Provable Self-Play Algorithms for Competitive Reinforcement Learning

Yu Bai [*1] **Chi Jin** [*2]

Abstract

Self-play, where the algorithm learns by playing against itself without requiring any direct supervision, has become the new weapon in modern Reinforcement Learning (RL) for achieving super-human performance in practice. However, the majority of exisiting theory in reinforcement learning only applies to the setting where the agent plays against a fixed environment; it remains largely open whether self-play algorithms can be provably effective, especially when it is necessary to manage the exploration/exploitation tradeoff. We study self-play in competitive reinforcement learning under the setting of Markov games, a generalization of Markov decision processes to the two-player case. We introduce a self-play algorithm—Value Iteration with Upper/Lower Confidence Bound (VI-ULCB)—and show that it achieves regret $\tilde{\mathcal{O}}(\sqrt{T})$ after playing T steps of the game, where the regret is measured by the agent's performance against a *fully adversarial* opponent who can exploit the agent's strategy at *any* step. We also introduce an explore-then-exploit style algorithm, which achieves a slightly worse regret of $\tilde{\mathcal{O}}(T^{2/3})$, but is guaranteed to run in polynomial time even in the worst case. To the best of our knowledge, our work presents the first line of provably sample-efficient self-play algorithms for competitive reinforcement learning.

1. Introduction

This paper studies *competitive reinforcement learning* (competitive RL), that is, reinforcement learning with two or more agents taking actions simultaneously, but each maximizing their own reward. Competitive RL is a major branch of the more general setting of multi-agent reinforcement learning (MARL), with the specification that the agents have conflicting rewards (so that they essentially compete with each other) yet can be trained in a centralized fashion (i.e. each agent has access to the other agents' policies) (Crandall & Goodrich, 2005).

There are substantial recent progresses in competitive RL, in particular in solving hard multi-player games such as GO (Silver et al., 2017), Starcraft (Vinyals et al., 2019), and Dota 2 (OpenAI, 2018). A key highlight in their approaches is the successful use of *self-play* for achieving super-human performance in absence of human knowledge or expert opponents. These self-play algorithms are able to learn a good policy for all players from scratch through repeatedly playing the current policies against each other and performing policy updates using these self-played game trajectories. The empirical success of self-play has challenged the conventional wisdom that expert opponents are necessary for achieving good performance, and calls for a better theoretical understanding.

In this paper, we take initial steps towards understanding the effectiveness of self-play algorithms in competitive RL from a theoretical perspective. We focus on the special case of two-player zero-sum Markov games (Shapley, 1953; Littman, 1994), a generalization of Markov Decision Processes (MDPs) to the two-player setting. In a Markov game, the two players share states, play actions simultaneously, and observe the same reward. However, one player aims to maximize the return while the other aims to minimize it. This setting covers the majority of two-player games including GO (there is a single reward of $\{+1, -1\}$ at the end of the game indicating which player has won), and also generalizes zero-sum matrix games (von Neumann, 1928)—an important game-theoretic problem—into the multi-step (RL) case.

More concretely, the goal of this paper is to design low-regret algorithms for solving episodic two-player Markov games in the general setting (Kearns & Singh, 2002), that is, the algorithm is allowed to play the game for a fixed amount of episodes using arbitrary policies, and its performance is measured in terms of the regret. We consider a strong notion of regret for two-player zero-sum games, where the performance of the deployed policies in each episode is measured against the best response *for that policy*, which can be different in different episodes. Such a regret bound

[*]Equal contribution [1]Salesforce Research [2]Princeton University. Correspondence to: Yu Bai <yu.bai@salesforce.com>, Chi Jin <chij@princeton.edu>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

measures the algorithm's ability in managing the exploration and exploitation tradeoff against fully adaptive opponents, and can directly translate to other types of guarantees such as the PAC sample complexity bound.

Our contribution This paper introduces the first line of provably sample-efficient self-play algorithms for zero-sum Markov game under no restrictive assumptions. Concretely,

- We introduce the first self-play algorithm with $\tilde{\mathcal{O}}(\sqrt{T})$ regret for zero-sum Markov games. More specifically, it achieves $\tilde{\mathcal{O}}(\sqrt{H^3S^2ABT})$ regret in the general case, where H is the length of the game, S is the number of states, A, B are the number of actions for each player, and T is the total number of steps played. In special case of turn-based games, it achieves $\tilde{\mathcal{O}}(\sqrt{H^3S^2(A+B)T})$ regret with guaranteed polynomial runtime.
- We also introduce an explore-then-exploit style algorithm. It has guaranteed polynomial runtime in the general setting of zero-sum Markov games, with a slightly worse $\tilde{\mathcal{O}}(T^{2/3})$ regret.
- We raise the open question about the optimal dependency of the regret on S, A, B. We provide a lower bound $\Omega(\sqrt{S(A+B)T})$, and show that the lower bound can be achieved in simple case of two-step turn-based games by a mirror descent style algorithm.

Above results are summarized in Table 1.

1.1. Related Work

There is a fast-growing body of work on multi-agent reinforcement learning (MARL). Many of them achieve striking empirical performance, or attack MARL in the cooperative setting, where agents are optimizing for a shared or similar reward. We refer the readers to several recent surveys for these results (see e.g. Buşoniu et al., 2010; Nguyen et al., 2018; OroojlooyJadid & Hajinezhad, 2019; Zhang et al., 2019). In the rest of this section we focus on theoretical results related to competitive RL.

Markov games Markov games (or stochastic games) is proposed as a mathematical model for compeitive RL back in the early 1950s (Shapley, 1953). There is a long line of classical work since then on solving this problem (see e.g. Littman, 1994; 2001; Hu & Wellman, 2003; Hansen et al., 2013). They design algorithms, possibly with runtime guarantees, to find optimal policies in Markov games when both the transition matrix and reward are known, or in the asymptotic setting where number of data goes to infinity. These results do not directly apply to the non-asymptotic setting where the transition and reward are unknown and only a limited amount of data are available for estimating them.

A few recent work tackles self-play algorithms for Markov games in the non-asymptotic setting, working under either structural assumptions about the game or stronger sampling oracles. Wei et al. (2017) propose an upper confidence algorithm for stochastic games and prove that a self-play style algorithm finds ϵ-optimal policies in $\mathrm{poly}(1/\epsilon)$ samples. Jia et al. (2019); Sidford et al. (2019) study turn-based stochastic games—a special case of general Markov games, and propose algorithms with near-optimal sample complexity. However, both lines of work make strong assumptions—on either the structure of Markov games or how we access data—that are not always true in practice. Specifically, Wei et al. (2017) assumes no matter what strategy one agent sticks to, the other agent can always reach all states by playing a certain policy, and Jia et al. (2019); Sidford et al. (2019) assume access to simulators (or generative models) which enable the agent to directly sample transition and reward information for any state-action pair. These assumptions greatly alleviate the challenge in exploration. In contrast, our results apply to general Markov games without further structural assumptions, and our algorithms have built-in mechanisms for solving the challenge in the exploration-exploitation tradeoff.

Finally, we note that classical R-MAX algorithm (Brafman & Tennenholtz, 2002) does not make restrictive assumptions. It also has provable guarantees even when playing against the adversarial opponent in Markov game. However, the theoretical guarantee in Brafman & Tennenholtz (2002) is weaker than the standard regret, and does not directly imply any self-play algorithm with regret bound in our setting (See Section E for more details).

Adversarial MDP Another line of related work focuses on provable algorithms against *adversarial opponents* in MDP. Most work in this line considers the setting with adversarial rewards (see e.g. Zimin & Neu, 2013; Rosenberg & Mansour, 2019; Jin et al., 2019). These results do not direcly imply provable self-play algorithms in our setting, because the adversarial opponent in Markov games can affect both the reward and the transition. There exist a few works that tackle both adversarial transition functions and adversarial rewards (Yu & Mannor, 2009; Cheung et al., 2019; Lykouris et al., 2019). In particular, Lykouris et al. (2019) considers a stochastic problem with C episodes arbitrarily corrupted and obtain $\mathcal{O}(C\sqrt{T} + C^2)$ regret. When applying these results to Markov games with an adversarial opponent, C can be $\Theta(T)$ without further assumptions, which makes the bound vacuous.

Single-agent RL There is an extensive body of research on the sample efficiency of reinforcement learning in the

Table 1. Regret and PAC guarantees of the Algorithms in this paper for zero-sum Markov games.

Settings	Algorithm	Regret	PAC	Runtime
General Markov Game	VI-ULCB (Theorem 2)	$\tilde{\mathcal{O}}(\sqrt{H^3S^2ABT})$	$\tilde{\mathcal{O}}(H^4S^2AB/\epsilon^2)$	PPAD-complete
	VI-explore (Theorem 5)	$\tilde{\mathcal{O}}((H^5S^2ABT^2)^{1/3})$	$\tilde{\mathcal{O}}(H^5S^2AB/\epsilon^2)$	Polynomial
	Mirror Descent ($H=1$) (Rakhlin & Sridharan, 2013)	$\tilde{\mathcal{O}}(\sqrt{S(A+B)T})$	$\tilde{\mathcal{O}}(S(A+B)/\epsilon^2)$	
Turn-Based Markov Game	VI-ULCB (Corollary 4)	$\tilde{\mathcal{O}}(\sqrt{H^3S^2(A+B)T})$	$\tilde{\mathcal{O}}(H^4S^2(A+B)/\epsilon^2)$	
	Mirror Descent ($H=2$) (Theorem 10)	$\tilde{\mathcal{O}}(\sqrt{S(A+B)T})$	$\tilde{\mathcal{O}}(S(A+B)/\epsilon^2)$	
Both	Lower Bound (Corollary 7)	$\Omega(\sqrt{H^2S(A+B)T})$	$\Omega(H^2S(A+B)/\epsilon^2)$	-

single agent setting (see e.g. Jaksch et al., 2010; Osband et al., 2014; Azar et al., 2017; Dann et al., 2017; Strehl et al., 2006; Jin et al., 2018), which are studied under the model of Markov decision process—a special case of Markov games. For the tabular episodic setting with nonstationary dynamics and no simulators, the best regrets achieved by existing model-based and model-free algorithms are $\tilde{\mathcal{O}}(\sqrt{H^2SAT})$ (Azar et al., 2017) and $\tilde{\mathcal{O}}(\sqrt{H^3SAT})$ (Jin et al., 2018), respectively, where S is the number of states, A is the number of actions, H is the length of each episode, and T is the total number of steps played. Both of them (nearly) match the minimax lower bound $\Omega(\sqrt{H^2SAT})$ (Jaksch et al., 2010; Osband & Van Roy, 2016; Jin et al., 2018).

2. Preliminaries

In this paper, we consider zero-sum Markov Games (MG) (Shapley, 1953; Littman, 1994), which also known as stochastic games in the literature. Zero-sum Markov games are generalization of standard Markov Decision Processes (MDP) into the two-player setting, in which the *max-player* seeks to maximize the total return and the *min-player* seeks to minimize the total return.

Formally, we consider tabular episodic zero-sum Markov games of the form $\mathrm{MG}(H, \mathcal{S}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$, where

- H is the number of steps in each episode.
- $\mathcal{S} = \cup_{h\in[H+1]}\mathcal{S}_h$, and $\mathcal{S}_h$ is the set of states at step h, with $\max_{h\in[H+1]} |\mathcal{S}_h| \le S$.
- $\mathcal{A} = \cup_{h\in[H]}\mathcal{A}_h$, and $\mathcal{A}_h$ is the set of actions of the max-player at step h, with $\max_{h\in[H]} |\mathcal{A}_h| \le A$.
- $\mathcal{B} = \cup_{h\in[H]}\mathcal{B}_h$, and $\mathcal{B}_h$ is the set of actions of the min-player at step h, with $\max_{h\in[H]} |\mathcal{B}_h| \le B$.
- $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is a collection of transition matrices, so that $\mathbb{P}_h(\cdot|s, a, b)$ gives the distribution over states if action pair (a, b) is taken for state s at step h.
- $r = \{r_h\}_{h\in[H]}$ is a collection of reward functions, and $r_h \colon \mathcal{S}_h \times \mathcal{A}_h \times \mathcal{B}_h \to [0, 1]$ is the deterministic reward function at step h. Note that we are assuming that rewards are in $[0, 1]$ for normalization. [1]

In each episode of this MG, an initial state s_1 is picked arbitrarily by an adversary. Then, at each step $h \in [H]$, both players observe state $s_h \in \mathcal{S}_h$, pick the action $a_h \in \mathcal{A}_h, b_h \in \mathcal{B}_h$ simultaneously, receive reward $r_h(s_h, a_h, b_h)$, and then transition to the next state $s_{h+1} \sim \mathbb{P}_h(\cdot|s_h, a_h, b_h)$. The episode ends when s_{H+1} is reached.

Policy and value function A policy μ of the max-player is a collection of H functions $\{\mu_h : \mathcal{S} \to \Delta_{\mathcal{A}_h}\}_{h\in[H]}$, where $\Delta_{\mathcal{A}_h}$ is the probability simplex over action set $\mathcal{A}_h$. Similarly, a policy ν of the min-player is a collection of H functions $\{\nu_h : \mathcal{S} \to \Delta_{\mathcal{B}_h}\}_{h\in[H]}$. We use the notation $\mu_h(a|s)$ and $\nu_h(b|s)$ to present the probability of taking action a or b for state s at step h under policy μ or ν respectively. We use $V_h^{\mu,\nu} \colon \mathcal{S}_h \to \mathbb{R}$ to denote the value function at step h under policy μ and ν, so that $V_h^{\mu,\nu}(s)$ gives the expected cumulative rewards received under policy μ and ν, starting from $s_h = s$, until the end of the episode:

$$V_h^{\mu,\nu}(s) := \mathbb{E}_{\mu,\nu}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}, b_{h'})\middle| s_h = s\right].$$

We also define $Q_h^{\mu,\nu} : \mathcal{S}_h \times \mathcal{A}_h \times \mathcal{B}_h \to \mathbb{R}$ to denote Q-value function at step h so that $Q_h^{\mu,\nu}(s, a)$ gives the cumulative rewards received under policy μ and ν, starting from $s_h = s, a_h = a, b_h = b$, till the end of the episode:

$$Q_h^{\mu,\nu}(s, a, b) := \mathbb{E}_{\mu,\nu}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}, b_{h'})\middle| s_h = s, a_h = a, b_h = b\right].$$

[1]While we study deterministic reward functions for notational simplicity, our results generalize to randomized reward functions.

For simplicity, we use notation of operator $\mathbb{P}_h$ so that $[\mathbb{P}_h V](s,a,b) := \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a,b)} V(s')$ for any value function V. By definition of value functions, for all $(s,a,b,h) \in \mathcal{S}_h \times \mathcal{A}_h \times \mathcal{B}_h \times [H]$, we have the Bellman equation

$$Q_h^{\mu,\nu}(s,a,b) = (r_h + \mathbb{P}_h V_{h+1}^{\mu,\nu})(s,a,b), \tag{1}$$

$$V_h^{\mu,\nu}(s) = \sum_{a,b} \mu_h(a|s)\nu_h(b|s) Q_h^{\mu,\nu}(s,a,b). \tag{2}$$

where we define $V_{H+1}^{\mu,\nu}(s) = 0$ for all $s \in \mathcal{S}_{H+1}$

Best response and regret We now define the notion of best response and review some basic properties of it (cf. (Filar & Vrieze, 2012)), which will motivate our definition of the regret in two-player Markov games. For any max-player strategy μ, there exists a *best response* of the min-player, which is a policy $\nu^\dagger(\mu)$ satisfying $V_h^{\mu,\nu^\dagger(\mu)}(s) = \inf_\nu V_h^{\mu,\nu}(s)$ for any (s,h). For simplicity, we denote $V_h^{\mu,\dagger} := V_h^{\mu,\nu^\dagger(\mu)}$. By symmetry, we can define the best response of the max-player $\mu^\dagger(\nu)$, and define $V_h^{\dagger,\nu}$. The value functions $V_h^{\mu,\dagger}$ and $V_h^{\dagger,\nu}$ satisfy the following Bellman optimality equations:

$$V_h^{\mu,\dagger}(s) = \inf_{\nu\in\Delta_{\mathcal{B}_h}} \sum_{a,b} \mu_h(a|s)\nu(b) Q_h^{\mu,\dagger}(s,a,b), \tag{3}$$

$$V_h^{\dagger,\nu}(s) = \sup_{\mu\in\Delta_{\mathcal{A}_h}} \sum_{a,b} \mu(a)\nu_h(b|s) Q_h^{\dagger,\nu}(s,a,b). \tag{4}$$

It is further known that there exist policies $\mu^\star$, $\nu^\star$ that are optimal against the best responses of the opponent:

$$\begin{aligned} V_h^{\mu^\star,\dagger}(s) &= \sup_\mu V_h^{\mu,\dagger}(s), \\ V_h^{\dagger,\nu^\star}(s) &= \inf_\nu V_h^{\dagger,\nu}(s), \end{aligned} \qquad \text{for all } (s,h).$$

It is also known that, for any (s,h), the minimax theorem holds:

$$\sup_\mu \inf_\nu V_h^{\mu,\nu}(s) = V_h^{\mu^\star,\nu^\star}(s) = \inf_\nu \sup_\mu V_h^{\mu,\nu}(s).$$

Therefore, the optimal strategies $(\mu^\star,\nu^\star)$ are also the Nash Equilibrium for the Markov game. Based on this, it is sensible to measure the suboptimality of any pair of policies $(\hat\mu,\hat\nu)$ using the gap between their performance and the performance of the optimal strategy when playing against the best responses respectively, i.e.,

$$\begin{aligned} &\left[V_h^{\dagger,\hat\nu}(s) - \inf_\nu V_h^{\dagger,\nu}(s)\right] + \left[\sup_\mu V_h^{\mu,\dagger}(s) - V_h^{\hat\mu,\dagger}(s)\right] \\ &= V_h^{\dagger,\hat\nu}(s) - V_h^{\hat\mu,\dagger}(s). \end{aligned} \tag{5}$$

We make this formal in the following definition of the regret.

Definition 1 (Regret). *For any algorithm that plays the Markov game for K episodes with (potentially adversarial) starting state s_1^k for each episode $k = 1,2,\ldots,K$, the regret is defined as*

$$\mathrm{Regret}(K) = \sum_{k=1}^K \left[V_1^{\dagger,\nu^k}(s_1^k) - V_1^{\mu^k,\dagger}(s_1^k)\right],$$

where (μ^k, ν^k) denote the policies deployed by the algorithm in the k-th episode.

We note that as a unique feature of self-play algorithms, the learner is playing against herself, and thus chooses strategies for both max-player and min-player at each episode.

2.1. Turn-based games

In zero-sum Markov games, each step involves the two players playing simultaneously and independently. It is a general framework, which contains a very important special case—*turn-based* games. (Shapley, 1953; Jia et al., 2019).

The main feature of a turn-based game is that only one player is taking actions in each step; in other words, the max and min player take turns to play the game. Formally, a turn-based game can be defined through a partition of steps $[H]$ into two sets $\mathcal{H}_{\max}$ and $\mathcal{H}_{\min}$, where $\mathcal{H}_{\max}$ and $\mathcal{H}_{\min}$ denote the sets of steps the max-player and the min-player choose the actions respectively, which satisfies $\mathcal{H}_{\max} \cap \mathcal{H}_{\min} = \emptyset$ and $\mathcal{H}_{\max} \cup \mathcal{H}_{\min} = [H]$. As a special example, GO is a turn-based game in which the two players play in alternate turns, i.e.

$$\mathcal{H}_{\max} = \{1,3,\ldots,H-1\} \quad \text{and} \quad \mathcal{H}_{\min} = \{2,4,\ldots,H\}$$

Mathematically, we can specialize general zero-sum Markov games to turn-based games by restricting $\mathcal{A}_h = \{\mathring{a}\}$ for all $h \in \mathcal{H}_{\min}$, and $\mathcal{B}_h = \{\mathring{b}\}$ for all $h \in \mathcal{H}_{\max}$, where $\mathring{a}$ and $\mathring{b}$ are special dummy actions. Consequently, in those steps, $\mathcal{A}_h$ or $\mathcal{B}_h$ has only a single action as its element, i.e. the corresponding player can not affect the game in those steps. A consequence of this specialization is that the Nash Equilibria for turn-based games are pure strategies (i.e. deterministic policies) (Shapley, 1953), similar as in one-player MDPs. This is not always true for general Markov games.

3. Main Results

In this section, we present our algorithm and main theorems. In particular, our algorithm is the first self-play algorithm that achieves $\tilde{\mathcal{O}}(\sqrt{T})$ regret in Markov Games. We describe the algorithm in Section 3.1, and present its theoretical guarantee for general Markov games in Section 3.2. In Section 3.3, we show that when specialized to turn-based games, the regret and runtime of our algorithm can be further improved.

3.1. Algorithm description

To solve zero-sum Markov games, the main idea is to extend the celebrated UCB (Upper Confidence Bounds) principle—an algorithmic principle that achieves provably efficient exploration in bandits (Auer et al., 2002) and single-agent RL (Azar et al., 2017; Jin et al., 2018)—to the two-player setting. Recall that in single-agent RL, the provably efficient `UCBVI` algorithm (Azar et al., 2017) proceeds as

> **Algorithm** (`UCBVI` for single-player RL): Compute $\{Q_h^{\mathsf{up}}(s,a) : h,s,a\}$ based on estimated transition and optimistic (upper) estimate of reward, then play one episode with the greedy policy with respect to Q^{up}.

Regret bounds for `UCBVI` is then established by showing and utilizing the fact that Q^{up} remains an optimistic (upper) estimate of the optimal $Q^\star$ throughout execution of the algorithm.

In zero-sum games, the two player have *conflicting* goals: the max-player seeks to maximize the return and the min-player seeks to minimize the return. Therefore, it seems natural here to maintain two sets of Q estimates, one upper bounding the true value and one lower bounding the true value, so that each player can play optimistically with respect to her own goal. We summarize this idea into the following proposal.

> **Proposal** (Naive two-player extension of `UCBVI`): Compute $\left\{Q_h^{\mathsf{up}}(s,a,b), Q_h^{\mathsf{low}}(s,a,b)\right\}$ based on estimated transition and {upper, lower} estimates of rewards, then play one episode where the max-player (μ) is greedy with respect to Q^{up} and the min-player (ν) is greedy with respect to Q^{low}.

However, the above proposal is not yet a well-defined algorithm: a greedy strategy μ with respect to Q^{up} requires the knowledge of how the other player chooses b, and vice versa. Therefore, what we really want is not that "μ is greedy with respect to Q^{up}", but rather that "μ is greedy with respect to Q^{up} when the other player uses ν", and vice versa. In other words, we rather desire that (μ,ν) are *jointly greedy* with respect to $(Q^{\mathsf{up}}, Q^{\mathsf{low}})$.

Our algorithm concretizes such joint greediness precisely, building on insights from one-step matrix games: we choose (μ_h,ν_h) to be the Nash equilibrium for the *general-sum game* in which the payoff matrix for the max player is Q^{up} and for the min player is Q^{low}. In other words, both player have their own payoff matrix (and they are not equal), but they jointly determine their policies. Formally, we let (μ,ν) be determined as

$$\begin{aligned}&(\mu_h(\cdot|s), \nu_h(\cdot|s))\\ &= \textsc{Nash\_General\_Sum}(Q_h^{\mathsf{up}}(s,\cdot,\cdot), Q_h^{\mathsf{low}}(s,\cdot,\cdot))\end{aligned}$$

for all (h,s), where NASH_GENERAL_SUM is a subroutine that takes two matrices $\mathbf{P},\mathbf{Q}\in\mathbb{R}^{A\times B}$, and returns the Nash equilibrium $(\phi,\psi)\in\Delta_A\times\Delta_B$ for general sum game, which satisfies

$$\phi^\top \mathbf{P}\psi = \max_{\tilde{\phi}} \tilde{\phi}^\top \mathbf{P}\psi, \quad \phi^\top \mathbf{Q}\psi = \min_{\tilde{\psi}} \phi^\top \mathbf{Q}\tilde{\psi}. \tag{6}$$

Such an equilibrium is guaranteed to exist due to the seminal work of Nash (1951), and is computable by algorithms such as the Lemke-Howson algorithm (Lemke & Howson, 1964). With the NASH_GENERAL_SUM subroutine in hand, our algorithm can be briefly described as

> **Our Algorithm** (VI-ULCB): Compute $\left\{Q_h^{\mathsf{up}}(s,a,b), Q_h^{\mathsf{low}}(s,a,b)\right\}$ based on estimated transition and {upper, lower} estimates of rewards, along the way determining policies (μ,ν) by running the NASH_GENERAL_SUM subroutine on $(Q^{\mathsf{up}}, Q^{\mathsf{low}})$. Play one episode according to (μ,ν).

The full algorithm is described in Algorithm 1.

3.2. Guarantees for General Markov Games

We are now ready to present our main theorem.

Theorem 2 (Regret bound for VI-ULCB). *For zero-sum Markov games, Algorithm 1 (with choice of bonus $\beta_t = c\sqrt{H^2 S\iota/t}$ for large absolute constant c) achieves regret*

$$\begin{aligned}\mathrm{Regret}(K) &\le \mathcal{O}\left(\sqrt{H^3S^2\left[\max_{h\in[H]} A_hB_h\right]T\iota}\right)\\ &\le \mathcal{O}\left(\sqrt{H^3S^2ABT\iota}\right)\end{aligned}$$

with probability at least $1-p$, where $\iota = \log(SABT/p)$.

We defer the proof of Theorem 2 into Appendix A.1.

Optimism in the face of uncertainty and best response An implication of Theorem 2 is that a low regret can be achieved via *self-play*, i.e. the algorithm plays with itself and does not need an expert as its opponent. This is intriguing because the regret is measured in terms of the suboptimality

Algorithm 1 Value Iteration with Upper-Lower Confidence Bound (VI-ULCB)

```
1: Initialize: for any $(s,a,b,h)$, $Q_h^{\mathsf{up}}(s,a,b) \leftarrow H$, $Q_h^{\mathsf{low}}(s,a,b) \leftarrow 0$, $N_h(s,a,b) \leftarrow 0$, $N_h(s,a,b,s') \leftarrow 0$.
2: for episode $k = 1, \dots, K$ do
3:     for step $h = H, H-1, \dots, 1$ do
4:         for $(s,a,b) \in \mathcal{S}_h \times \mathcal{A}_h \times \mathcal{B}_h$ do
5:             $t = N_h(s,a,b)$;
6:             $Q_h^{\mathsf{up}}(s,a,b) \leftarrow \min\{\hat{r}_h(s,a,b) + [\widehat{\mathbb{P}}_h V_{h+1}^{\mathsf{up}}](s,a,b) + \beta_t, H\}$
7:             $Q_h^{\mathsf{low}}(s,a,b) \leftarrow \max\{\hat{r}_h(s,a,b) + [\widehat{\mathbb{P}}_h V_{h+1}^{\mathsf{low}}](s,a,b) - \beta_t, 0\}$
8:         end for
9:         for $s \in \mathcal{S}_h$ do
10:            $(\mu_h(\cdot|s), \nu_h(\cdot|s)) \leftarrow$ NASH_GENERAL_SUM$(Q_h^{\mathsf{up}}(s,\cdot,\cdot), Q_h^{\mathsf{low}}(s,\cdot,\cdot))$
11:            $V_h^{\mathsf{up}}(s) \leftarrow \sum_{a,b} \mu_h(a|s)\nu_h(b|s) Q_h^{\mathsf{up}}(s,a,b)$.
12:            $V_h^{\mathsf{low}}(s) \leftarrow \sum_{a,b} \mu_h(a|s)\nu_h(b|s) Q_h^{\mathsf{low}}(s,a,b)$.
13:        end for
14:    end for
15:    Receive $s_1$.
16:    for step $h = 1, \dots, H$ do
17:        Take action $a_h \sim \mu_h(s_h)$, $b_h \sim \nu_h(s_h)$.
18:        Observe reward $r_h$ and next state $s_{h+1}$.
19:        $N_h(s_h,a_h,b_h) \leftarrow N_h(s_h,a_h,b_h) + 1$.
20:        $N_h(s_h,a_h,b_h,s_{h+1}) \leftarrow N_h(s_h,a_h,b_h,s_{h+1}) + 1$
21:        $\hat{\mathbb{P}}_h(\cdot|s_h,a_h,b_h) \leftarrow \frac{N_h(s_h,a_h,b_h,\cdot)}{N_h(s_h,a_h,b_h)}$.
22:        $\hat{r}_h(s_h,a_h,b_h) \leftarrow r_h$.
23:    end for
24: end for
```

against the worst-case opponent:

$$\mathrm{Regret}(K) = \sum_{k=1}^{K} \left[V_1^{\dagger,\nu^k}(s_1^k) - V_1^{\mu^k,\dagger}(s_1^k) \right]$$

$$= \sum_{k=1}^{K} \underbrace{\left[\max_{\mu} V_1^{\mu,\nu^k}(s_1^k) - V_1^{\mu^k,\nu^k}(s_1^k) \right]}_{\text{gap between } \mu^k \text{ and the best response to } \nu^k} + \underbrace{\left[V_1^{\mu^k,\nu^k}(s_1^k) - \min_{\nu} V_1^{\mu^k,\nu}(s_1^k) \right]}_{\text{gap between } \nu^k \text{ and the best response to } \mu^k}.$$

(Note that this decomposition of the regret has a slightly different form from (5).) Therefore, Theorem 2 demonstrates that self-play can protect against fully adversarial opponent even when such a strong opponent is not explicitly available.

The key technical reason enabling such a guarantee is that our Q estimates are optimistic in the face of both the uncertainty of the game, as well as the best response from the opponent. More precisely, we show that the $(Q^{\mathsf{up}}, Q^{\mathsf{low}})$ in Algorithm 1 satisfy with high probability

$$Q_h^{\mathsf{up},k}(s,a,b) \ge \sup_{\mu} Q_h^{\mu,\nu^k}(s,a,b)$$
$$\ge \inf_{\nu} Q_h^{\mu^k,\nu}(s,a,b) \ge Q_h^{\mathsf{low},k}(s,a,b)$$

for all (s,a,b,h,k), where $(Q^{\mathsf{up},k}, Q^{\mathsf{low},k})$ denote the running $(Q^{\mathsf{up}}, Q^{\mathsf{low}})$ at the beginning of the k-th episode (Lemma 1). In constrast, such a guarantee (and consequently the regret bound) is not achievable if the upper and lower estimates are only guaranteed to {upper, lower} bound the *values* of the Nash equilibrium.

Translation to PAC bound Our regret bound directly implies a PAC sample complexity bound for learning near-equilibrium policies, based on an online-to-batch conversion. We state this in the following Corollary, and defer the proof to Appendix A.2.

Corollary 3 (PAC bound for VI-ULCB). *Suppose the initial state of Markov game is fixed at s_1, then there exists a pair of (randomized) policies $(\widehat{\mu}, \widehat{\nu})$ derived through the VI-ULCB algorithm such that with probability at least $1-p$ (over the randomness in the trajectories) we have*

$$\mathbb{E}_{\widehat{\mu},\widehat{\nu}}\left[V^{\dagger,\widehat{\nu}}(s_1) - V^{\widehat{\mu},\dagger}(s_1) \right] \le \epsilon,$$

as soon as the number of episodes $K \ge \Omega(H^4 S^2 AB\iota/\epsilon^2)$, where $\iota = \log(HSAB/(p\epsilon))$, and the expectation is over the randomization in $(\widehat{\mu}, \widehat{\nu})$.

Runtime of Algorithm 1 Algorithm 1 involves the NASH_GENERAL_SUM subroutine for computing the Nash equilibrium of a general sum matrix game. However, it is known that the computational complexity for approximating[2] such an equilibrium is PPAD-complete (Daskalakis, 2013), a complexity class conjectured to not enjoy polynomial or quasi-polynomial time algorithms. Therefore, Algorithm 1 is strictly speaking not a polynomial time algorithm, despite of being rather sample-efficient.

We note however that there exists practical implementations of the subroutine such as the Lemke-Howson algorithm (Lemke & Howson, 1964) that can usually find the solution efficiently. We will further revisit the computational issue in Section 4, in which we design a computationally efficient algorithm for zero-sum games with a slightly worse $\tilde{O}(T^{2/3})$ regret.

[2]More precisely, our proof requires the subroutine to find a $(1+1/H)$-multiplicative approximation of the equilibrium, that is, for payoff matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{A\times B}$ we desire vectors $\phi \in \Delta_A$ and $\psi \in \Delta_B$ such that $\max_{\tilde{\phi}} \tilde{\phi}^\top \mathbf{P}\psi - \min_{\tilde{\psi}} \phi^\top \mathbf{Q}\tilde{\psi} \le (1 + 1/H)\phi^\top(\mathbf{P}-\mathbf{Q})\psi$.

3.3. Guarantees for Turn-based Markov Games

We now instantiate Theorem 2 on turn-based games (introduced in Section 2.1), in which the same algorithm enjoys better regret guarantee and polynomial runtime. Recall that in turn-based games, for all h, we have either $A_h = 1$ or $B_h = 1$, therefore given $\max_h A_h \le A$ and $\max_h B_h \le B$ we have

$$\max_h A_h B_h \le \max\{A, B\} \le A + B,$$

and thus by Theorem 2 the regret of Algorithm 1 on turn-based games is bounded by $\mathcal{O}(\sqrt{H^3 S^2 (A+B) T})$.

Further, since either $A_h = 1$ or $B_h = 1$, all the NASH_GENERAL_SUM subroutines reduce to *vector* games rather than matrix games, and can be trivially implemented in polynomial (indeed linear) time. Indeed, suppose the payoff matrices in (6) has dimensions $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{A\times 1}$, then NASH_GENERAL_SUM reduces to finding $\phi \in \Delta_A$ and $\psi \equiv 1$ such that

$$\phi^\top \mathbf{P} = \max_{\tilde{\phi}} \tilde{\phi}^\top \mathbf{P}$$

(the other side is trivialized as $\psi \in \Delta_1$ has only one choice), which is solved at $\phi = e_{a^\star}$ where $a^\star = \arg\max_{a\in[A]} \mathbf{P}_a$. The situation is similar if $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{1\times B}$.

We summarize the above results into the following corollary.

Corollary 4 (Regret bound for VI-ULCB on turn-based games)**.** *For turn-based zero-sum Markov games, Algorithm 1 has runtime* $\mathrm{poly}(S, A, B, T)$ *and achieves regret bound* $\mathcal{O}(\sqrt{H^3 S^2 (A+B) T\iota})$ *with probability at least* $1-p$*, where* $\iota = \log(SABT/p)$.

4. Computationally Efficient Algorithm

In this section, we show that the computational issue of Algorithm 1 is not intrinsic to the problem: there exists a sublinear regret algorithm for general zero-sum Markov games that has a guaranteed polynomial runtime, with regret scaling as $\mathcal{O}(T^{2/3})$, slightly worse than that of Algorithm 1. Therefore, computational efficiency can be achieved if one is willing to trade some statistical efficiency (sample complexity). For simplicity, we assume in this section that the initial state s_1 is fixed.

Value Iteration after Exploration At a high level, our algorithm follows an explore-then-exploit approach. We begin by running a (polynomial time) reward-free exploration procedure REWARD_FREE_EXPLORATION(ϵ) on a small number of episodes, which queries the MDP and outputs an estimate $(\hat{\mathbb{P}}, \hat{r})$. Then, we run value iteration on the empirical version of Markov game with transition $\hat{\mathbb{P}}$ and reward $\hat{r}$, which finds its Nash equilibrium $(\hat{\mu}, \hat{\nu})$. Finally, the algorithm simply plays the policy $(\hat{\mu}, \hat{\nu})$ for the remaining episodes. The full algorithm is described in Algorithm 2 in the Appendix.

Algorithm 2 Value Iteration after Exploration (VI-Explore)

```
1: $(\hat{\mathbb{P}}, \hat{r}) \leftarrow$ REWARD_FREE_EXPLORATION($\epsilon$).
2: $V_H(s) \leftarrow 0$ for any $s \in \mathcal{S}_H$.
3: for step $h = H-1, \ldots, 1$ do
4:     for $(s, a, b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ do
5:         $Q_h(s,a,b) \leftarrow \hat{r}_h(s,a,b) + [\hat{\mathbb{P}}_h V_{h+1}](s,a,b)$.
6:     end for
7:     for $s \in \mathcal{S}$ do
8:         $(\hat{\mu}_h(\cdot|s), \hat{\nu}_h(\cdot|s)) \leftarrow$
               NASH_ZERO_SUM($Q_h(s,\cdot,\cdot)$).
9:     end for
10: end for
11: for all remaining episodes do
12:     Play the game with policy $(\hat{\mu}, \hat{\nu})$.
13: end for
```

By "reward-free" exploration, we mean the procedure will not use any reward information to guide exploration. Instead, the procedure prioritize on visiting all possible states and gathering sufficient information about their transition and rewards, so that $(\hat{\mathbb{P}}, \hat{r})$ are close to $(\mathbb{P}, r)$ in the sense that the Nash equilibria of $\mathrm{MG}(\hat{\mathbb{P}}, \hat{r})$ and $\mathrm{MG}(\mathbb{P}, r)$ are close, where $\mathrm{MG}(\hat{\mathbb{P}}, \hat{r})$ denotes the Markov game with transition $\hat{\mathbb{P}}$ and reward $\hat{r}$.

This goal can be achieved by the following algorithm. For any fixed state s, we can create an artificial reward $\tilde{r}$ defined as $\tilde{r}(s,a,b) = 1$ and $\tilde{r}(s',a,b) = 0$ for any $s' \neq s$, a and b. Then, we can treat $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ as a new action set for a single agent, and run any standard reinforcement learning algorithm with PAC or regret guarantees to find a near-optimal policy $\tilde{\pi}$ of $\mathrm{MDP}(H, \mathcal{S}, \mathcal{C}, \mathbb{P}, \tilde{r})$. It can be shown that the optimal policy for this MDP is the policy that maximize the probability to reach state s. Therefore, by repeatedly playing $\tilde{\pi}$, we can gather transition and reward information at state s as well as we can. Finally, we repeat the routine above for all state s. See Appendix B for more details.

In this paper, we adapt the sharp treatments in Jin et al. (2020) which studies reward-free exploration in the single-agent MDP setting, and provide following guarantees for the REWARD_FREE_EXPLORATION procedure.

Theorem 5 (PAC bound for VI-Explore)**.** *With probability at least* $1-p$, REWARD_FREE_EXPLORATION(ϵ) *runs for* $c(H^5 S^2 AB\iota/\epsilon^2 + H^7 S^4 AB\iota^3/\epsilon)$ *episodes with some large constant* c*, and* $\iota = \log(HSAB/(p\epsilon))$*, and outputs* $(\hat{\mathbb{P}}, \hat{r})$ *such that the Nash equilibrium* $(\hat{\mu}, \hat{\nu})$ *of* $\mathrm{MG}(\hat{\mathbb{P}}, \hat{r})$ *satisfies*

$$\left[V^{\dagger,\hat{\nu}}(s_1) - V^{\hat{\mu},\dagger}(s_1)\right] \le \epsilon.$$

Importantly, such Nash equilibrium $(\hat{\mu}, \hat{\nu})$ of $\mathrm{MG}(\hat{\mathbb{P}}, \hat{r})$ can be computed by Value Iteration (VI) using $\hat{\mathbb{P}}$ and $\hat{r}$. VI only calls NASH_ZERO_SUM subroutine, which takes a matrix $\mathbf{Q} \in \mathbb{R}^{A \times B}$ and returns the Nash equilibrium $(\phi, \psi) \in \Delta_A \times \Delta_B$ for zero-sum game, which satisfies

$$\max_{\tilde{\phi}} \tilde{\phi}^\top \mathbf{Q}\psi = \phi^\top \mathbf{Q}\psi = \min_{\tilde{\psi}} \phi^\top \mathbf{Q}\tilde{\psi}. \tag{7}$$

This problem can by solved efficiently (in polynomial time) by many existing algorithms designed for convex-concave optimization (see, e.g. (Koller, 1994)), and does not suffer from the PPAD-completeness that NASH_GENERAL_SUM does.

The PAC bound in Theorem 5 can be easily converted into a regret bound, which is presented as follows.

Corollary 6 (Polynomial time algorithm via explore-then–exploit)**.** *For zero-sum Markov games, with probability at least $1 - p$, Algorithm 2 runs in $\mathrm{poly}(S, A, B, H, T)$ time, and achieves regret bound*

$$\mathcal{O}\Big((H^5 S^2 ABT^2 \iota)^{\frac{1}{3}} + \sqrt{H^7 S^4 ABT \iota^3}\Big),$$

where $\iota = \log(SABT/p)$.

5. Towards the Optimal Regret

We investigate the tightness of our regret upper bounds in Theorem 2 and Corollary 4 through raising the question of optimal regret in two-player Markov games, and making initial progresses on it by providing lower bounds and new upper bounds in specific settings. Specifically, we ask an

Open question: What is the optimal regret for general Markov games (in terms of dependence on (H, S, A, B))?

It is known that the (tight) regret lower bound for single-player MDPs is $\Omega(\sqrt{SAT \cdot \mathrm{poly}(H)})$ (Azar et al., 2017). By restricting two-player games to a single-player MDP (making the other player dummy), we immediately have

Corollary 7 (Regret lower bound, corollary of Jaksch et al. (2010), Theorem 5)**.** *The regret*[3] *for any algorithm on turn-based games (and thus also general zero-sum games) is lower bounded by $\Omega(\sqrt{H^2 S(A+B)T})$.*

Comparing this lower bound with the upper bound in Theorem 2 ($\tilde{\mathcal{O}}(\sqrt{S^2 ABT \cdot \mathrm{poly}(H)})$ regret for general games and $\tilde{\mathcal{O}}(\sqrt{S^2(A+B)T \cdot \mathrm{poly}(H)})$ regret for turn-based games), there are gaps in both the S-dependence and the (A, B)-dependence.

Matching the lower bound on short-horizon games Towards closing the gap between lower and upper bounds, we develop alternative algorithms in the special case where *each player only plays once*, i.e. one-step general games with $H = 1$ and two-step turn-based games. In these cases, we show that there exists mirror descent type algorithms that achieve an improved regret $\tilde{\mathcal{O}}(\sqrt{S(A+B)T})$ (and thus matching the lower bounds), *provided that we consider a weaker notion of the regret* defined as

Definition 8 (Weak Regret)**.** *The* weak regret *for any algorithm that deploys policies (μ^k, ν^k) in episode k is defined as*

$$\begin{aligned} &\mathrm{WeakRegret}(K) \\ &:= \max_{\mu} \sum_{k=1}^{K} V^{\mu, \nu^k}(s_1^k) - \min_{\nu} \sum_{k=1}^{K} V^{\mu^k, \nu}(s_1^k). \end{aligned} \tag{8}$$

The difference in the weak regret is that it uses *fixed* opponents—as opposed to adaptive opponents—for measuring the performance gap: the max is taken with respect to a fixed μ for all episodes $k = 1, \ldots, K$, rather than a different μ for each episode. By definition, we have for any algorithm that $\mathrm{WeakRegret}(K) \le \mathrm{Regret}(K)$.

With the definition of the weak regret in hand, we now present our results for one-step games. Their proofs can be found in Appendix C.

Theorem 9 (Weak regret for one-step simultaneous game, adapted from Rakhlin & Sridharan (2013))**.** *For one-step simultaneous games ($H = 1$), there exists a mirror descent type algorithm that achieves weak regret bound $\mathrm{WeakRegret}(T) \le \tilde{\mathcal{O}}(\sqrt{S(A+B)T})$ with high probability.*

Theorem 10 (Weak regret for two-step turn-based game)**.** *For one-step turn-based games ($H = 2$), there exists a mirror descent type algorithm that achieves weak regret bound $\mathrm{WeakRegret}(T) \le \tilde{\mathcal{O}}(\sqrt{S(A+B)T})$ with high probability.*

Proof insights; bottleneck in multi-step case The improved regret bounds in Theorem 9 and 10 are possible due to availability of *unbiased estimates of counterfactual Q values*, which in turn can be used in mirror descent type algorithms with guarantees. Such unbiased estimates are only achievable in one-step games as the two policies are "not intertwined" in a certain sense. In contrast, in multi-step games (where each player plays more than once), such unbiased estimates of counterfactual Q values are no longer available, and it is unclear how to construct a mirror descent algorithm there. We believe it would be an important open question to close the gap in multi-step games (as well as the gap between regret and weak regret) for a further understanding of exploration in games.

[3]This also applies to the weak regret defined in (8).

6. Conclusion

In this paper, we studied the sample complexity of finding the equilibrium policy in the setting of competitive reinforcement learning, i.e. zero-sum Markov games with two players. We designed a self-play algorithm for zero-sum games and showed that it can efficiently find the Nash equilibrium policy in the exploration setting through establishing a regret bound. Our algorithm—Value Iteration with Upper and Lower Confidence Bounds—builds on a principled extension of UCB/optimism into the two-player case by constructing upper and lower bounds on the value functions and iteratively solving general sum subgames.

Towards investigating the optimal runtime and sample complexity in two-player games, we provided accompanying results showing that (1) the computational efficiency of our algorithm can be improved by explore-then-exploit type algorithms, which has a slightly worse regret; (2) the state and action space dependence in the regret can be reduced in the special case of one-step games via alternative mirror descent type algorithms.

We believe this paper opens up many interesting directions for future work. For example, can we design a computationally efficient algorithms that achieves $\tilde{\mathcal{O}}(\sqrt{T})$ regret? What are the optimal dependence of the regret on (S, A, B) in multi-step games? Also, the present results only work in tabular games, and it would be of interest to investigate if similar results can hold in presence of function approximation.

Acknowledgments

We thank Sham Kakade and Haipeng Luo for valuable discussions on the related work. We also thank the Simons Institute at Berkeley and its Foundations of Deep Learning program in Summer 2019 for hosting the authors and incubating our initial discussions.

References

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

Brafman, R. I. and Tennenholtz, M. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct): 213–231, 2002.

Buşoniu, L., Babuvska, R., and De Schutter, B. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*, pp. 183–221. Springer, 2010.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. Reinforcement learning under drift. *arXiv preprint arXiv:1906.02922*, 2019.

Crandall, J. W. and Goodrich, M. A. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proceedings of the 22nd international conference on Machine learning*, pp. 161–168, 2005.

Dann, C., Lattimore, T., and Brunskill, E. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 5713–5723, 2017.

Daskalakis, C. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 9(3):23, 2013.

Filar, J. and Vrieze, K. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

Hansen, T. D., Miltersen, P. B., and Zwick, U. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

Hu, J. and Wellman, M. P. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Jia, Z., Yang, L. F., and Wang, M. Feature-based q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*, 2019.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4868–4878, 2018.

Jin, C., Jin, T., Luo, H., Sra, S., and Yu, T. Learning adversarial markov decision processes with bandit feedback and unknown transition. *arXiv preprint arXiv:1912.01192*, 2019.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. *arXiv preprint arXiv:2002.02794*, 2020.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

Koller, D. Fast algorithms for finding randomized strategies in game trees* daphne koller† daphne@ cs. berkeley. edu. *Computing*, 750:759, 1994.

Lemke, C. E. and Howson, Jr, J. T. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics*, 12(2):413–423, 1964.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Littman, M. L. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pp. 322–328, 2001.

Lykouris, T., Simchowitz, M., Slivkins, A., and Sun, W. Corruption robust exploration in episodic reinforcement learning. *arXiv preprint arXiv:1911.08689*, 2019.

Nash, J. Non-cooperative games. *Annals of mathematics*, pp. 286–295, 1951.

Nguyen, T. T., Nguyen, N. D., and Nahavandi, S. Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications. *arXiv preprint arXiv:1812.11794*, 2018.

OpenAI. Openai five. `https://blog.openai.com/openai-five/`, 2018.

OroojlooyJadid, A. and Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963*, 2019.

Osband, I. and Van Roy, B. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

Rakhlin, S. and Sridharan, K. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pp. 3066–3074, 2013.

Rosenberg, A. and Mansour, Y. Online convex optimization in adversarial markov decision processes. *arXiv preprint arXiv:1905.07773*, 2019.

Shapley, L. S. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

Sidford, A., Wang, M., Yang, L. F., and Ye, Y. Solving discounted stochastic two-player games with near-optimal time and sample complexity. *arXiv preprint arXiv:1908.11071*, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. PAC model-free reinforcement learning. In *International Conference on Machine Learning*, pp. 881–888, 2006.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

von Neumann, J. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Wei, C.-Y., Hong, Y.-T., and Lu, C.-J. Online reinforcement learning in stochastic games. In *Advances in Neural Information Processing Systems*, pp. 4987–4997, 2017.

Yu, J. Y. and Mannor, S. Arbitrarily modulated markov decision processes. In *Proceedings of the 48h IEEE Conference on Decision and Control*, pp. 2946–2953, 2009.

Zhang, K., Yang, Z., and Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Zimin, A. and Neu, G. Online learning in episodic markovian decision processes by relative entropy policy search. In *Advances in neural information processing systems*, pp. 1583–1591, 2013.