STOCHASTIC NO-REGRET LEARNING FOR GENERAL GAMES WITH VARIANCE REDUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that a stochastic version of optimistic mirror descent (OMD), a variant of mirror descent with recency bias, converges fast in general games. More specifically, with our algorithm, the individual regret of each player vanishes at a speed of $O(1/T^{3/4})$ and the sum of all players' regret vanishes at a speed of $O(1/T)$, which is an improvement upon the $O(1/\sqrt{T})$ convergence rate of prior stochastic algorithms, where T is the number of interaction rounds.

Due to the advantage of stochastic methods in the computational cost, we significantly improve the time complexity over the deterministic algorithms to approximate coarse correlated equilibrium. To achieve lower time complexity, we equip the stochastic version of OMD in (AM21) with a novel low-variance Monte-Carlo estimator. Our algorithm extends previous works (AM21; CJST19) from two-player zero-sum games to general games.

1 INTRODUCTION

How does a player in a game interact with others, and selfishly maximize its own utilities? This is one central problem in online learning and game theory and has intimate connections to economics, auction design, and machine learning. The study of this problem was pioneered by (Bro49; Rob51). Robinson (Rob51) shows that fictitious play asymptotically converges to Nash equilibrium in two-player zero-sum games. But its convergence rate is exponentially slow and it may not even converge in non-zero-sum games (Sha64).

Another natural choice for each player is to use no-regret learning algorithms. With some well-known families of no-regret learning algorithms, e.g., mirror descent (NY83) and follow-the-regularized-leader (KV05), the average regret of each player vanishes at a speed of $O(1/\sqrt{T})$ where T is the number of interaction rounds. This regret bound implies an $O(1/\sqrt{T})$ convergence rate to the coarse correlated equilibrium in general games (or Nash equilibrium in two-player zero-sum games). And it is noteworthy that Chen and Peng (CP20) show that these algorithms' convergence rate is $\Omega(1/\sqrt{T})$.

Players can do even better with some special no-regret algorithms tailored for games. The most representative one is known as optimistic mirror descent (OMD) which is a variant of mirror descent with recency bias. Syrgkanis et al. (SALS15) and Rakhlin et al. (RS13) show that OMD approaches optimal social welfare (or equivalently, minimizes the sum of all players' regrets) at a speed of $O(1/T)$ and minimizes each player's individual regret at a speed of $O(1/T^{3/4})$. Several works (CP20; HAM21; DFG21) improve the results in (SALS15) under different settings or assumptions. Remarkably, Daskalakis et al. (DFG21) improve the convergence rate of players' individual regret of OMD to $O(\text{poly}\log T/T)$ in general games.

However, the computational cost of players to use OMD, as well as other deterministic no-regret algorithms, could be not manageable. This is because each player needs to compute the exact loss vector to update its strategy in OMD. And the time complexity of computing this exact loss vector is, in the worst case, exponential in the number of players in the game. One standard method to accelerate the computation is to estimate the loss vector with Monte-Carlo methods. But a Monte-Carlo estimator with uncontrolled variance will immediately make the convergence rate degenerate to $O(1/\sqrt{T})$.

To alleviate the effect of Monte-Carlo estimator's variance, Carmon et al. (CJST19) and Alacaoglu et al. (AM21) propose variance reduced stochastic no-regret algorithms with convergence rate of $O(1/T)$ for two-player zero-sum games. As a result, they improve the time complexity of computing ϵ-Nash Equilibrium in two-player zero-sum games from $O(\text{Cost}/\epsilon)$ of deterministic algorithms to $O(\text{Cost} + \sqrt{\text{Cost}}/\epsilon)$ (some lower order terms are omitted) where Cost is the time complexity of computing the loss vector.

While Carmon et al. (CJST19) and Alacaoglu et al. (AM21) make a huge step towards developing efficient stochastic algorithms for games, their algorithms are tailored for the simplest two-player zero-sum games and could not cover more practical settings, such as auction, which may involve multiple players and not be zero-sum. One crucial factor of the algorithms in Carmon et al. (CJST19) and Alacaoglu et al. (AM21) is a stochastic loss estimator with small variance. However, the time complexity of calculating this estimator is $O(A^{N-1})$ where N is the number of players, which is exponentially large in general games. The high complexity of the estimator becomes a major obstacle to develop efficient stochastic algorithms for general games.

Contributions. We consider general normal-form games with an arbitrary number of players. Compared to the two-player zero-sum case, this is more challenging and practically significant. We show that in general games, a stochastic version of OMD converges to the optimal social welfare (or equivalently, minimizes the sum of all players' regrets) at a rate of $\tilde{O}(1/T)$ and minimizes the individual regret at a speed of $O(1/T^{3/4})$ in contrast to the $O(1/\sqrt{T})$ convergence rate of existing stochastic algorithms. Due to the advantage of stochastic methods in the computational cost, this significantly improves the time-complexity to approximate coarse correlated equilibrium in general games. Please see Table 1 for the comparison of the time complexity of our algorithm against prior works.

To achieve the above regret bounds, we make two main technical contributions. Firstly, we extend the theoretical framework of analyzing regret bounds of stochastic OMD in (AM21) from two-player zero-sum games to general games. Secondly, we propose a novel low-variance Monte-Carlo estimator for general games. The computational complexity of this estimator is exponentially faster than Carmon et al. (CJST19) and Alacaoglu et al. (AM21) while the variance is only slightly larger. The stochastic OMD algorithm equipped with our novel estimator achieves the above results.

The rest of the paper is organized as follows: In Section 2, we discuss prior works related to this problem. In Section 3, we provide necessary preliminaries for games, coarse correlated equilibrium and optimistic mirror descent. In Section 4, we introduce our algorithm and present a general regret upper bound in Theorem 1. In Section 5, we introduce our low-variance Monte-Carlo estimator and analyze its variance in Lemma 3. In Section 6, by combining the results in Theorem 1 and Lemma 3, we present our final regret bounds in Theorem 2 and 3, as well as the time complexity to approximate coarse correlated equilibrium in Corollary 1 and 2.

2 Related Work

Comparisons to existing algorithms. Table 1 compares the time complexity of our algorithm to compute ϵ-coarse correlated equilibrium for general games (and ϵ-Nash equilibrium for two-player zero-sum games) against prior no-regret algorithms. The time complexity is determined by two terms: the convergence rate (or the regret) and the computational cost in each round.

Deterministic algorithms (PSS21; DFG21; SALS15) converge fast, but with a relatively higher per round time complexity since they have to compute the loss in each round. Stochastic algorithms exploit the Monte-Carlo approach to accelerate the computation of loss. However, the variance of the estimated loss may slow down the convergence rate. To alleviate the effect of the variance, Carmon et al. (CJST19) and Alacaoglu et al. (AM21) develop variance reduced stochastic no-regret learning algorithms. Their algorithms significantly accelerate the computation of ϵ-Nash equilibrium in two-player zero-sum games.

Variance reduction. Variance reduction is one of the most useful techniques to accelerate stochastic algorithm (see (GSBR20) for a comprehensive survey). Typically, when optimizing the finite sum problem $\min_x F(x) = \sum_{i=1}^N F_i(x)$, instead of estimating the gradient by $\nabla F_i(x)$ like Stochastic Gradient Descent (SGD), the variance reduction method proposes to use $\tilde{\nabla} F(x) = \nabla F(w^k) +$

	ϵ-Nash equilibrium Two-player zero-sum	Weak ϵ-CCE General games	Strong ϵ-CCE General games
(D)(SALS15)	$\tilde{O}(\text{Cost}/\epsilon)$	$\tilde{O}(N^3\text{Cost}/\epsilon)$	$\tilde{O}(N^{3/2}\text{Cost}/\epsilon^{4/3})$
(D)(DFG21)	$\tilde{O}(\text{Cost}/\epsilon)$	$\tilde{O}(N^3\text{Cost}/\epsilon)$	$\tilde{O}(N^2\text{Cost}/\epsilon)$
(D)(PSS21)	$\tilde{O}(\text{Cost}/\epsilon)$	$\tilde{O}(N^3\text{Cost}/\epsilon)$	$\tilde{O}(N^2\text{Cost}/\epsilon)$
(S)(AM21)	$\tilde{O}(\text{Cost}+\sqrt{A\text{Cost}}/\epsilon)$	-	-
(S)(CJST19)	$\tilde{O}(\text{Cost}+\sqrt{A\text{Cost}}/\epsilon)$	-	-
(S) Ours	$\tilde{O}(\text{Cost}+\sqrt{A\text{Cost}}/\epsilon)$	$\tilde{O}(\text{Cost}+N^{\frac{7}{2}}\sqrt{A\text{Cost}}/\epsilon)$	$O(\text{Cost}+\text{Cost}^{\frac{2}{3}}\frac{N^{\frac{7}{3}}A^{\frac{2}{3}}}{\epsilon^{4/3}})$

Table 1: Comparisons of time complexity to compute ϵ-CCE for general games and ϵ-Nash equilibrium for two-player zero-sum games. Cost denotes the time complexity of computing the loss vector. N is the number of players and A is the number of actions for each player. (S) means the algorithm is stochastic and (D) means the algorithm is deterministic. For stochastic algorithms, the time complexity is the expected running time to reach ϵ-equilibrium. $\tilde{O}$ hides factors which are polynomial in terms of $\log(1/\epsilon)$ and $\log(A)$. For stochastic algorithms, the time complexity is the expected running time of achieving an expected approximation error, which directly follows (AM21).

$\nabla F_i(x) - \nabla F_i(w^k)$, where w^k is called the "snapshot". In many cases, e.g., when F_i's are convex, sampling i from a uniform distribution is sufficient to reduce the variance. When dealing with games, it turns out that one has to be meticulous to design a low-variance sampling distribution, even when the game is two-player zero-sum (CJST19; AM21). And it remains unclear how to design such a low-variance distribution for general games. One of our main contributions is a low-variance Monte-Carlo estimator for general games, which ensures the fast convergence in general games.

Stochastic no-regret learning for large-scale sequential decision-making. Counterfactual regret minimization (CFR) (ZJBP07) generalizes no regret learning to games with sequential decision-making, namely extensive-form games. A long list of works study the variants of CFR (LWZB09; BS19a; ZRL+19; MSB+17; TBJB15). And these works lead to the superhuman AI, Libratus (BS18) and Pluribus (BS19b), for two-player and six-player heads-up no-limit Texas Hold'em (HUNL). Since the state space of HUNL is super huge, both Libratus and Pluribus adopt the Monte-Carlo CFR with external sampling (LWZB09). And the experimental results suggest that the variance of Monte-Carlo sampling significantly slows down the convergence rate. Several works empirically reduce the variance of Monte-Carlo sampling (SBL+19; BS19b). However, to the best of our knowledge, these algorithms do not enjoy any theoretical guarantees on the convergence rate. We hope our algorithms could provide insights into the developing of low-variance stochastic algorithms for large-scale games.

3 PRELIMINARY

Notation. For any vector $v \in \mathbb{R}^d$, denote $v(j)$ as its jth coordinate, $\|v\|_1 = \sum_{j=1}^d |v(j)|$ as its ℓ_1-norm, $\|v\|_2 = \sqrt{\sum_{j=1}^d v^2(j)}$ as its ℓ_2-norm, and $\|v\|_\infty = \max_{j=1,\cdots,d} |v(j)|$ as its ℓ_∞ norm. For a general norm $\|\cdot\|$, let $\|\cdot\|_*$ represent its dual norm. Denote $\langle v, w\rangle = \sum_{i=1}^d v(i)w(i)$ as the standard inner product of two vectors $v, w \in \mathbb{R}^d$. For a positive integer n, let $[n] = \{1, \cdots, n\}$. For a discrete set S, let $\Delta(S)$ be the set of distributions over S.

Basics of game. We consider general games with N players. The action space of each player $i \in [N]$ is $\mathcal{A}_i$. Denote $A = \max_i |\mathcal{A}_i|$ as the cardinality of the largest action space. The joint action space of all players is $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$. For simplicity, let $\mathcal{A}_{-i} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \cdots \times \mathcal{A}_N$ be the joint action space of all players except for player $i \in [N]$. The loss of players can be specified by the functions $F_1, F_2, \cdots, F_N : \mathcal{A} \to [0, 1]$, which map the joint action space to a real value.

Specifically, if each player j selects action $a_j \in \mathcal{A}_j$, then $F_i(a)$ is the loss of player i where $a := (a_1, a_2, \ldots, a_N)$.

A mixed strategy σ_i is a probability distribution over $\mathcal{A}_i$. We say a player p_i plays according to σ_i if it selects action $a_i \in \mathcal{A}_i$ with probability $\sigma_i(a_i)$. For any strategy profile $\sigma := (\sigma_i)_{i\in[N]}$, let $\sigma_{-i} := (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_N)$ denote the strategy profile σ after removing σ_i. And for convenience, let (σ_i', σ_{-i}) denote the strategy profile σ after replacing σ_i with σ_i'. Similarly, given an action profile $a := (a_i)_{i\in[N]}$, let $a_{-i} := (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N)$ denote the action profile after removing a_i and (a_i', a_{-i}) be the action profile a after replacing a_i with a_i'.

With a little abuse of notation, for a strategy profile σ, let $F_i(\sigma) := \mathbb{E}_{a\sim\sigma}\left[F_i(a)\right]$ be the expected loss of player i if each player $j \in [N]$ plays according to σ_j. For convenience, define the vector $F_i(a_{-i}) \in [0,1]^{|\mathcal{A}_i|}$ with $[F_i(a_{-i})](a_i') = F_i((a_i', a_{-i}))$ representing the loss of player i when each player $j \neq i$ selects a_j and i selects $a_i' \in \mathcal{A}_i$. Similarly, define the vector $F_i(\sigma_{-i}) \in [0,1]^{|\mathcal{A}_i|}$ with $[F_i(\sigma_{-i})](a_i') = \mathbb{E}_{a_{-i}\sim\sigma_{-i}}\left[F_i((a_i', a_{-i}))\right]$ representing the expected loss of player i if each player j plays according to σ_j and i selects action a_i'. Further, denote Cost as the time complexity of computing the vector $F_i(\sigma_{-i})$.

No-regret learning. In online learning, N players play the game for T rounds. In round $k \in [T]$, player i plays according to the strategy σ_i^k and suffers the expected loss $\langle F_i(\sigma_{-i}^k), \sigma_i^k\rangle$. Each player aims to minimize its cumulative loss, which is equivalent to minimizing its regret $\max_{\sigma_i} R_i(\sigma_i) := \max_{\sigma_i} \sum_{k=1}^T \langle F_i(\sigma_{-i}^k), \sigma_i^k - \sigma_i\rangle$ where $R_i(\sigma_i)$ is the cumulative loss difference between the adopted strategies and the fixed strategy σ_i.

Given the strategy at round $1, ..., k$, the optimistic mirror descent (OMD) (RS13; SALS15) method calculates the strategy in round $k+1$ as:

$$\sigma_i^{k+1} = \operatorname*{arg\,min}_{\sigma_i\in\Delta(\mathcal{A}_i)} D(\sigma_i, \sigma_i^k - \tau(2F_i(\sigma_{-i}^k) - F_i(\sigma_{-i}^{k-1}))) \,. \tag{1}$$

where τ is the step size and $D(x, y)$ is the Bregman divergence induced by some mirror map $h(\cdot)$ [1]. Without loss of generality, we mainly consider two common mirror maps in this paper: negative entropy $h_1(x) = \sum_{a=1}^d x(a)\log x(a)$ and squared ℓ_2-norm $h_2(x) = \sum_{a=1}^d x^2(a)$. Their corresponding Bregman divergences are $D_1(x, y) = \sum_{a=1}^d x(a)\log(x(a)/y(a))$ and $D_2(x, y) = \sum_{a=1}^d (x(a) - y(a))^2$, respectively. Our analyses apply to other mirror maps as well.

Coarse correlated equilibrium (CCE). A correlated strategy ζ is a distribution over the joint action space $\mathcal{A}$. We call ζ a coarse correlated equilibrium (CCE) if no player can benefit from unilaterally deviating from ζ given that others take actions according to ζ. There are two versions of approximate CCE: one corresponds to the social welfare and is referred to as weak ϵ-CCE; the other corresponds to the individual loss and is referred to as strong ϵ-CCE. Intuitively, weak ϵ-CCE states that the averaged difference among all players $i \in [N]$ between the expected loss $\mathbb{E}_{a\sim\zeta}\left[F_i(a)\right]$ by following ζ and the least expected loss $\min_{a_i^*} \mathbb{E}_{a\sim\zeta}\left[F_i((a_i^*, a_{-i}))\right]$ by deviating from ζ is no more than ϵ. And strong ϵ-CCE requires that the maximum difference among all players $i \in [N]$ between these two types of expected loss to be smaller than ϵ. The formal definitions of these two versions of CCE are as follows.

Definition 1. *A correlated strategy ζ is a weak ϵ-CCE (corresponding to social welfare) if*

$$\sum_{i=1}^N \left(\mathbb{E}_{a\sim\zeta}\left[F_i(a)\right] - \min_{a_i^*} \mathbb{E}_{a\sim\zeta}\left[F_i((a_i^*, a_{-i}))\right]\right) \leq \epsilon$$

and it is a strong ϵ-CCE (corresponding to individual loss) if

$$\max_{i\in[N]} \left(\mathbb{E}_{a\sim\zeta}\left[F_i(a)\right] - \min_{a_i^*} \mathbb{E}_{a\sim\zeta}\left[F_i((a_i^*, a_{-i}))\right]\right) \leq \epsilon \,.$$

[1] $h : \mathbb{R}^d \to \mathbb{R}$ is called a mirror map if h is strongly convex with respect to some norm and $\nabla h(\mathbb{R}^d) = \mathbb{R}^d$.

Here for convenience, we simply multiply the average difference by N as the sum over all players in the definition of Weak ϵ-CCE. Thus a strong ϵ-CCE is a a weak $N\epsilon$-CCE. Since this work mainly focuses on the stochastic method, we study the expected performance of the generated strategy. Specifically, a randomly generated correlated strategy ζ is called a weak ϵ-CCE if $\sum_{i=1}^{N}\left(\mathbb{E}_{\zeta,a\sim\zeta}\left[F_i(a)\right]-\min_{a_i^*}\mathbb{E}_{\zeta,a\sim\zeta}\left[F_i((a_i^*,a_{-i}))\right]\right)\leq\epsilon$ and a strong ϵ-CCE if $\max_{i\in[N]}\left(\mathbb{E}_{\zeta,a\sim\zeta}\left[F_i(a)\right]-\min_{a_i^*}\mathbb{E}_{\zeta,a\sim\zeta}\left[F_i((a_i^*,a_{-i}))\right]\right)\leq\epsilon$, where the expectation is additionally taken over the randomness of the generated strategy ζ.

Connection between no-regret learning and CCE. The following Lemma 1 shows a well known connection between no-regret learning and CCE.

Lemma 1. *Let $\sigma^1,\cdots,\sigma^T$ denote an arbitrary collection of strategy profiles and $\zeta(a):=\frac{1}{T}\sum_{k=1}^{T}\prod_{i=1}^{N}\sigma_i^k(a_i)$. It holds that*

$$\mathbb{E}_{a\sim\zeta}\left[F_i(a)\right]-\min_{a_i^*}\mathbb{E}_{a\sim\zeta}\left[F_i((a_i^*,a_{-i}))\right]\leq\max_{\sigma_i}R_i(\sigma_i)/T,\forall i\in[N]\,. \tag{2}$$

Lemma 1 shows that the running time to approximate CCE depends on both the regret and the running time in each round. Deterministic algorithms, such as OMD, though enjoy excellent regret bounds, take long running time for computation in each round. In next sections, we will show how to use Monte-Carlo method to accelerate the calculations without significantly hurting the regret bound.

4 Optimistic Mirror Descent with Monte-Carlo Estimation

In this section, we present our algorithm. We first introduce some necessary notations. Let $F_{i,a_{-i}}(\sigma_{-i})=F_i(a_{-i})\frac{\sigma_{-i}(a_{-i})}{q(a_{-i})}$ denote an unbiased estimator of $F_i(\sigma_{-i})$ where $a_{-i}\in\mathcal{A}_{-i}$ denote a random variable drawn from a sampling distribution q_{-i}.

The algorithm is presented in Algorithm 1, which is a stochastic version of OMD shown in equation 1 with variance reduction. There are two crucial modifications over the vanilla OMD to accelerate calculation: one is on the estimated loss and the other is on the update starting points.

Algorithm 1 Optimistic mirror descent with variance reduction

1: **Input:** hyper-parameters p,τ and α; mirror map h
2: **Initialize:** $w_i^1(a)=1/A$ and $\sigma_i^1(a)=1/A$
3: **for** $k=1,2,\cdots,T$ **do**
4: Sample u from uniform distribution over $[0,1]$.
5: **for** $i=1,\cdots,N$ **do**
6: Compute $\hat{\sigma}_i^k$ such that $\nabla h(\hat{\sigma}_i^k)=\alpha\nabla h(\sigma_i^k)+(1-\alpha)\nabla h(w_i^k)$.
7: Sample $a_{-i}^k\sim\mathrm{LVE}(i,\sigma^k,w^{k-1})$.
8: Compute $\tilde{\sigma}_i^{k+1}$ such that

$$\nabla h(\tilde{\sigma}_i^{k+1})=\nabla h(\hat{\sigma}_i^k)-\tau(F_i(w_{-i}^k)+F_{i,a_{-i}^k}(\sigma_{-i}^k)-F_{i,a_{-i}^k}(w_{-i}^{k-1})).$$

9: Compute $\sigma_i^{k+1}=\arg\min_{\sigma_i\in\Delta(\mathcal{A}_i)}D(\sigma_i,\tilde{\sigma}_i^{k+1})$.
10: Update w_i^{k+1} as

$$w_i^{k+1}=\begin{cases}\sigma_i^{k+1}, & u<p;\\ w_i^k, & u\geq p.\end{cases} \tag{3}$$

11: **end for**
12: **end for**

Estimated loss. As can be seen in Line 8, Algorithm 1 updates the strategy using the estimated loss $F_i(w_{-i}^k)+F_{i,a_{-i}^k}(\sigma_{-i}^k)-F_{i,a_{-i}^k}(w_{-i}^{k-1})$ instead of $2F_i(\sigma_{-i}^k)-F_i(\sigma_{-i}^{k-1})$ in the vanilla OMD equation 1. By constructing appropriate distribution q_{-i} to sample a_{-i}^k (Line 7), it can be shown that

the adopted estimated loss $F_{i,a^k_{-i}}(\sigma^k_{-i}) - F_{i,a^k_{-i}}(w^{k-1}_{-i})$ where $F_{i,a_{-i}}(\sigma_{-i}) = F_i(a_{-i})\frac{\sigma_{-i}(a_{-i})}{q(a_{-i})}$ and $a_{-i} \sim q_{-i}$ in the algorithm is an unbiased estimator of $F_i(\sigma^k_{-i}) - F_i(w^{k-1}_{-i})$ in the vanilla OMD sampled from the distribution q_{-i}. The objective to introduce such estimated loss is to reduce the computation complexity. Since w^{k+1} is only updated with probability p at each round (Line 10), the running time to compute $F_i(w^k_{-i})$ over T rounds is just $O(pT\text{Cost})$. It is also worth noting that such estimator may bring high variance. So how to construct q_{-i} to accelerate convergence while ensuring low variance is the key innovation of our algorithm. We postpone the descriptions and discussions on this estimator to Section 5.

Different update starting points. As shown in Line 8, Algorithm 1 updates σ^{k+1}_i from the starting point $\hat{\sigma}^k_i$ instead of σ^k_i in the vanilla OMD equation 1. This modification was first introduced by (AM21) for two-player zero-sum games. We prove its effect in general games is to admit a faster learning rate in the case of maximizing social welfare. The detailed proof for this part can be seen in the full analysis.

It is remarkable that under special settings of hyper-parameters, our Algorithm 1 degenerates to standard OMD (SALS15) and stochastic OMD for two-player zero-sum games (AM21). Specifically, when setting $p = 1, \alpha = 1$ and using the exact value of $F_i(\sigma^k_{-i}) - F_i(w^{k-1}_{-i})$ in Line 8, Algorithm 1 is equivalent to standard OMD (SALS15). And when the underlying game is two-player zero-sum, our constructed unbiased estimator for $F_i(\sigma^k_{-i}) - F_i(w^{k-1}_{-i})$ is equivalent to that in (AM21) and thus Algorithm 1 can also recover their stochastic OMD.

In the following, we show how to compute σ^{k+1} by taking two common examples of the mirror map function: negative entropy $h_1(x) = \sum_{a=1}^d x(a)\log x(a)$ and square of ℓ_2-norm $h_2(x) = \sum_{a=1}^d x^2(a)$. When the mirror map is h_1, it is easy to verify that $\nabla_{x(a)}h_1(x) = 1 + \log x(a)$. Then according to Line 6, $\hat{\sigma}^k_i(a) = (\sigma^k_i(a))^\alpha (w^k_i(a))^{1-\alpha}$. So we can update σ^{k+1}_i in $O(A)$ time as

$$\sigma^{k+1}_i(a) \propto (\sigma^k_i(a))^\alpha (w^k_i(a))^{1-\alpha} \exp\left(-\tau[F_i(w^k_{-i}) + F_{i,a^k_{-i}}(\sigma^k_{-i}) - F_{i,a^k_{-i}}(w^{k-1}_{-i})](a)\right) .$$

Similarly, when the mirror map is h_2, we can directly compute $\hat{\sigma}^k_i(a) = \alpha\sigma^k_i(a) + (1-\alpha)w^k_i(a)$. The computation of $\tilde{\sigma}^{k+1}_i$ is easy according to the gradient of the ℓ_2-norm and σ^{k+1}_i can then be obtained by standard procedure of projecting $\tilde{\sigma}^{k+1}_i$ to the simplex.

4.1 Theoretical Analysis

We first provide a general regret upper bound for Algorithm 1 in the following Theorem 1. The final results (Theorem 2 and Theorem 3) will be postponed in later sections by combining this general upper bound and the variance upper bound for the unbiased estimator in Lemma 3 as well as some delicate derivations.

Theorem 1. *If $D(x,y) \geq \gamma\|x-y\|^2$ for all x, y and some norm $\|\cdot\|$ with $\gamma > 0$ being a constant, then the expected regret of Algorithm 1 is upper-bounded by*

$$\tau \max_{\sigma_i} \mathbb{E}\left[R_i(\sigma_i)\right] \leq U_i - (1-\alpha)\mathbb{E}\left[\sum_{k=1}^T D\left(\sigma^k_i, w^{k-1}_i\right)\right] + \tau^2\left(1 + \frac{1}{\alpha\gamma}\right)\mathbb{E}\left[\sum_{k=1}^T \|F_{i,a^k_{-i}}(\sigma^k_{-i}) - F_{i,a^k_{-i}}(w^{k-1}_{-i})\|_*^2\right] , \tag{4}$$

where $U_i := \max_{\sigma_i}\mathbb{E}\left[\phi^1_i(\sigma_i) - \phi^{T+1}_i(\sigma_i)\right] + \max_{\sigma_i} D(\sigma_i, \sigma^0_i)$ and

$$\phi^k_i(\sigma_i) = \alpha D(\sigma_i, \sigma^k_i) + (1-\alpha)/p \cdot D(\sigma_i, w^k_i) + (1-\alpha)D(\sigma^k_i, w^{k-1}_i) + \tau\langle F_i(\sigma^k_{-i}) - F_i(w^{k-1}_{-i}), \sigma_i - \sigma^k_i\rangle . \tag{5}$$

Due to the space limit, the full proof of Theorem 1 is deferred to Appendix A.2. To obtain the main order of the above general upper bound, we first bound U_i, i.e., the first term in equation 4.

According to the following Lemma 2, it is easy to see that U_i is of order $\tilde{O}\left(1+\tau+(1-\alpha)/p\right)$. When adopting hyper-parameters such that $\alpha \geq 1-p$ and $\tau = O(1)$, we would have $U_i = \tilde{O}(1)$. Therefore, the key to bound the regret in Theorem 1 is to control the variance of the estimator, i.e., the third term in equation 4, which will be completed by our low-variance estimator introduced in Section 5.

Lemma 2 (Bounds for ϕ_i^k defined in equation 5)**.** *For any k and σ_i, $\phi_i^k(\sigma_i) \geq -4\tau$. If the mirror map is h_1 and $w_i^k(a) = 1/A$, then $\phi_i^k(\sigma_i) \leq 4\tau + (1+(1-\alpha)/p)\log A$. If the mirror map is h_2 and $w_i^k(a) = 1/A$, then $\phi_i^k(\sigma_i) \leq 4\tau + 2(1+(1-\alpha)/p)$.*

5 Low-Variance Monte-Carlo Estimator

In this section, we present our low-variance estimator for $F_i(\sigma_{-i}) - F_i(w_{-i})$ where σ and w are two strategy profiles. For convenience, we assume $\sigma \neq w$. It is standard to use importance sampling to construct an unbiased estimator. Specifically, let q_{-i} denote a distribution over $\mathcal{A}_{-i}$ such that $q_{-i}(a_{-i}) = 0$ only if $\sigma_{-i}(a_{-i}) = 0$ and $w_{-i}(a_{-i}) = 0$. Recall $F_{i,a_{-i}}(\sigma_{-i}) = F_i(a_{-i})\frac{\sigma_{-i}(a_{-i})}{q(a_{-i})}$. Clearly, we have

$$\mathbb{E}_{a_{-i}\sim q_{-i}}\left[F_{i,a_{-i}}(\sigma_{-i})\right] = \sum_{a_{-i}} F_i(a_{-i})\sigma_{-i}(a_{-i}) = F_i(\sigma_{-i})\,.$$

Therefore, $F_{i,a_{-i}}(\sigma_{-i}) - F_{i,a_{-i}}(w_{-i})$ is an unbiased estimator for $F_i(\sigma_{-i}) - F_i(w_{-i})$ when $a_{-i} \sim q_{-i}$. However, for an arbitrary q_{-i}, the variance of the estimator can be very large. For example, let q_{-i} denote the uniform distribution over $\mathcal{A}_{-i}$. Then, in the worst case, the variance of $F_{i,a_{-i}}(\sigma_{-i}) - F_{i,a_{-i}}(w_{-i})$ can be A^N. So we need to carefully design q_{-i} to ensure low variance.

Note the variance $\mathbb{E}_{a_{-i}\sim q_{-i}}\left[\|F_{i,a_{-i}}(\sigma_{-i}) - F_{i,a_{-i}}(w_{-i})\|_\infty^2\right]$ [2] is upper bounded by $\sum_{a_{-i}} \|F_i(a_{-i})\|_\infty^2(\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i}))^2/q_{-i}(a_{-i})$. Intuitively, to control the variance, we should allocate a large probability mass to a_{-i} where $(\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i}))^2$ is large. With the observation that the difference between $\sigma_{-i}(a_{-i})$ and $w_{-i}(a_{-i})$ can be decomposed as $\sigma(a) - w(a) = \sum_{i=1}^N(\sigma_i(a_i) - w_i(a_i))\prod_{x<i}\sigma_x(a_x)\prod_{y>i}w_y(a_y)$ (see Lemma 7 for more details), we propose to sample a_{-i} according to the following distribution:

$$q_{-i}(a_{-i}) = \frac{1}{\sum_{j'\neq i} Z_{j'}}\sum_{j\neq i}|\sigma_j(a_j) - w_j(a_j)|\prod_{x<j,x\neq i}\sigma_x(a_x)\prod_{y>j,y\neq i}w_y(a_y)\,, \tag{6}$$

where $Z_j = \sum_{a_j}|\sigma_j(a_j) - w_j(a_j)|$. It is easy to verify that $q_{-i}(a_{-i}) \geq 0$ and $\sum_{a_{-i}} q_{-i}(a_{-i}) = 1$.

The following Algorithm 2 summarizes an efficient sampling procedure from q_{-i} in equation 6 with polynomial time complexity $O((N-1)A)$. It takes two main steps to sample a_{-i}. Specifically, the first step is to sample index j with probability $Z_j/\sum_{j'\neq i} Z_{j'}$ (Line 1). And the second step is to sample a_{-i} with probability proportional to $|\sigma_j(a_j) - w_j(a_j)|\prod_{x<j,x\neq i}\sigma_x(a_x)\prod_{y>j,y\neq i}w_y(a_y)$ (Line 3-5).

Algorithm 2 LVE(i, σ, w)

1: Compute $Z_j = \sum_{a_j}|\sigma_j(a_j) - w_j(a_j)|$ for $j \neq i$.
2: Sample j with probability $\frac{Z_j}{\sum_{j'\neq i} Z_{j'}}$.
3: For $1 \leq j' < j, j' \neq i$, sample $a_{j'}$ according to $\sigma_{j'}$.
4: Sample a_j with probability $\frac{|\sigma_j(a_j) - w_j(a_j)|}{\sum_{a'_j}|\sigma_j(a'_j) - w_j(a'_j)|}$.
5: For $j < j' \leq N, j' \neq i$, sample $a_{j'}$ according to $w_{j'}$.
6: **return** : a_{-i}

[2] In Theorem 1, we consider the general dual norm $\|\cdot\|_*$. Here, we only upper bound $\|\cdot\|_\infty$ because when the mirror map is h_1 or h_2, the dual norm is $\|\cdot\|_\infty$ or $\|\cdot\|_2 \leq \sqrt{A}\|\cdot\|_\infty$, respectively.

The following Lemma 3 provides an upper bound for the variance of the constructed estimator in equation 6. We believe this result is also of independent interest since it is quite general and may be used to control the variance of other stochastic algorithms for general games.

Lemma 3 (Variance bound)**.** *Sampling $a_{-i} \sim q_{-i}$ defined in equation 6 can be implemented by Algorithm 2. Moreover, we can upper bound its variance by*

$$\mathbb{E}_{a_{-i}\sim q_{-i}}\left[\|F_{i,a_{-i}}(\sigma_{-i}) - F_{i,a_{-i}}(w_{-i})\|_\infty^2\right] \leq (N-1)\sum_{j\neq i}\|\sigma_j - w_j\|_1^2 . \tag{7}$$

Proof. According to the definition of q_{-i}, the variance of the estimated loss is

$$\mathbb{E}_{a_{-i}\sim q_{-i}}\left[\|F_{i,a_{-i}}(\sigma_{-i}) - F_{i,a_{-i}}(w_{-i})\|_\infty^2\right] \leq \sum_{a_{-i}}\left|(\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i}))\right|^2 / q_{-i}(a_{-i}) .$$

Further, recall that the difference between $\sigma(a)$ and $w(a)$ can be decomposed as $\sum_{i=1}^N(\sigma_i(a_i) - w_i(a_i))\prod_{x<i}\sigma_x(a_x)\prod_{y>i}w_y(a_y)$ in Lemma 7 (Please see appendix for the statement and proof), it holds that

$$\begin{aligned}
&\sum_{a_{-i}}\left|\left(\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i})\right)\right|^2 / q_{-i}(a_{-i}) \\
=&\sum_{a_{-i}}\left|\left(\sum_{j\neq i}(\sigma_j(a_j) - w_j(a_j))\prod_{x<j,x\neq i}\sigma_x(a_x)\prod_{y>j,y\neq i}w_y(a_y)\right)\right|^2 / q_{-i}(a_{-i}) \\
\leq&\sum_{a_{-i}}\left|\left(\sum_{i=1}^N|\sigma_i(a_i) - w_i(a_i)|\prod_{x<j,x\neq i}\sigma_x(a_x)\prod_{y>j,y\neq i}w_y(a_y)\right)\right|^2 / q_{-i}(a_{-i}) \\
=&\left(\sum_{j\neq i}Z_j\right)\sum_{a_{-i}}\left|\left(\sum_{i=1}^N|\sigma_i(a_i) - w_i(a_i)|\prod_{x<j,x\neq i}\sigma_x(a_x)\prod_{y>j,y\neq i}w_y(a_y)\right)\right| \\
=&\left(\sum_{j\neq i}Z_j\right)^2 \leq (N-1)\sum_{j\neq i}\|\sigma_j - w_j\|_1^2 .
\end{aligned}$$

□

Comparison to the estimator from (CJST19): (CJST19) uses $q_{-i}^{\text{CLST19}}(a_{-i}) \propto |\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i})|$ as the unbiased estimator. This estimator actually achieves the smallest variance of $(\sum_{a_{-i}}|\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i})|)^2$. However, the challenge comes from the computational cost. In order to sample from q^{CLST19}, we have to access every configurations of a_{-i} to compute $|\sigma_{-i}(a_{-i}) - w_{-i}(a_{-i})|$. Therefore, the time complexity of sampling from q^{CLST19} is $O(|A|^{N-1})$. In the case of two-player zero-sum games ($N = 2$), this time complexity is small. But when the number N of players increases, sampling from q^{CLST19} becomes impractical.

6 Regret Upper Bounds and Time Complexity to Approximate CCE

Now we are ready to provide our final guarantees to reach weak ϵ-CCE which corresponds to the social welfare and strong ϵ-CCE which corresponds to the individual loss.

6.1 Social welfare

We first consider the time complexity to reach a weak ϵ-CCE that guarantees social welfare. For this case, we take the mirror map $h_1(x) = \sum_{a=1}^d x(a)\log x(a)$ as an example. Recall that Lemma 1 shows that the time complexity to approximate CCE depends on both the regret and the per round running time. So we first provide an upper bound for the regret $\sum_{i\in N}\max_{\sigma_i}\mathbb{E}\left[R_i(\sigma_i)\right]$ defined in weak ε-CCE.

Theorem 2. *Let hyper-parameters* $\alpha = 1 - p$, $\tau = \sqrt{\gamma\alpha(1-\alpha)/2}/\big((N-1)\sqrt{1+\alpha\gamma}\big)$, $\gamma \in (0, 1/2)$ *be a constant and the mirror map be* h_1. *Then there exists a constant* C *such that* $\max_\sigma \mathbb{E}\left[\sum_{i=1}^N R_i(\sigma_i)\right] \leq CN^2 \log A/\sqrt{p}$.

With the above regret upper bound, we can then guarantee the time complexity to reach a weak ϵ-CCE.

Corollary 1. *With* $p = NA/Cost$ *and the hyper-parameters defined as Theorem 2, the time complexity of Algorithm 1 to reach a weak* ϵ*-CCE is* $O\left(Cost + N^{7/2}\sqrt{ACost}\log A/\epsilon\right)$.

6.2 Individual Regret

In this subsection, we provide the time complexity to reach a strong ϵ-CCE. As previous analysis, we will also first give the upper bound for the individual regret. Here we take the mirror map $h_2(x) = \sum_{a=1}^d x^2(a)$ as an example.

Before providing the upper bound for the individual regret, we first present a useful lemma to bound the term $\mathbb{E}\left[\|\sigma_j^k - w_j^{k-1}\|_1^2\right]$ appearing in the general upper bound in Theorem 1.

Lemma 4. *If the hyper-parameter* $\alpha = 1$, *then* $\mathbb{E}\left[\|\sigma_j^k - w_j^{k-1}\|_1^2\right] \leq 2\tau^2/p^2$.

With Lemma 4, we now provide an upper bound on individual regret.

Theorem 3. *With* $\tau = \left(A\left(2 + 1/\gamma\right)(N-1)^2 T/p^2\right)^{-1/4}$, $\alpha = 1$, $\gamma \in (0, 1/2)$ *and mirror map* h_2, *we have* $\max_{\sigma_i} \mathbb{E}\left[R_i(\sigma_i)\right] = O\left(A(N-1)^2 T/p^2\right)^{1/4}$ *for any* $i \in [N]$.

The above individual regret upper bound in Theorem 3 implies the following time-complexity to reach a strong ϵ-CCE.

Corollary 2. *With* $p = NA/Cost$ *and the hyper-parameters defined in Theorem 3, the running time to reach a strong* ϵ*-CCE is* $O\left(Cost + N^2 ACost^{2/3}/\epsilon^{4/3}\right)$.

7 Conclusion

In this paper, we propose a stochastic version of OMD with variance reduction. Our algorithm extends prior works on variance-reduced stochastic algorithms from two-player zero-sum games to general games. The key innovation of this work is a low-variance Monte-Carlo estimator. In comparison with prior estimator in (CJST19), our estimator is exponentially fast with a slightly larger variance.

There are several directions to extend our algorithm: Firstly, despite our algorithm enjoying an $O(1/\epsilon)$ convergence rate to weak-CCE, its convergence rate to strong CCE is $O(1/\epsilon^{4/3})$ which seems to be sub-optimal in comparison to the convergence rate in (DFG21) and (PSS21). Thus, developing stochastic algorithms with an $O(1/\epsilon)$ convergence rate to strong-CCE is an interesting direction; Secondly, we only consider normal-form games. However, it is more realistic to consider games with sequential structures, e.g., extensive-form games. We hope this work could be a starting point for developing more efficient stochastic algorithms for general games.

References

[AM21] Ahmet Alacaoglu and Yura Malitsky. Stochastic variance reduction for variational inequality methods. *arXiv preprint arXiv:2102.08352*, 2021.

[Bro49] George W Brown. Some notes on computation of games solutions. Technical report, RAND CORP SANTA MONICA CA, 1949.

[BS18] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[BS19a] Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1829–1836, 2019.

[BS19b] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[CJST19] Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. *Advances in Neural Information Processing Systems*, 32, 2019.

[CP20] Xi Chen and Binghui Peng. Hedging in games: Faster convergence of external and swap regrets. *Advances in Neural Information Processing Systems*, 33:18990–18999, 2020.

[DFG21] Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, 34, 2021.

[GSBR20] Robert M Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

[HAM21] Yu-Guan Hsieh, Kimon Antonakopoulos, and Panayotis Mertikopoulos. Adaptive learning in continuous games: Optimal regret bounds and convergence to nash equilibrium. In *Conference on Learning Theory*, pages 2388–2422. PMLR, 2021.

[KV05] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[LWZB09] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. *Advances in Neural Information Processing Systems*, 22, 2009.

[MSB+17] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[NY83] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[PSS21] Georgios Piliouras, Ryann Sim, and Stratis Skoulakis. Optimal no-regret learning in general games: Bounded regret with unbounded step-sizes via clairvoyant mwu. *arXiv preprint arXiv:2111.14737*, 2021.

[Rob51] Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[RS13] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 26, 2013.

[SALS15] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 28, 2015.

[SBL+19] Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, and Michael Bowling. Variance reduction in monte carlo counterfactual regret minimization (vr-mccfr) for extensive form games using baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2157–2164, 2019.

[Sha64] Lloyd Shapley. Some topics in two-person games. *Advances in Game Theory*, 52:1–29, 1964.

[TBJB15] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit texas hold'em. In *International Joint Conference on Artificial Intelligence*, 2015.

[ZJBP07] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20, 2007.

[ZRL+19] Yichi Zhou, Tongzheng Ren, Jialian Li, Dong Yan, and Jun Zhu. Lazy-cfr: fast and near-optimal regret minimization for extensive games with imperfect information. In *International Conference on Learning Representations*, 2019.