optimality for RMDPs is closely related to the concept of approximate saddle points in games.

Proposition 3.2. *Suppose that* $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{p}}) \in \mathfrak{S}_\epsilon(\rho_{\mathrm{R}})$ *for some* $\epsilon \geq 0$. *Then* $\hat{\boldsymbol{\pi}}$ *is* 2ϵ-robust optimal *in the sense that*

$$\min_{\boldsymbol{p} \in \mathcal{P}} \rho_{\mathrm{R}}(\hat{\boldsymbol{\pi}}, \boldsymbol{p}) \;\geq\; \min_{\boldsymbol{p} \in \mathcal{P}} \rho_{\mathrm{R}}(\boldsymbol{\pi}^\star, \boldsymbol{p}) - 2 \cdot \epsilon.$$

For RMDPs, value functions and Bellman operators are defined analogously to how they are defined for MGs. For more detail, please see Badger, Huang, and Petrik (2025). In the remainder of the paper, we describe the algorithms for MGs which generalize to RMDPs.

Computationally, the main difference between RMDPs and MGs is in computing the Bellman operator. For most common ambiguity sets $\mathcal{P}$, the robust Bellman operator can be implemented by solving a convex optimization problem (Wiesemann, Kuhn, and Rustem 2013). For the L_1-bound ambiguity sets, the robust Bellman operator can be implemented by solving a linear program (Ho, Petrik, and Wiesemann 2021, appendix C). Significantly more efficient methods exist for ambiguity sets bounded by norms and φ-divergences (Ho, Petrik, and Wiesemann 2021, 2022; Behzadian, Petrik, and Ho 2021).

4 Filar-Tolwinski Algorithm May Not Converge

Pollatschek and Avi-Itzhak (1969) proposed one of the first alternatives to value iteration (Shapley 1953) for solving MGs. This algorithm, which we refer to as the PAI algorithm, can be viewed as applying Newton's method to the problem of finding a zero of $\psi_2(\boldsymbol{v})^2$. While PAI is known to converge to the optimal value function $\boldsymbol{v}^\star$ under certain restrictive conditions (Pollatschek and Avi-Itzhak 1969, theorem 5), it is also known to not converge at all for certain MGs (Van der Wal 1978). Filar and Tolwinski (Filar and Tolwinski 1991) proposed a modified Newton method intended to fix this convergence issue. In this section, we provide a counterexample to Filar and Tolwinski (1991, theorem 3.3), where it is claimed that the modified Newton method converges from some constant initial vector to $\boldsymbol{v}^\star$. The Filar-Tolwinski (FT) algorithm is described in Algorithm 1.

Algorithm 1: Filar-Tolwinski (FT) Algorithm

Input: Initial value $\boldsymbol{v}^0$, tolerance ϵ, backtracking line search coefficients $\beta \in (0,1), \delta \in (0,1)$
Output: $(\boldsymbol{\pi}, \boldsymbol{\sigma}) \in \mathfrak{S}_\epsilon(\rho_{\mathrm{G}})$

1 $k \leftarrow 0$;
2 **repeat**
3 $\quad k \leftarrow k + 1$;
4 $\quad$ Select $(\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k) \in \mathfrak{B}^\star \boldsymbol{v}^{k-1}$;
5 $\quad \boldsymbol{d}^k \leftarrow (\boldsymbol{I} - \gamma \boldsymbol{P}^{\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k})^{-1} \boldsymbol{r}^{\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k} - \boldsymbol{v}^{k-1}$;
$\quad$ `// Line search, Armijo's rule:`
$\quad$ // $\nabla \psi_2(\boldsymbol{v})^2 = 2(\gamma \boldsymbol{P}^{\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k} - \boldsymbol{I})^\mathsf{T} (\mathfrak{T}^\star \boldsymbol{v} - \boldsymbol{v})$
6 $\quad i_k \leftarrow \min\{ i \in \mathbb{N} \mid \psi_2(\boldsymbol{v}^{k-1} + \beta^i \boldsymbol{d}^k)^2 \leq$
7 $\qquad \leq \psi_2(\boldsymbol{v}^{k-1})^2 + \delta \beta^i \cdot (\boldsymbol{d}^k)^\mathsf{T} \nabla \psi_2(\boldsymbol{v}^{k-1})^2 \}$;
8 $\quad \boldsymbol{v}^k \leftarrow \boldsymbol{v}^{k-1} + \beta^{i_k} \cdot \boldsymbol{d}^k$;
9 **until** $\frac{2\gamma}{1-\gamma} \cdot \psi_\infty(\boldsymbol{v}^k) \leq \epsilon$;
10 **return** $(\boldsymbol{v}^k, \boldsymbol{\pi}^k, \boldsymbol{\sigma}^k)$;

To derive the FT algorithm, one interprets PAI as the pure Newton's method for solving $\min_{\boldsymbol{v} \in \mathbb{R}^S} \psi_2(\boldsymbol{v})^2$ (Filar and Tolwinski 1991; Filar and Vrieze 1997). Recall that the pure Newton's method direction $\boldsymbol{d}^k$ in iteration $k \in \mathbb{N}$ is

$$\begin{aligned} \boldsymbol{d}^k &:= -(\nabla^2 \psi_2(\boldsymbol{v}^{k-1})^2)^{-1} \nabla \psi_2(\boldsymbol{v}^{k-1})^2 \\ &= \left(\boldsymbol{I} - \gamma \boldsymbol{P}^{\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k} \right)^{-1} \left(\mathfrak{T}^\star \boldsymbol{v}^{k-1} - \boldsymbol{v}^{k-1} \right). \end{aligned}$$

FT's insight is to replace the pure Newton's step size of 1 in PAI with a backtracking line search. Setting the step size in Line 7 in Algorithm 1 to $i_k = 0$ recovers PAI exactly. The use of Armijo's rule in determining FT ensures that the objective function $\psi_2(\boldsymbol{v})^2$ decreases in every step. Since $\psi_2(\boldsymbol{v}^\star) = 0$ is the unique global minimum of $\boldsymbol{v} \mapsto \psi_2(\boldsymbol{v})^2$ and each of FT's iterations decreases the objective function, FT cannot cycle and does not terminate until reaching the optimal value function $\boldsymbol{v}^\star$.

Theorem 3.3 in (Filar and Tolwinski 1991) states that Algorithm 1 is guaranteed to converge to the optimal value function. However, there is a gap in the proof. In particular, while each step of the iteration reduces the value function, it is not guaranteed that a step size satisfying Armijo's rule exists. Since the gradient of $\boldsymbol{v} \mapsto \psi_2(\boldsymbol{v})^2$ may be discontinuous, it is possible that no i in Line 5 in Algorithm 1 satisfies the inequality; leading to an infinite loop in the search for the step size. We construct a simple MDP example demonstrating

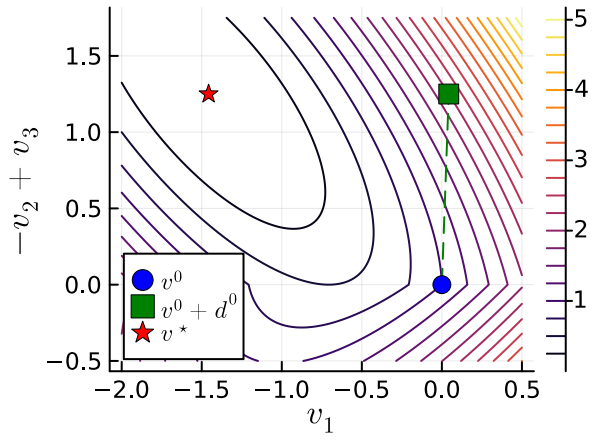

Figure 1: Plot of $\psi_2(\boldsymbol{v})^2$ projected onto the plane that spans the initial value function, optimal value function, and the step direction.

s_1	
b_1	b_2
$-\sqrt{2}/2$	$-\sqrt{2}/2$
$[0, 0, 1]$	$[0, 1, 0]$

s_2
b_1
$-1/2$
$[0, 1, 0]$

s_3
b_1
$1/2$
$[0, 0, 1]$

Figure 2: Rewards and transition probabilities of the Markov game for states s_1, s_2, s_3 from Example 4.1.

this behavior to show that this can happen.

Example 4.1. Consider a MG with $\mathcal{S} = \{s_1, s_2, s_3\}$, $\mathcal{A} = \{a_1\}$, and $\mathcal{B} = \{b_1, b_2\}$. The transition probabilities and rewards are defined in Figure 2. The columns represent actions. When only one column exists in a state, all actions behave identically. The top row of each cell represents the reward associated with the action, and the bottom row represents the transition probability function for that state and action. The discount factor is $\gamma = 0.6$.

The following theorem formally states that Example 4.1 is a counterexample to the optimality of FT.

Theorem 4.2. *FT in Algorithm 1 initialized to $\boldsymbol{v}^0 = \mathbf{0}$ and applied to the MG in Example 4.1 visits only suboptimal policies and never terminates.*

We note that Filar and Tolwinski (1991, theorem 3.3) assumes that FT is initialized to a constant value determined by the maximum reward instead of a zero vector. However, the algorithm makes no progress even with such initialization as shown in the appendix of Badger, Huang, and Petrik (2025).

We now discuss the gap in the proof of convergence in Filar and Tolwinski (1991, theorem 3.3) which Theorem 4.2 contradicts. As noted in Filar and Tolwinski (1991, theorem 2.1) the function $\boldsymbol{v} \mapsto \psi_2(\boldsymbol{v})^2$ is differentiable almost everywhere. However, because the function is not differentiable everywhere, Armijo's rule fails to find a positive step size. Example 4.1 initializes FT in exactly a point of non-differentiability. One attempt to circumvent this problem would be to argue that the probability of being at a point of non-differentiability is zero. However, it may be possible to modify our example so that the initialization does not happen at a point of non-differentiability. Yet, the line search method will take increasingly smaller steps, such that it approaches the point of non-differentiability without ever passing it. Because it is unclear how one may rectify the non-differentiability to ensure Newton's method's convergence, we propose an alternative approach in the following section.

Algorithm 2: RCPI: Residual Conditioned PI

Input: Initial value $\boldsymbol{v}^0$, tolerance ϵ, backup tolerance $\delta < \epsilon \cdot \frac{(1-\gamma)^2}{2\gamma(3+\gamma)}$, max recovery steps $m \in \mathbb{N}$

Output: $(\boldsymbol{\pi}, \boldsymbol{\sigma}) \in \mathfrak{S}_\epsilon(\rho_{\mathrm{G}})$

$k \leftarrow 0$;
repeat
  $k \leftarrow k + 1$;
  Select $(\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k) \in \mathfrak{B}^\delta \boldsymbol{v}^{k-1}$;
  $\boldsymbol{u}^{k,0} \leftarrow (\boldsymbol{I} - \gamma \boldsymbol{P}^{\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k})^{-1} \boldsymbol{r}^{\boldsymbol{\pi}^k, \boldsymbol{\sigma}^k}$;
  if $\gamma^{m-1} \psi^\delta_\infty(\boldsymbol{u}^{k,0}) + \frac{2(1+\gamma)\delta}{1-\gamma} > \psi^\delta_\infty(\boldsymbol{v}^{k-1})$ **then**
    $\boldsymbol{v}^k \leftarrow \mathfrak{T}^\delta \boldsymbol{v}^{k-1}$;
  else
    $l \leftarrow 0$;
    while $\psi^\delta_\infty(\boldsymbol{u}^{k,l}) > \gamma \psi^\delta_\infty(\boldsymbol{v}^{k-1}) + 2(1+\gamma)\delta$ **do** $\boldsymbol{u}^{k,l+1} \leftarrow \mathfrak{T}^\delta \boldsymbol{u}^{k,l}$; $l \leftarrow l + 1$;
    $\boldsymbol{v}^k \leftarrow \boldsymbol{u}^{k,l}$;
until $\frac{2\gamma}{1-\gamma} \left(\psi^\delta_\infty(\boldsymbol{v}^k) + \delta\right) \le \epsilon$;
return $(\boldsymbol{v}^k, \boldsymbol{\pi}^k, \boldsymbol{\sigma}^k)$;

5 RCPI: Residual Conditioned Policy Iteration

In this section, we propose and analyze a new algorithm, RCPI, for solving MGs and RMDPs. RCPI builds on the strengths of PAI and FT but with convergence guarantees. Our theoretical analysis demonstrates that RCPI is guaranteed to converge to the optimal value function at a rate that at least matches that of value iteration.

RCPI, summarized in Algorithm 2, can be viewed as a direct modification of FT in Algorithm 1. The first two steps of RCPI's iteration are identical to FT. First, RCPI jointly updates both the primary and adversarial policies to be greedy with respect to the current value function. Second, RCPI evaluates the value function for the updated policies. Simply adopting this value function would lead to PAI (See the appendix of Badger, Huang, and Petrik (2025)), which is prone to getting stuck in infinite cycles (Van der Wal 1978). Such infinite cycles must involve steps that do not decrease the residual. RCPI detects when the residual does not decrease sufficiently and reverts to a value function update to guarantee its reduction. As a result, RCPI will never cycle or terminate before reaching the optimal value function.

RCPI guarantees convergence to the optimal value function as follows. Each iteration of the outer loop guarantees that the residual of the incumbent value function decreases

at least by the factor γ. The parameter m determines how reduction is achieved. If the residual of the proposed value function can be reduced in at most m steps of value iteration, then the Bellman operator is applied until the reduction is achieved. Otherwise, the proposed value function is discarded and replaced by a plain value iteration update.

We now turn to the proof of RCPI's correctness and computation complexity. First, we need to discuss the worst-case runtime of the Bellman backups. For s-rectangular L_1 robust MDPs the runtime of computing $\mathfrak{T}^\delta \boldsymbol{v}$ and $\mathfrak{B}^\delta \boldsymbol{v}$ is (Ho, Petrik, and Wiesemann 2021)

$$T_{\mathrm{R}} = O\left(S^{4.5} A^{4.5}\right) ,$$

and, for MGs, it is given by Proposition 5.1.

Proposition 5.1. *The runtime T_{G} of computing $\mathfrak{T}^\delta \boldsymbol{v}$ and $\mathfrak{B}^\delta \boldsymbol{v}$ for a Markov game satisfies that*

$$T_{\mathrm{G}} = O\left(S^2 AB + S(A+B)^{1.5}(A)^2 \log(\delta^{-1})\right) , \qquad (9)$$

where, without loss of generality, $A \geq B$.

We are now ready to state the central claim of this section, which proves the correctness and computational complexity of RCPI.

Theorem 5.2. *Suppose that $\gamma > 0$, and $\epsilon > 0$ satisfies that*

$$\epsilon \; > \; \frac{2(1+\gamma)\delta}{(1-\gamma)^2} \; > \; 0.$$

for δ in (7). Then Algorithm 2 returns $(\boldsymbol{\pi}, \boldsymbol{\sigma}) \in \mathfrak{S}_\epsilon(\rho)$ in $O\left(Z\left(T \cdot (1+m) + S^2 AB + S^3\right)\right)$ operations where

$$Z := \left\lceil \frac{\log\left(\frac{1-\gamma}{2\gamma}\epsilon - \frac{3+\gamma}{1-\gamma}\delta\right) - \log(r_{\max} + \delta)}{\log(\gamma)} \right\rceil , \qquad (10)$$

and $T \in \{T_{\mathrm{R}}, T_{\mathrm{G}}\}$ is the complexity of computing $\mathfrak{T}^\delta \boldsymbol{v}$ and $\mathfrak{B}^\delta \boldsymbol{v}$ for RMDP or MG, respectively.

The proof of Theorem 5.2 follows standard contraction arguments and is deferred to the appendix of Badger, Huang, and Petrik (2025). The main argument relies on the following lemma, which bounds the computational time and establishes the contraction property of each iteration of RCPI.

Lemma 5.3. *Each loop of Algorithm 2 (Lines 3–10) runs in*

$$O\left((1+m)\, T + S^2 AB + S^3\right) \qquad (11)$$

operations for $T \in \{T_{\mathrm{R}}, T_{\mathrm{G}}\}$ and $(\boldsymbol{v}^k)_{k \in \mathbb{N}}$ satisfies that

$$\psi_\infty^\delta(\boldsymbol{v}^{k+1}) \; \leq \; \gamma \cdot \psi_\infty^\delta(\boldsymbol{v}^k) + 2 \cdot (1+\gamma) \cdot \delta \, . \qquad (12)$$

Note that the number of Bellman backups required for RCPI to find $(\boldsymbol{\pi}, \boldsymbol{\sigma}) \in \mathfrak{S}_\epsilon(\rho)$ shares the same upper bound as robust VI for both games and robust MDPs if the maximum number of recovery steps m is set to 0. RCPI's main attraction is that it can leverage its exact policy evaluation to aid in finding an optimal solution, while ignoring or correcting it when issues arise. This gives it speeds that are close to, if not faster than, PAI when solving problems in practice. As a result, the worst-case time complexity of RCPI is no worse than value iteration, but it offers significant possible speedup due to the policy evaluation step.

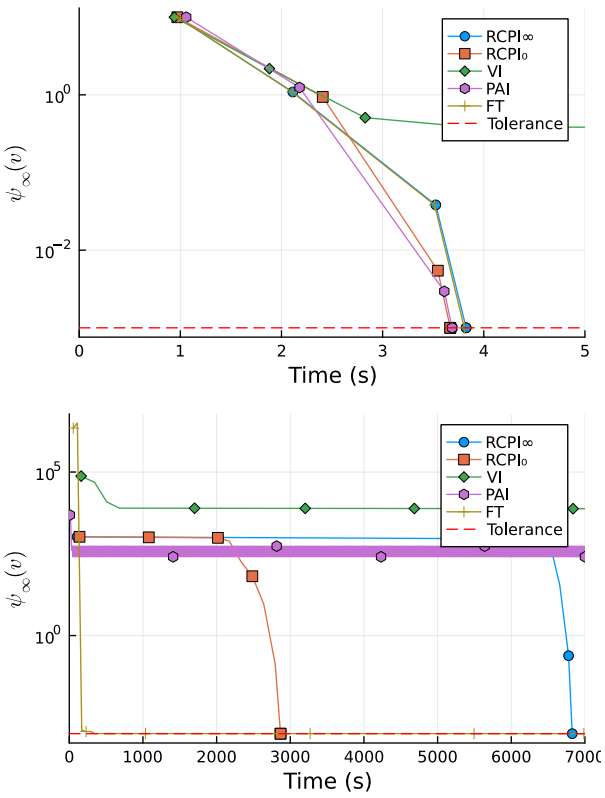

Figure 3: The Bellman residual of each algorithm's value function plotted as a function of time for the *large Markov games (top)* with 200 to 1000 states, and the *large inventory problems (bottom)* with 40 to 200 states.

6 Numerical Results

To evaluate the effectiveness of RCPI a series of examples were solved using a range of algorithms including PAI, Hoffman Karp (HK), Filar Tolwinski's algorithm (FT), robust value iteration (VI), a variation of Hoffman Karp (PPI) (Ho, Petrik, and Wiesemann 2021), a variation on PAI (WS) (Winnicki and Srikant 2023), and our algorithm (RCPI). To simplify comparison, RCPI's hyperparameter m was either set to 0, producing a method that never fixed its evaluation step, called RCPI$_0$, or m was left unbounded, making a method that always fixed its evaluation step, called RCPI$_\infty$. The full source code for the algorithms and domains is available at https://github.com/keithbadger/Fast-Policy-Iteration-for-Markov-Games-and-Robust-MDPs.

For each domain, there was a smaller set of problems solved with γ set to 0.5, 0.75, 0.9, and 0.99, and a larger set of problems where we set γ to 0.9, 0.99, and 0.999. We use the smaller problems in order to evaluate the slower algorithms in a reasonable time. The larger problems can only be solved by the faster methods within our time limits. We allocated a larger time budget to VI to establish a reference.

Examining Table 1, VI is the slowest algorithm tested for games. Every other algorithm achieves a faster median solve

	Markov Games		Inventory		Gambler's Ruin		Gridworld	
Algorithm	Small	Large	Small	Large	Small	Large	Small	Large
RCPI_∞	0.3	2.3	0.1	84.8	4.7	54.3	1.5	23.7
RCPI_0	0.3	2.3	0.2	87.8	5.1	54.4	1.4	23.7
VI	3.4	253.0	2.4	23629.6	8.7	106.6	6.9	145.2
PAI	0.3	2.3	0.1	87.2	4.8	54.3	1.4	23.6
FT	0.3	2.4	0.2	86.9	5.6	77.1	1.4	23.3
HK	0.5	*	0.4	*	14.4	*	5.7	*
WS	1.0	*	0.8	*	5.2	*	3.2	*
PPI	0.6	*	0.4	*	10.6	*	4.9	*

Table 1: The median runtime of each algorithm's in seconds for the small and large problem sets of every domain.

time. The difference in solution time is due to VI exclusively using policy improvement steps to improve its estimated value function, whereas other methods incorporate policy evaluation steps, which are often more efficient.

WS is the closest method to VI conceptually, only adding a fixed number of policy evaluation backups in between policy improvement steps. The policy evaluation backups can significantly reduce the solve time of games, as shown in Table 1, where all of WS's median runtimes are below those of VI.

The remaining methods use exact policy evaluation steps. HK and PPI's evaluations differ from the others, as they both evaluate the primary policy by holding it constant and optimizing the adversary. In contrast, the other methods hold both policies constant and find the stationary value function $\boldsymbol{v}$ which satisfies $\mathfrak{T}\boldsymbol{v} = \boldsymbol{v}$. Although HK and PPI's policy evaluation methods do improve upon VI's solve time, they are still cumbersome when compared to the closed-form methods of PAI, FT, RCPI_0, and RCPI_∞, which evaluate both policies simultaneously. As a result, HK and PPI are the next slowest methods for games.

The median runtimes of the closed-form methods were approximately the same across all domains. RCPI_0 and RCPI_∞ were always within a few seconds of the fastest runtime. The worst-case runtimes of the closed-form methods from Figure 3 were similarly close, with all of their Bellman residual curves indicating a super-linear convergence rate.

Several domains were tested for robust MDPs including gamblers ruin (Kallenberg 2022), gridworld (Sutton and Barto 2018, section 6.5), and inventory management (Puterman 2005, section 3.2). From Table 1, each algorithm maintained the same relative performance from Markov games, except that WS and VI did comparatively better in the gambler's ruin. WS and VI did well because the optimal betting scheme in the non-robust version of gambler's ruin is to bet \$1 if the win rate is greater than 50% and to bet all money otherwise. The gambler repeats this action until reaching the maximum capital, when it obtains the reward. The result is a singular optimal state trajectory following the current state. The reward from obtaining the maximum capital does not affect the current state's policy until there is a Bellman backup for every state in the optimal trajectory following it. This type of domain favors methods with inexpensive evaluations, as only states that reward has been reached via policy improvement will be worth evaluating. Time spent doing exact evaluations for the other states does not provide any benefit.

In Table 1, the closed evaluation methods have the lowest median runtimes for the small and large inventory problem sets. By examining the inventory problems in Figure 3, PAI stops converging around 10^2, where it becomes stuck cycling between suboptimal value function estimates, as described in (Van der Wal 1978). At points, FT's Bellman residual also increases, which comes from using $\psi_2(\boldsymbol{v})^2$ as an objective instead of $\psi_\infty(\boldsymbol{v})$. The larger number of available actions and transitions stemming from those actions makes policy improvement for inventory management more expensive than for the other domains. As a result, methods that evaluate their policies simultaneously benefit more heavily from limiting the number of policy improvement steps needed to converge.

7 Conclusion

Historically, solving Markov games and robust MDPs has involved choosing between a slow method that always converges, or a fast method that may never finish. Attempts have been made to provide a solution with both speed and convergence guarantees, such as the algorithm of Filar and Tolwinski, but such attempts have failed. RCPI is a simple solution which provides the best possible worst-case convergence rate, and empirically performs as fast if not faster than any method proposed before it.

It remains to be seen if there is a way to fix the algorithm of Filar and Tolwinski that keeps the Newtonian interpretation of PAI with $\psi_2(\boldsymbol{v})^2$ as the objective function. Such a solution could provide insight into how to deal with discontinuities caused by the min and max operations more generally. It is not known how to optimize the hyperparameter of RCPI m which could reduce the level of knowledge required to use RCPI effectively.

Acknowledgments

We thank the anonymous reviewers for their detailed reviews and thoughtful comments, which significantly improved the paper's clarity. This work was supported, in part, by NSF grants 2144601 and 2218063 and ONR grant N0001425GI01179.

References

Badger, K.; Huang, J.; and Petrik, M. 2025. Convergence of Fast Policy Iteration in Markov Games and Robust MDPs. arXiv:2508.06661.

Behzadian, B.; Petrik, M.; and Ho, C. P. 2021. Fast Algorithms for L-infinity-constrained s-Rectangular Robust MDPs. In *Neural Information Processing Systems (NeurIPS)*.

Behzadian, B.; Russel, R.; Ho, C. P.; and Petrik, M. 2021. Optimizing Percentile Criterion Using Robust MDPs. In *International Conference on Artificial Intelligence and Statistics (AIStats)*.

Condon, A. 1993. On Algorithms for Simple Stochastic Games. *Advances in Computational Complexity Theory, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 13: 51–71.

Filar, J.; and Vrieze, K. 1997. *Competitive Markov Decision Processes*. Springer.

Filar, J. A.; and Tolwinski, B. 1991. On the Algorithm of Pollatschek and Avi-ltzhak. In Raghavan, T. E. S.; Ferguson, T. S.; Parthasarathy, T.; and Vrieze, O. J., eds., *Stochastic Games and Related Topics: In Honor of Professor L. S. Shapley*, Theory and Decision Library, 59–70. Springer Netherlands.

Grand-Clément, J.; and Petrik, M. 2024. On the Convex Formulations of Robust Markov Decision Processes. *Mathematics of Operations Research*.

Grand-Clément, J.; Petrik, M.; and Vieille, N. 2025. Beyond Discounted Returns: Robust Markov Decision Processes with Average and Blackwell Optimality. arXiv:2312.03618.

Hansen, TD.; Miltersen, PB.; 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. 2021. Partial Policy Iteration for L1-robust Markov Decision Processes. *Journal of Machine Learning Research*, 22: 1–46.

Ho, C. P.; Petrik, M.; and Wiesemann, W. 2022. Robust Phi-Divergence MDPs. In *Neural Information Processing Systems (NeurIPS)*.

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

Kallenberg, L. 2022. Markov Decision Processes.

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

Littman, ML. 1994. Markov Games as a Framework for Multi-Agent Reinforcement Learning. *International Conference on Machine Learning International Conference of Machine Learning (ICML)*.

Lobo, E.; Cousins, C.; Petrik, M.; and Zick, Y. 2023. Percentile Criterion Optimization in Offline Reinforcement Learning. In *Neural Information Processing Systems (NeurIPS)*.

Pérolat, J.; Piot, B.; Geist, M.; Scherrer, B.; and Pietquin, O. 2016. Softened Approximate Policy Iteration for Markov Games. In *International Conference on Machine Learning (ICML)*, 1860–1868. PMLR.

Pollatschek, M. A.; and Avi-Itzhak, B. 1969. Algorithms for Stochastic Games with Geometrical Interpretation. *Management Science*, 15.

Puterman, M. L. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

Shapley, L. S. 1953. Stochastic Games. In *National Academy of the Sciences of the USA*, volume 39.

Shou, Z.; Chen, X.; Fu, Y.; and Di, X. 2022. Multi-Agent Reinforcement Learning for Markov Routing Games: A New Modeling Paradigm for Dynamic Traffic Assignment. *Transportation Research Part C: Emerging Technologies*, 137: 103560.

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

Van der Wal, J. 1978. Discounted Markov games: Generalized policy iteration method. *Journal of Optimization Theory and Applications*, 25(1): 125–138.

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

Williams, R. J. R.; and Baird, L. C. L. 1993. Tight Performance Bounds on Greedy Policies Based on Imperfect Value Functions. In *Yale Workshop on Adaptive and Learning Systems*. Northeastern University.

Winnicki, A.; and Srikant, R. 2023. A New Policy Iteration Algorithm for Reinforcement Learning in Zero-Sum Markov Games. arXiv:2303.09716.

Ye, Y. 2011. 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).

Zhang, R.; Liu, Q.; Wang, H.; Xiong, C.; Li, N.; and Bai, Y. 2022. Policy Optimization for Markov Games: Unified Framework and Faster Convergence. *Advances in Neural Information Processing Systems*, 35: 21886–21899.