Stochastic Regret Minimization in Extensive-Form Games

Gabriele Farina [1] **Christian Kroer** [2] **Tuomas Sandholm** [1 3 4 5]

Abstract

Monte-Carlo counterfactual regret minimization (MCCFR) is the state-of-the-art algorithm for solving sequential games that are too large for full tree traversals. It works by using gradient estimates that can be computed via sampling. However, stochastic methods for sequential games have not been investigated extensively beyond MCCFR. In this paper we develop a new framework for developing stochastic regret minimization methods. This framework allows us to use any regret-minimization algorithm, coupled with any gradient estimator. The MCCFR algorithm can be analyzed as a special case of our framework, and this analysis leads to significantly stronger theoretical guarantees on convergence, while simultaneously yielding a simplified proof. Our framework allows us to instantiate several new stochastic methods for solving sequential games. We show extensive experiments on five games, where some variants of our methods outperform MCCFR.

1. Introduction

Extensive-form games (EFGs) are a broad class of games that can model sequential and simultaneous moves, outcome uncertainty, and imperfect information. This includes real-world settings such as negotiation, sequential auctions, security games (Lisý et al., 2016; Munoz de Cote et al., 2013), cybersecurity games (DeBruhl et al., 2014; Chen et al., 2018), recreational games such as poker (Sandholm, 2010) and billiards (Archibald & Shoham, 2009), and medical treatment (Chen & Bowling, 2012; Sandholm, 2015).

[1]Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213 [2]IEOR Department, Columbia University, New York NY 10027 [3]Strategic Machine, Inc. [4]Strategy Robot, Inc. [5]Optimized Markets, Inc.. Correspondence to: Gabriele Farina <gfarina@cs.cmu.edu>, Christian Kroer <christian.kroer@columbia.edu>, Tuomas Sandholm <sandholm@cs.cmu.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Typically, EFG models are operationalized by computing either a *Nash equilibrium* of the game, or an approximate Nash equilibrium if the game is large. Approximate Nash equilibrium of zero-sum EFGs has been the underlying idea of several recent AI milestones, where strong AIs for two-player poker were created (Moravčík et al., 2017; Brown & Sandholm, 2017b). In principle, a zero-sum EFG can be solved in polynomial time using a linear program whose size is linear in the size of the game tree (von Stengel, 1996). However, for most real-world games this linear program is much too large to solve, either because it does not fit in memory, or because iterations of the simplex algorithm or interior-point methods become prohibitively expensive due to matrix inversion. Instead, first-order methods (Hoda et al., 2010; Kroer et al., 2020) or regret-based methods (Zinkevich et al., 2007; Tammelin et al., 2015; Brown & Sandholm, 2019a) are used in practice. These methods work by only keeping one or two strategies around for each player (typically the size of a strategy is much smaller than the size of the game tree). The game tree is then only accessed for computing gradients, which can be done via a single tree traversal (which can often be done without storing the tree), and sometimes game-specific structure can be exploited to speed this up further (Johanson et al., 2011). Finally, these gradients are used to update the strategy iterates.

However, for large games, even these gradient-based methods that require traversing the entire game tree are prohibitively expensive (henceforth referred to as *deterministic methods*). This was seen in two recent superhuman poker AIs: Libratus (Brown & Sandholm, 2017b) and Pluribus (Brown & Sandholm, 2019b). Both AIs were generated in a two-stage manner: an offline blueprint strategy was computed, and then refinements to the blueprint solution were computed online while actually playing against human opponents. The online solutions were computed using deterministic methods (since those subgames are significantly smaller than the entire game). However, the original blueprint strategies had to be computed without traversing the entire game tree, as this game tree is far too large for even a moderate amount of traversals.

When full tree traversals are too expensive, stochastic methods can be used to compute approximate gradients instead. The most common stochastic method for solving large EFGs is the *Monte-Carlo Counterfactual Regret Minimization*

(MCCFR) algorithm (Lanctot et al., 2009). This algorithm, enhanced with certain dynamic pruning techniques, was also used to compute the blueprint strategies in the above-mentioned superhuman poker milestones (Brown & Sandholm, 2015; 2017a;b; 2019b; Brown et al., 2017). MCCFR combines the CFR algorithm (Zinkevich et al., 2007) with certain stochastic gradient estimators. Follow-up papers have been written on MCCFR, investigating various methods for improving the sampling schemes used in estimating gradients and so on (Gibson et al., 2012; Schmid et al., 2019). However, beyond the MCCFR setting, stochastic methods have not been studied extensively for solving EFGs[1].

In this paper we develop a general framework for constructing stochastic regret-minimization methods for solving EFGs. In particular, we introduce a way to combine *any* regret-minimizing algorithm with *any* gradient estimator, and obtain high-probability bounds on the performance of the resulting combined algorithm. As a first application of our approach, we show that with probability $1-p$, the regret in MCCFR is at most $O(\sqrt{\log(1/p)})$ worse than that of CFR, an exponential improvement over the bound $O(\sqrt{1/p})$ previously known in the literature. Second, our approach enables us to develop a slew of other stochastic methods for solving EFGs. As an example of our framework, we show how each of two popular online convex optimization algorithms, *follow-the-regularized-leader (FTRL)* and *online mirror descent (OMD)*, can be used to obtain stochastic EFG-solving algorithms with these guarantees. We then provide extensive numerical simulations on four diverse games, showing that it is possible to beat MCCFR in several of the games using our new methods. Because of the flexibility and modularity of our approach, it paves the way for many potential future investigations into stochastic methods for EFGs, either via better gradient estimators, via better deterministic regret minimization methods that can now be converted into stochastic methods, or both.

2. Preliminaries

2.1. Two-Player Zero-Sum Extensive-Form Games

In this subsection we introduce the notation that we will use in the rest the paper when dealing with two-player zero-sum extensive-form games.

An extensive-form game is played on a tree rooted at a node r. Each node v in the tree belongs to a player from the set $\{1, 2, \texttt{C}\}$, where $\texttt{C}$ is called the *chance player*. The chance player plays actions from a fixed distribution known to Player 1 and 2, and it is used as a device to model stochastic events such as drawing a random card from a deck. We denote the set of actions available at node v by A_v. Each action corresponds to an outgoing edges from v. Given $a \in A_v$, we let $\rho(v, a)$ denote the node that is reached by following the edge corresponding to action a at node v. Nodes v such that $A_v = \emptyset$ are called *leaves* and represent terminal states of the game. We denote by Z the set of leaves of the game. Associated with each leaf $z \in Z$ is a pair $(u_1(z), u_2(z)) \in \mathbb{R}^2$ of payoffs for Player 1 and 2, respectively. We denote by Δ the *payoff range* of the game, that is the value $\Delta := \max_{z\in Z} \max\{u_1(z), u_2(z)\} - \min_{z\in Z} \min\{u_1(z), u_2(z)\}$. In this paper we are concerned with *zero-sum* extensive-form games, that is games in which $u_1(z) = -u_2(z)$ for all $z \in Z$.

To model private information, the set of all nodes for each player $i \in \{1, 2, \texttt{C}\}$ is partitioned into a collection $\mathcal{I}_i$ of non-empty sets, called *information sets*. Each information set $I \in \mathcal{I}_i$ contains nodes that Player i cannot distinguish among. In this paper, we will only consider *perfect-recall* games, that is, games in which no player forgets what he or she observed or knew earlier. Necessarily, if two nodes u and v belong to the same information set I, the set of actions A_u and A_v must be the same (or the player would be able to tell u and v apart). So, we denote by A_I the set of actions of any node in I.

Sequences. The set of *sequences* for Player i, denoted Σ_i, is defined as the set of all possible information set-action pairs, plus a special element called *empty sequence* and denoted $\varnothing$. Formally, $\Sigma_i := \{(I, a) : I \in \mathcal{I}_i, a \in A_I\} \cup \{\varnothing\}$. Given a node v for Player i, we denote with $\sigma_i(v)$ the last information set-action pair of Player i encountered on the path from the root to node v; if the player does not act before v, $\sigma_i(I) = \varnothing$. It is known that in perfect-recall games $\sigma_i(u) = \sigma_i(v)$ for any two nodes u, v in the same information set. For this reason, for each information set I we define $\sigma_i(I)$ to equal $\sigma_i(v)$ for any $v \in I$.

Sequence-Form Strategies. A strategy for Player $i \in \{1, 2, \texttt{C}\}$ is an assignment of a probability distribution over the set of actions A_I to each information set I that belongs to Player i. In this paper, we represent strategies using their *sequence-form representation* (Romanovskii, 1962; Koller et al., 1996; von Stengel, 1996). A *sequence-form strategy* for Player i is a non-negative vector $\boldsymbol{z}$ indexed over the set of sequences Σ_i of that player. For each $\sigma = (I, a) \in \Sigma_i$, the entry $z[\sigma]$ contains the product of the probability of all the actions that Player i takes on the path from the root of the game tree down to action a at information set I, included. In order for these probabilities to be consistent, it is necessary and sufficient that $z[\varnothing] = 1$ and

$$\sum_{a\in A_I} z[(I, a)] = z[\sigma_i(I)] \quad \forall I \in \mathcal{I}_i.$$

A strategy such that exactly one action is selected with probability 1 at each node is called a *pure* strategy.

[1] Kroer et al. (2015) studies the stochastic mirror prox algorithm for EFGs, but it is not the primary focus of the paper, and seems to be more of a preliminary investigation.

We denote by $\mathcal{X}$ and $\mathcal{Y}$ the set of all sequence-form strategies for Player 1 and Player 2, respectively. We denote by $\boldsymbol{c}$ the fixed sequence-form strategy of the chance player.

For any leaf $z \in Z$, the probability that the game ends in z is the product of the probabilities of all the actions on the path from the root to z. Because of the definition of sequence-form strategies, when Player 1 and 2 play according to strategies $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$, respectively, this probability is equal to $x[\sigma_1(z)] \cdot y[\sigma_2(z)] \cdot c[\sigma_{\mathsf{C}}(z)]$. So, Player 2's expected utility is computed via the trilinear map

$$\bar{u}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c}) := \sum_{z \in Z} u_2(z) \cdot x[\sigma_1(z)] \cdot y[\sigma_2(z)] \cdot c[\sigma_{\mathsf{C}}(z)]. \quad (1)$$

Since the strategy of the chance player is fixed, the above expression is bilinear in $\boldsymbol{x}$ and $\boldsymbol{y}$ and therefore can be expressed more concisely as $\bar{u}_2(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{A}_2\, \boldsymbol{y}$, where $\boldsymbol{A}_2$ is called the *sequence-form payoff matrix* of Player 2.

2.2. Regret Minimization

In this section we present the regret minimization algorithms that we will work with. We will operate within the framework of *online convex optimization* (Zinkevich, 2003). In this setting, a decision maker repeatedly makes decisions $\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots$ from some convex compact set $\mathcal{Z} \subseteq \mathbb{R}^n$. After each decision $\boldsymbol{z}^t$ at time t, the decision maker faces a *linear loss* $\boldsymbol{z}^t \mapsto (\boldsymbol{\ell}^t)^\top \boldsymbol{z}^t$, where $\boldsymbol{\ell}^t$ is a *gradient vector* in $\mathbb{R}^n$.

Give $\hat{\boldsymbol{z}} \in \mathcal{Z}$, the *regret compared to* $\boldsymbol{z}$ of the regret minimizer up to time T, denoted as $R^T(\hat{\boldsymbol{z}})$, measures the difference between the loss cumulated by the sequence of output decisions $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^T$ and the loss that would have been cumulated by playing a fixed, time-independent decision $\hat{\boldsymbol{z}} \in \mathcal{Z}$. In symbols, $R^T(\hat{\boldsymbol{z}}) := \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top (\boldsymbol{z}^t - \hat{\boldsymbol{z}})$. A "good" regret minimizer is such that the regret compared to *any* $\hat{\boldsymbol{z}} \in \mathcal{Z}$ grows *sublinearly in* T.

The two algorithms beyond MCCFR that we consider assume access to a *distance-generating function* $d : \mathcal{Z} \to \mathbb{R}$, which is 1-strongly convex (with respect to some norm) and continuously differentiable on the interior of $\mathcal{Z}$. Furthermore, d should be such that the gradient of the convex conjugate $\nabla d(\boldsymbol{g}) = \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \langle \boldsymbol{g}, \boldsymbol{z} \rangle - d(\boldsymbol{z})$ is easy to compute. From d we also construct the *Bregman divergence* $D(\boldsymbol{z} \parallel \boldsymbol{z}') := d(\boldsymbol{z}) - d(\boldsymbol{z}') - \langle \nabla d(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}' \rangle$.

We will use the following two classical regret minimization algorithms as examples that can be used in the framework that we introduce in this paper. The *online mirror descent* (OMD) algorithm produces iterates according to the rule

$$\boldsymbol{z}^{t+1} = \operatorname*{arg\,min}_{\boldsymbol{z} \in \mathcal{Z}} \left\{ \langle \boldsymbol{\ell}^t, \boldsymbol{z} \rangle + \frac{1}{\eta} D(\boldsymbol{z} \parallel \boldsymbol{z}^t) \right\}. \quad (2)$$

The *follow the regularized leader* (FTRL) algorithm produces iterates according to the rule (Shalev-Shwartz & Singer, 2007)

$$\boldsymbol{z}^{t+1} = \operatorname*{arg\,min}_{\boldsymbol{z} \in \mathcal{Z}} \left\{ \left\langle \sum_{\tau=1}^{t} \boldsymbol{\ell}^\tau, \boldsymbol{z} \right\rangle + \frac{1}{\eta} d(\boldsymbol{z}) \right\}. \quad (3)$$

OMD and FTRL satisfy regret bounds of the form $\max_{\hat{\boldsymbol{z}} \in \mathcal{Z}} R^T(\hat{\boldsymbol{z}}) \leq 2L\sqrt{D(\boldsymbol{z}^* \| \boldsymbol{z}^1) T}$, where L is an upper bound on $\max_{\boldsymbol{x} \in \mathbb{R}^n} \frac{(\boldsymbol{\ell}^t)^\top \boldsymbol{x}}{\|\boldsymbol{x}\|}$ for all t. Here $\| \cdot \|$ is the norm with respect to which we measure strong convexity of d. (see, e.g., Orabona (2019)).

2.3. Equilibrium Finding in Extensive-Form Games using Regret Minimization

It is known that in a two-player extensive-form game, a Nash equilibrium (NE) is the solution to the *bilinear saddle-point problem*

$$\min_{\hat{\boldsymbol{x}} \in \mathcal{X}} \max_{\hat{\boldsymbol{y}} \in \mathcal{Y}} \hat{\boldsymbol{x}}^\top \boldsymbol{A}_2\, \hat{\boldsymbol{y}}.$$

Given a pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$ of sequence-form strategies for the Player 1 and 2, respectively, the *saddle-point gap*

$$\xi(\boldsymbol{x}, \boldsymbol{y}) := \max_{\hat{\boldsymbol{y}} \in \mathcal{Y}} \{\boldsymbol{x}^\top \boldsymbol{A}_2\, \hat{\boldsymbol{y}}\} - \min_{\hat{\boldsymbol{x}} \in \mathcal{X}} \{\hat{\boldsymbol{x}}^\top \boldsymbol{A}_2\, \boldsymbol{y}\}$$

measures of how far the pair is to being a Nash equilibrium. In particular, $(\boldsymbol{x}, \boldsymbol{y})$ is a Nash equilibrium if and only if $\xi(\boldsymbol{x}, \boldsymbol{y}) = 0$.

Regret minimizers can be used to find a sequence of points $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ whose saddle-point gap converges to 0. The fundamental idea is to instantiate two regret minimizers $\mathcal{R}_1$ and $\mathcal{R}_2$ for the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and let them respond to each other in a self-play fashion using a particular choice of loss vectors (see Figure 1).

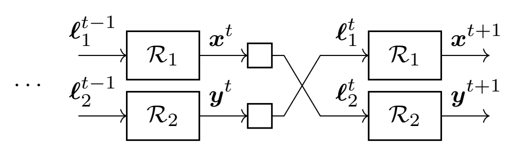

Figure 1. Self-play method for computing NE in EFGs.

At each time t, the strategies $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$ output by the regret minimizers are used to compute the loss vectors

$$\boldsymbol{\ell}_1^t := \boldsymbol{A}_2\, \boldsymbol{y}^t, \quad \boldsymbol{\ell}_2^t := -\boldsymbol{A}_2^\top\, \boldsymbol{x}^t. \quad (4)$$

Let $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ be the average of the strategies output by $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively, up to time T. Furthermore, let $R_1^T := \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} R_1^T(\hat{\boldsymbol{x}})$ and $R_2^T := \max_{\hat{\boldsymbol{y}} \in \mathcal{Y}} R_2^T(\hat{\boldsymbol{y}})$ be the maximum regret cumulated by $\mathcal{R}_1$ and $\mathcal{R}_2$ against any sequence-form strategy in $\mathcal{X}$ and $\mathcal{Y}$, respectively. A well-known folk lemma asserts that

$$\xi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \leq (R_1^T + R_2^T)/T.$$

So, if $\mathcal{R}_1$ and $\mathcal{R}_2$ have regret that grows sublinearly, then the strategy profile $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ converges to a saddle point.

3. Stochastic Regret Minimization for Extensive-Form Games

In this section we provide some key analytical tools to understand the performance of regret minimization algorithms when gradient estimates are used instead of exact gradient vectors. The results in this sections are complemented by those of Section 4, where we introduce computationally cheap gradient estimators for the purposes of equilibrium finding in extensive-form games.

3.1. Regret Guarantees when Gradient Estimators are Used

We start by studying how much the guarantee on the regret degrades when gradient estimators are used instead of exact gradient vectors. Our analysis need not assume that we operate over extensive-form strategy spaces, so we present our results in full generality.

Let $\tilde{\mathcal{R}}$ be a deterministic regret minimizer over a convex and compact set $\mathcal{Z}$, and consider a second regret minimizer $\mathcal{R}$ over the same set $\mathcal{Z}$ that is implemented starting from $\tilde{\mathcal{R}}$ as in Figure 2. In particular, at all times t,

- $\mathcal{R}$ queries the next decision $\boldsymbol{z}^t$ of $\tilde{\mathcal{R}}$, and outputs it;
- each gradient vector $\boldsymbol{\ell}^t$ received by $\mathcal{R}$ is used by $\mathcal{R}$ to compute a *gradient estimate* $\tilde{\boldsymbol{\ell}}^t$ such that
$$\mathbb{E}_t[\tilde{\boldsymbol{\ell}}^t] := \mathbb{E}[\tilde{\boldsymbol{\ell}}^t \mid \tilde{\boldsymbol{\ell}}^1, \dots, \tilde{\boldsymbol{\ell}}^{t-1}] = \boldsymbol{\ell}^t.$$
(that is, the estimate in unbiased). The internal regret minimizer $\tilde{\mathcal{R}}$ is then shown $\tilde{\boldsymbol{\ell}}^t$ instead of $\boldsymbol{\ell}^t$.

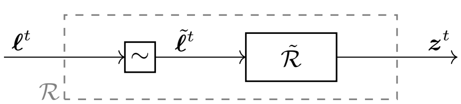

Figure 2. Abstract regret minimizer considered in Section 3.1.

The regret minimizer $\mathcal{R}$ is a purely conceptual construction. We introduce $\mathcal{R}$ in order to compare the regret incurred by $\mathcal{R}$ to that incurred by $\tilde{\mathcal{R}}$. This will allow us to quantify the degradation in regret that is incurred when the gradient vectors are estimated instead of exact. In practice, it is not necessary to explicitly construct $\mathcal{R}$ and fully observe the gradient vectors $\boldsymbol{\ell}^t$ in order to compute the estimates $\tilde{\boldsymbol{\ell}}^t$. Examples of cheap gradient estimators for extensive-form games are given in Section 4.

When the estimate of the gradient is very accurate (for instance, it has low variance), it is reasonable to expect that the regret R^T incurred by $\mathcal{R}$ up to any time T is roughly equal to the regret $\tilde{R}^T$ that is incurred by $\tilde{\mathcal{R}}$, plus some degradation term that depends on the error of the estimates. We can quantify this relationship by fixing an arbitrary $\boldsymbol{u} \in \mathcal{Z}$ and introducing the discrete-time stochastic process

$$d^t := (\boldsymbol{\ell}^t)^\top(\boldsymbol{z}^t - \boldsymbol{u}) - (\tilde{\boldsymbol{\ell}}^t)^\top(\boldsymbol{z}^t - \boldsymbol{u}). \tag{5}$$

Since by hypothesis $\mathbb{E}_t[\tilde{\boldsymbol{\ell}}^t] = \boldsymbol{\ell}^t$ and $\tilde{\mathcal{R}}$ is a deterministic regret minimizer, $\mathbb{E}_t[d^t] = 0$ and so $\{d^t\}$ is a martingale difference sequence. This martingale difference sequence is well-known, especially in the context of *bandit* regret minimization (Abernethy & Rakhlin, 2009; Bartlett et al., 2008). Using the Azuma-Hoeffding concentration inequality (Hoeffding, 1963; Azuma, 1967), we can prove the following.

Proposition 1. *Let M and $\tilde{M}$ be positive constants such that $|(\boldsymbol{\ell}^t)^\top(\boldsymbol{z} - \boldsymbol{z}')| \le M$ and $|(\tilde{\boldsymbol{\ell}})^\top(\boldsymbol{z} - \boldsymbol{z}')| \le \tilde{M}$ for all times $t = 1, \dots, T$ and all feasible points $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$. Then, for all $p \in (0,1)$ and all $\boldsymbol{u} \in \mathcal{Z}$,*

$$\mathbb{P}\left[R^T(\boldsymbol{u}) \le \tilde{R}^T(\boldsymbol{u}) + (M + \tilde{M})\sqrt{2T \log \frac{1}{p}}\,\right] \ge 1 - p.$$

A straightforward consequence of Proposition 1 is that if $\tilde{\mathcal{R}}$ has regret that grows sublinearly in T, then also the regret of $\mathcal{R}$ will grow sublinearly in T with high probability.

Remark. As shown in Proposition 1, using gradient estimators instead of exact gradients incurs an additive regret degradation term that scales proportionally with the bound $\tilde{M}$ on the norm of the gradient estimates $\tilde{\boldsymbol{\ell}}^t$. We remark that the regret $\tilde{R}^T(\boldsymbol{u})$ also scales proportionally to the norm of the gradient estimates $\tilde{\boldsymbol{\ell}}^t$. So, increasing the value of p in Proposition 1 is not enough to counter the dependence on $\tilde{M}$.

3.2. Connection to Equilibrium Finding

We now apply the general theory of Section 3.1 for the specific application of this paper—that is, Nash equilibrium computation in large extensive-form games.

We start from the construction of Section 2.3. In particular, we instantiate two deterministic regret minimizers $\tilde{\mathcal{R}}_1, \tilde{\mathcal{R}}_2$ and let them play strategies against each other. However, instead of computing the exact losses $\boldsymbol{\ell}_1^t$ and $\boldsymbol{\ell}_2^t$ as in (4), we compute their estimates $\tilde{\boldsymbol{\ell}}_1^t$ and $\tilde{\boldsymbol{\ell}}_2^t$ according to some algorithm that guarantees that $\mathbb{E}_t[\tilde{\boldsymbol{\ell}}_1^t] = \boldsymbol{\ell}_1^t$ and $\mathbb{E}_t[\tilde{\boldsymbol{\ell}}_2^t] = \boldsymbol{\ell}_2^t$ at all times t. We will show that despite this modification, the average strategy profile has a saddle point gap that is guaranteed to converge to zero with high probability.

Because of the particular definition of $\boldsymbol{\ell}_1^t$, we have that at all times t,

$$\max_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}} \left|(\boldsymbol{\ell}_1^t)^\top(\boldsymbol{x} - \boldsymbol{x}')\right| = \max_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}} \left|(\boldsymbol{x}^t)^\top \boldsymbol{A}_2 \boldsymbol{y}^t - (\boldsymbol{x}')^\top \boldsymbol{A}_2 \boldsymbol{y}^t\right| = \Delta,$$

where Δ is the payoff range of the game (see Section 2.1). (A symmetric statement holds for Player 2.) For $i \in \{1, 2\}$, let $\tilde{M}_i$ be positive constants such that $|(\tilde{\boldsymbol{\ell}}_i^t)^\top(\boldsymbol{z} - \boldsymbol{z}')| \le \tilde{M}_i$ at all times $t = 1, \dots, T$ and all strategies $\boldsymbol{z}, \boldsymbol{z}'$ in the sequence-form polytope for Player i (that is, $\mathcal{X}$ when $i = 1$ and $\mathcal{Y}$ when $i = 2$). Using Proposition 1, we find that for all $\hat{\boldsymbol{x}} \in \mathcal{X}$

and $\hat{\boldsymbol{y}} \in \mathcal{Y}$, with probability (at least) $1-p$,

$$\sum_{t=1}^{T} (\boldsymbol{x}^t - \hat{\boldsymbol{x}})^\top \boldsymbol{A}_2\, \boldsymbol{y}^t \le \tilde{R}_1^T(\hat{\boldsymbol{x}}) + (\Delta + \tilde{M}_1)\sqrt{2T \log \frac{1}{p}}$$

$$-\sum_{t=1}^{T} (\boldsymbol{x}^t)^\top \boldsymbol{A}_2\, (\boldsymbol{y}^t - \hat{\boldsymbol{y}}) \le \tilde{R}_2^T(\hat{\boldsymbol{y}}) + (\Delta + \tilde{M}_2)\sqrt{2T \log \frac{1}{p}}$$

where $\tilde{R}_i$ denotes the regret of the regret minimizer $\tilde{\mathcal{R}}_i$ that at each time t observes $\tilde{\boldsymbol{\ell}}_i^t$.

Summing the above inequalities, dividing by T, and using the union bound, we obtain that, with probability at least $1-2p$,

$$\begin{aligned}\bar{\boldsymbol{x}}^\top \boldsymbol{A}_2\, \hat{\boldsymbol{y}} - \hat{\boldsymbol{x}}^\top \boldsymbol{A}_2\, \bar{\boldsymbol{y}} &\le (\tilde{R}_1^T(\hat{\boldsymbol{x}}) + \tilde{R}_2^T(\hat{\boldsymbol{y}}))/T \\ &\quad + (2\Delta + \tilde{M}_1 + \tilde{M}_2)\sqrt{\frac{2}{T} \log \frac{1}{p}},\end{aligned} \tag{6}$$

where $\bar{\boldsymbol{x}} := \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}^t$ and $\bar{\boldsymbol{y}} := \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}^t$. Since (6) holds for all $\hat{\boldsymbol{x}} \in \mathcal{X}$ and $\hat{\boldsymbol{y}} \in \mathcal{Y}$, we obtain the following.

Proposition 2. *With probability at least* $1-2p$*,*

$$\xi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \le \frac{\tilde{R}_1^T(\hat{\boldsymbol{x}}) + \tilde{R}_2^T(\hat{\boldsymbol{y}})}{T} + (2\Delta + \tilde{M}_1 + \tilde{M}_2)\sqrt{\frac{2}{T} \log \frac{1}{p}}.$$

If $\tilde{\mathcal{R}}_1$ and $\tilde{\mathcal{R}}_2$ have regret that is sublinear in T, then we conclude that the saddle point gap $\xi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ converges to 0 with high probability like in the non-stochastic setting. So, $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ converges to a saddle point over time.

4. Game-Theoretic Gradient Estimators

We complete the theory of Sections 3.1 and 3.2 by showing some examples of computationally cheap gradient estimators designed for game-theoretic applications. We will illustrate how each technique can be used to construct an estimate $\tilde{\boldsymbol{\ell}}_1^t$ for the gradient $\boldsymbol{\ell}_1^t = \boldsymbol{A}_2\, \boldsymbol{y}^t$ for Player 1 defined in (4). The computation of an estimate for $\boldsymbol{\ell}_2^t$ is analogous.

4.1. External Sampling

An unbiased estimator of the gradient vector $\boldsymbol{\ell}_1^t = \boldsymbol{A}_2\, \boldsymbol{y}^t$ can be easily constructed by independently sampling *pure* strategies $\tilde{\boldsymbol{y}}^t$ for Player 2 and $\tilde{\boldsymbol{c}}^t$ for the chance player. Indeed, as long as $\mathbb{E}_t[\tilde{\boldsymbol{y}}^t] = \boldsymbol{y}^t$ and $\mathbb{E}_t[\tilde{\boldsymbol{c}}^t] = \boldsymbol{c}$, from (1) we have that for all $\boldsymbol{x} \in \mathcal{X}$, $\bar{u}_2(\boldsymbol{x}, \boldsymbol{y}^t, \boldsymbol{c}) = \mathbb{E}_t[\bar{u}_2(\boldsymbol{x}, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{c}}^t)]$. Hence, the vector corresponding to the (random) linear function $\boldsymbol{x} \mapsto \bar{u}_2(\boldsymbol{x}, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{c}}^t)$ is an unbiased gradient estimator, called the *external sampling* gradient estimator.

Since at all times t, $\tilde{\boldsymbol{y}}^t$ and $\tilde{\boldsymbol{c}}^t$ are sequence-form strategies, $\bar{u}_2(\boldsymbol{x}, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{c}}^t)$ is lower bounded by the minimum payoff of the game and upper bounded by the maximum payoff of the game. Hence, for this estimator, $\tilde{M}$ in Proposition 1 is equal to the payoff range Δ of the game. Substituting that value into Proposition 2, we conclude that when the external sampling gradient estimator is used to estimate the gradient for both players, with probability at least $1-2p$ the saddle point gap of the average strategy profile $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is

$$\xi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \le \frac{\tilde{R}_1^T(\hat{\boldsymbol{x}}) + \tilde{R}_2^T(\hat{\boldsymbol{y}})}{T} + 4\Delta\sqrt{\frac{2}{T} \log \frac{1}{p}}. \tag{7}$$

The external sampling gradient estimator, that is, the vector corresponding to the linear function $\boldsymbol{x} \mapsto \bar{u}_2(\boldsymbol{x}, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{c}}^t)$, can be computed via a simple traversal of the game tree. The algorithm starts at the root of the game tree and starts visiting the tree. Every time a node that belongs to the chance player or to Player 2 is encountered, an action is sampled according to the strategy $\boldsymbol{c}$ or $\boldsymbol{y}^t$, respectively. Every time a node for Player 1 is encountered, the algorithm branches on all possible actions and recurses. A simple linear-time implementation is given as Algorithm 1. For every node of Player 2 or chance player, the algorithm branches on only one action. Thus computing an external sampling gradient estimate is significantly cheaper to compute than the exact gradient $\boldsymbol{\ell}_1^t$.

Algorithm 1: Efficient implementation of the external sampling gradient estimator

Input: $\boldsymbol{y}^t$ strategy for Player 2
Output: $\tilde{\boldsymbol{\ell}}_1^t$ unbiased gradient estimate for $\boldsymbol{\ell}_1^t$ defined in (4)

$\tilde{\boldsymbol{\ell}}_1^t \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma_1|}$
subroutine TraverseAndSample(v)
  $I \leftarrow$ infoset to which v belongs
  if v is a leaf **then**
    $\tilde{\ell}_1^t[\sigma_1(v)] \leftarrow u_1(v)$
  else if v belongs to the chance player **then**
    Sample an action $a^* \sim \left(\frac{c[(I,a)]}{c[\sigma_c(I)]}\right)_{a \in A_v}$
    TraverseAndSample($\rho(v, a^*)$)
  else if v belongs to Player 2 **then**
    Sample an action $a^* \sim \left(\frac{y^t[(I,a)]}{y^t[\sigma_2(I)]}\right)_{a \in A_v}$
    TraverseAndSample($\rho(v, a^*)$)
  else if v belongs to Player 1 **then**
    for $a \in A_v$ **do**
      TraverseAndSample($\rho(v, a)$)
TraverseAndSample(r) ▷ r is root of the game tree
return $\tilde{\boldsymbol{\ell}}_1^t$

Remark. Analogous estimators where only the chance player's strategy $\boldsymbol{c}$ or only Player 2's strategy $\boldsymbol{y}^t$ are sampled are referred to as *chance sampling* estimator and *opponent sampling* estimator, respectively. In both cases, the same value of $\tilde{M} = \Delta$ (and therefore the bound in (7)) applies.

Remark. In the special case in which $\mathcal{R}_1$ and $\mathcal{R}_2$ run the CFR regret minimization algorithm, our analysis immediately implies the correctness of external-sampling MCCFR, chance-sampling MCCFR, and opponent-sampling MCCFR, while at the same time yielding a significant improvement over the theoretical convergence rate to Nash equilibrium of the overall algorithm: the right hand side of (7) grows as $\sqrt{\log(1/p)}$ in p, compared to the $O(\sqrt{1/p})$ of the original analysis by Lanctot et al. (2009).

Finally, we remark that our regret bound has a more favorable dependence on game-specific constants (for example, the number of information sets of each player) than the original analysis by (Lanctot et al., 2009).

4.2. Outcome Sampling

Let $\boldsymbol{w}^t \in \mathcal{X}$ be an arbitrary strategy for Player 1. Furthermore, let $\tilde{z}^t \in Z$ be a random variable such that for all $z \in Z$,

$$\mathbb{P}_t[\tilde{z}^t = z] = w^t[\sigma_1(z)] \cdot y^t[\sigma_2(z)] \cdot c[\sigma_{\mathsf{c}}(z)],$$

and let $\boldsymbol{e}_z$ be defined as the vector such that $e_z[\sigma_1(z)] = 1$ and $e_z[\sigma] = 0$ for all other $\sigma \in \Sigma_1, \sigma \neq \sigma_1(z)$. It is a simple exercise to prove that the random vector

$$\tilde{\boldsymbol{\ell}}_1^t := \frac{u_2(\tilde{z}^t)}{w^t[\sigma_1(\tilde{z}^t)]}\boldsymbol{e}_{\tilde{z}^t}$$

is such that $\mathbb{E}_t[\tilde{\boldsymbol{\ell}}_1^t] = \boldsymbol{\ell}_1^t$ (see Appendix A for a proof). This particular definition of $\tilde{\boldsymbol{\ell}}_1^t$ is called the *outcome sampling* gradient estimator.

Computationally, the outcome sampling gradient estimator is cheaper than the external sampling gradient estimator. Indeed, since $\boldsymbol{w}^t \in \mathcal{X}$, one can sample $\tilde{z}^t$ by following a random path from the root of the game tree by sampling (from the appropriate player's strategy) one action at each node encountered along the way. The walk terminates as soon as it reaches a leaf, which corresponds to $\tilde{z}$.

As we show in Appendix A, the value of $\tilde{M}$ for the outcome sampling gradient estimator is

$$\tilde{M} = \Delta \cdot \max_{\sigma \in \Sigma_1} \frac{1}{w^t[\sigma]}.$$

So, the high-probability bound on the saddle point gap is inversely proportional to the minimum entry in w^t, as already noted by Lanctot et al. (2009).

4.2.1. Exploration-Balanced Outcome Sampling

In Appendix A we show that a strategy $\boldsymbol{w}^*$ exists such that $w^*[\sigma] \geq 1/(|\Sigma_1| - 1)$ for every $\sigma \in \Sigma_1$. Since $\boldsymbol{w}^*$ guarantees that all of the $|\Sigma_1|$ entries of w^* are at least $1/(|\Sigma_1| - 1)$, we call $\boldsymbol{w}^*$ the *exploration-balanced strategy*, and the corresponding outcome sampling regret estimator the *exploration-balanced outcome sampling* regret estimator. As a consequence of the above analysis, when both players' gradients are estimated using the exploration-balanced outcome sampling regret estimator, with probability at least $1 - 2p$ the saddle point gap of the average strategy profile $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is upper bounded as

$$\xi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \leq \frac{\tilde{R}_1^T(\hat{\boldsymbol{x}}) + \tilde{R}_2^T(\hat{\boldsymbol{y}})}{T} + 2(|\Sigma_1| + |\Sigma_2|)\Delta\sqrt{\frac{2}{T}\log\frac{1}{p}}.$$

To our knowledge, this is the first time that the exploration-balanced outcome sampling gradient estimator has been introduced.

The final remark of Section 4.1 applies to outcome sampling as well.

5. Experiments

In this section we perform numerical simulations to investigate the practical performance of several stochastic regret-minimization algorithms. First, we have the MCCFR algorithm instantiated with *regret matching* (Hart & Mas-Colell, 2000). Second, we instantiate two algorithms through our framework: FTRL and OMD, both using the dilated entropy distance-generating function from Kroer et al. (2020), using their theoretically correct recursive scheme for information-set weights.[2] We will show two sections of experiments, one with external sampling and one with exploration-balanced outcome sampling.

For each game, we try four choices of stepsize η in FTRL and OMD: $0.1, 1, 10, 100$. For each algorithm-game pair we show only the best-performing of these four stepsizes in the plots below. The results for all stepsizes can be found in Appendix C. The stepsize is important: for most games where FTRL or OMD beats MCCFR, only the best stepsize does so. At the same time, we did not extensively tune stepsizes (four stepsizes increasing by a factor of 10 per choice leads to very coarse tuning), so there is room for better tuning of these. Figuring out how to intelligently choose, or adapt, stepsizes is an important future research direction to the present paper, and would likely lead to even faster algorithms.

For each game-algorithm pair, we run the experiment 50 times, in order to account for variance in gradient estimates. All plots show the mean performance, and each line is surrounded by shading indicating one standard deviation around the mean performance.

In each plot we show the number of nodes of the game tree touched on the x-axis. On the y-axis we show the saddle-point gap. All algorithms are run until the number of nodes touched corresponds to 50 full tree traversals (or, equivalently, 25 iterations of deterministic CFR or CFR$^+$).

We run our experiments on four different games. Below, we summarize some key properties of the games. The full description of each game is in Appendix B.

Leduc poker is a standard parametric benchmark game in the EFG-solving community (Southey et al., 2005). For our experiments we consider the largest variant of the game,

[2] We do this as opposed to constant information-set weights as used numerically by some past papers. Our preliminary experiments with constant weights gave worse results.

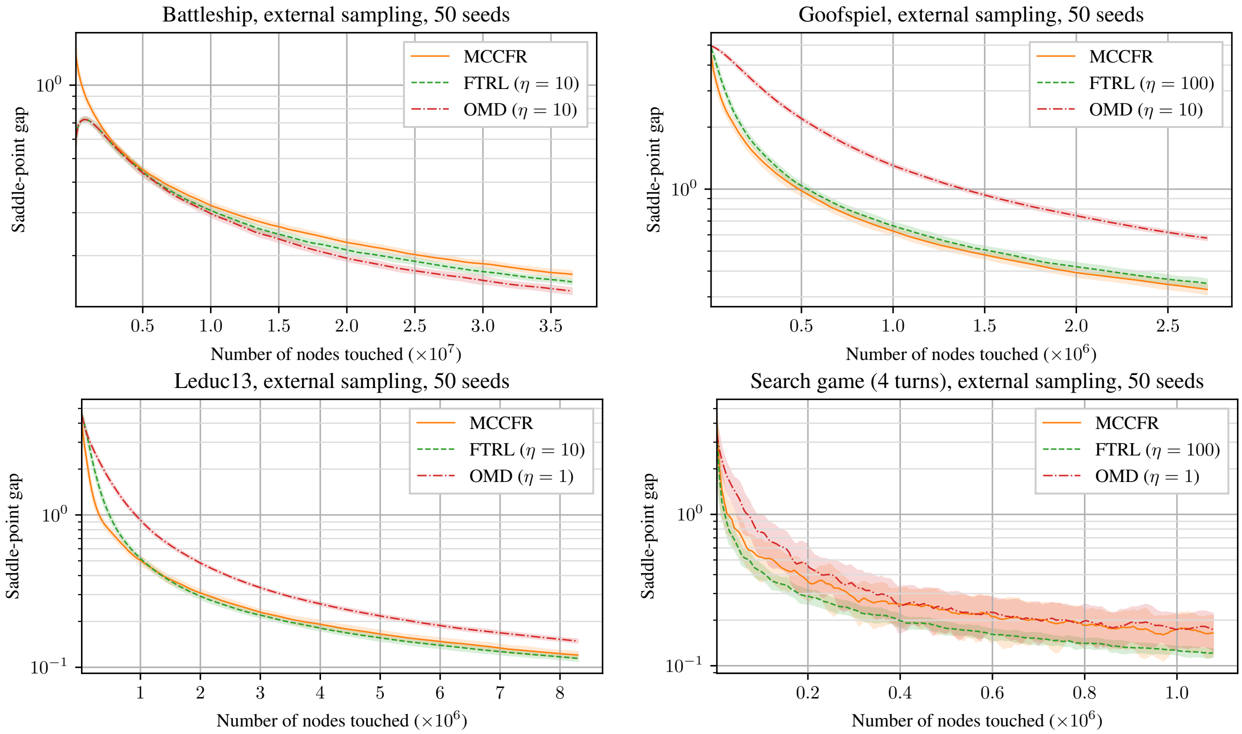

Figure 3. Performance of MCCFR, FTRL, and OMD when using the external sampling gradient estimator.

Leduc13. Leduc13 uses a deck of 13 unique cards, with two copies of each card. The game has 166,336 nodes and 6,007 sequences per player.

Goofspiel The variant of Goofspiel (Ross, 1971) that we use in our experiments is a two-player card game, employing three identical decks of 4 cards each. This game has 54,421 nodes and 21,329 sequences per player.

Search is a security-inspired pursuit-evasion game. The game is played on a graph shown in Figure 5 in Appendix B. We consider two variants of the game, which differ in the number k of simultaneous moves allowed before the game ties out. Search-4 uses $k = 4$ and has 21,613 nodes, 2,029 defender sequences, and 52 attacker sequences. Search-5 uses $k = 5$ and has 87,972 nodes, 11,830 defender sequences, and 69 attacker sequences. Our search game is a zero-sum variant of the one used by Kroer et al. (2018). A similar search game was considered by Bošanskỳ et al. (2014) and Bošanskỳ & Čermák (2015).

Battleship is a parametric version of a classic board game, where two competing fleets take turns shooting at each other (Farina et al., 2019c). The game has 732,607 nodes, 73,130 sequences for Player 1, and 253,940 sequences for Player 2.

5.1. External Sampling

Figure 3 (top left) shows the performance on Battleship with external sampling. We see that both FTRL and OMD perform better than MCCFR when using stepsize $\eta = 10$. In Goofspiel (top right plot) we find that OMD performs significantly worse than MCCFR and FTRL. MCCFR performs slightly better than FTRL also. In Leduc 13 (bottom left) we find that OMD performs significantly worse than MCCFR and FTRL. FTRL performs slightly better than MCCFR. Finally, in Search-4 (bottom right) we find that OMD and MCCFR perform comparably, while FTRL performs significantly better. Due to space limitations, we show the experimental evaluation for Search-5 in Appendix C. In Search-5 all algorithms perform comparably, with FTRL performing slightly better than OMD and MCCFR.

Summarizing across all five games for external sampling, we see that FTRL, either with $\eta = 10$ or $\eta = 100$, was better than MCCFR on four out of five games (and essentially tied on the last game), with significantly better performance in the Search games. OMD performs significantly better then MCCFR and FTRL on Battleship.

5.2. Exploration-Balanced Outcome Sampling

Next, we investigate the performance of our exploration-balanced outcome sampling. For that gradient estimator we drew 100 outcome samples per gradient estimate, and use the empirical mean of those 100 samples as our estimate. The reason for this is that FTRL and OMD seem more sensitive to stepsize issues under outcome sampling. It can be shown easily that by averaging gradient estimators, the constant $\tilde{M}$ required in Proposition 1 does not increase.

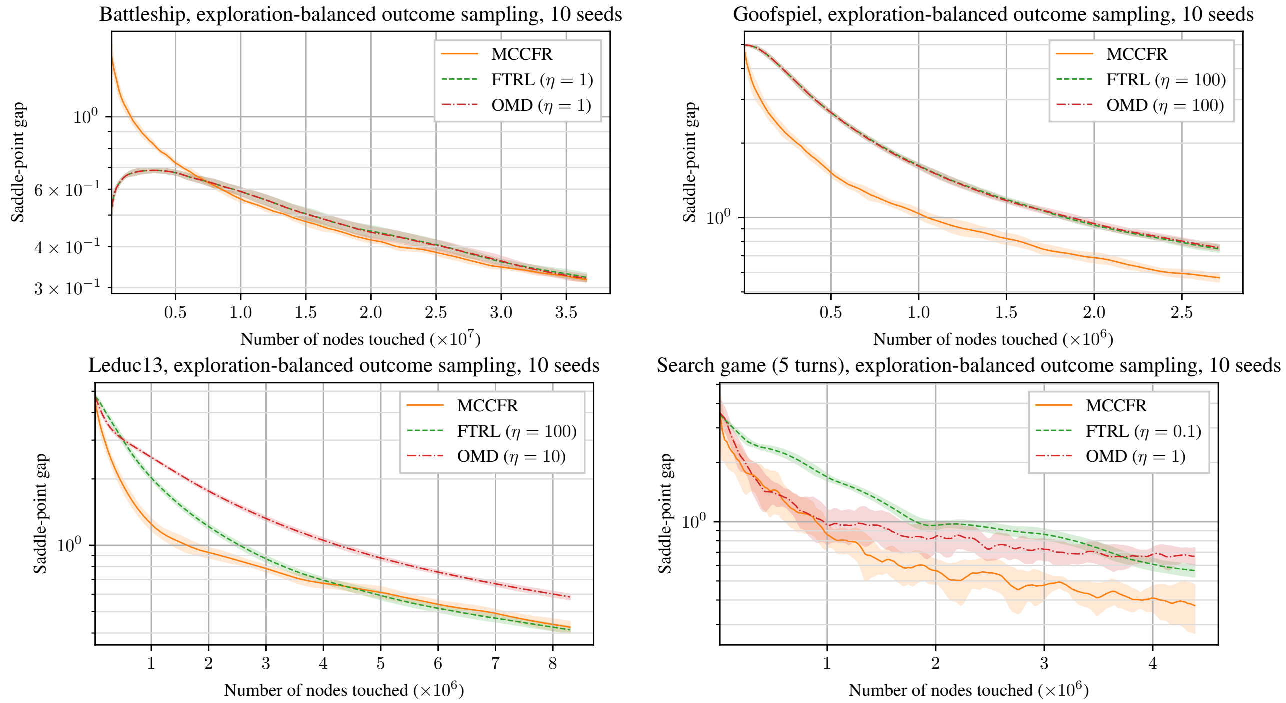

Figure 4. Performance of MCCFR, FTRL, and OMD when using the exploration-balanced outcome sampling gradient estimator.

Due to computational time issues, we present performance for only 10 random seeds per game in outcome sampling. For this reason we omit performance on Search-4, which seemed too noisy to make conclusions about. Search-4 plots can be found in Appendix C.

Figure 4 (top left) shows the performance on Battleship with outcome sampling. Here all algorithms perform essentially identically, with MCCFR performing significantly worse for a while, then slightly better, and then they all become similar around 3×10^7 nodes touched.

In Goofspiel (top right), MCCFR performs significantly better than both FTRL and OMD. Both FTRL and OMD were best with $\eta = 100$, our largest stepsize. It thus seems likely that even more aggressive stepsizes are needed in order to get better performance in Goofspiel.

In Leduc13 (bottom left), FTRL with outcome sampling is initially slower than MCCFR, but eventually overtakes it. OMD is significantly worse than the other algorithms.

Finally, in Search-5 (bottom right), MCCFR performs significantly better than FTRL and OMD, although FTRL seems to be catching up in later iterations.

Overall, when the exploration-balanced outcome sampling gradient estimator is used for all three algorithms, MCCFR seems to perform better than FTRL and OMD. In two out of four games it is significantly better, in one it is marginally better, and in one FTRL is marginally better. We hypothesize that FTRL and OMD are much more sensitive to stepsize issues with outcome sampling as opposed to external sampling. This would make sense, as the variance becomes much higher.

6. Conclusion

We introduced a new framework for constructing stochastic regret-minimization methods for solving zero-sum games. This framework completely decouples the choice of regret minimizer and gradient estimator, thus allowing any regret minimizer to be coupled with any gradient estimator. Our framework also yields a streamlined and dramatically simpler proof of MCCFR. Furthermore, it immediately gives a significantly stronger bound on the convergence rate of MCCFR, whereby with probability $1-p$ the regret grows as $O(\sqrt{T \log(1/p)})$ instead of $O(\sqrt{T/p})$ as in the original analysis—an exponentially better bound. We also instantiated stochastic variants of the FTRL and OMD algorithms for solving zero-sum EFGs using our framework. Extensive numerical experiments showed that it is often possible to beat MCCFR using these algorithms, even with a very mild amount of stepsize tuning. Due to its modular nature, our framework opens the door to many possible future research questions around stochastic methods for solving EFGs. Among the most promising are methods for controlling the stepsize in, for instance, FTRL or OMD, as well as instantiating our framework with other regret minimizers.

One potential avenue for future work is to develop gradient-estimation techniques with stronger control over the variance. In that case, it is possible to derive a variation of Proposition 1 that is based on the sum of conditional variances, an *intrinsic* notion of time in martingales (e.g., Blackwell & Freedman (1973)). In particular, using the Freedman-style (Freedman, 1975) concentration result of Bartlett et al. (2008) for martingale difference sequences, we obtain:

Proposition 3. *Let $T \geq 4$, and let M and $\tilde{M}$ be positive constants such that $|(\boldsymbol{\ell}^t)^\top(\boldsymbol{z}-\boldsymbol{u})| \leq M$ and $|(\tilde{\boldsymbol{\ell}})^\top(\boldsymbol{z}-\boldsymbol{u})| \leq \tilde{M}$ for all times $t = 1, \dots, T$ and all feasible points $\boldsymbol{z}, \boldsymbol{u} \in \mathcal{X}$. Furthermore, let $\sigma := \sqrt{\sum_{t=1}^T \mathrm{Var}[d^t \mid \tilde{\boldsymbol{\ell}}^1, \dots, \tilde{\boldsymbol{\ell}}^{t-1}]}$ be the square root of the sum of conditional variances of the random variables d^t introduced in* (5)*. Then, for all $p \in (0, 1/2]$ and all $\boldsymbol{u} \in \mathcal{X}$,*

$$\mathbb{P}\Big[R^T(\boldsymbol{u}) \leq \tilde{R}^T(\boldsymbol{u}) + 4\max\{\sigma\beta, (M+\tilde{M})\beta^2\}\Big] \geq 1-p,$$

where

$$\beta := \sqrt{\log\left(\frac{\log T}{p}\right)}.$$

The concentration result of Proposition 3 takes into account the variance of the martingale difference sequences. When the variance is low, the dominant term in the right hand side of the inequality is $(M+\tilde{M})\beta^2 = O(\log\log T)$. On the other hand, when the variance is high (that is, σ grows as $\sqrt{T}$), we recover a bound similar to the Azuma-Hoeffding inequality (albeit with a slightly worse polylog dependence on T).

Finally, our framework can also be applied to more general EFG-like problems, and thus this work also enables one to instantiate MCCFR or other stochastic methods for new sequential decision-making problems, for example by using the generalizations of CFR in Farina et al. (2019a) or Farina et al. (2019b).

Acknowledgments

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1617590, IIS-1901403, and CCF-1733556, and the ARO under awards W911NF-17-1-0082 and W911NF2010081. Gabriele Farina is supported by a Facebook fellowship.

References

Abernethy, J. D. and Rakhlin, A. Beating the adaptive bandit with high probability. *2009 Information Theory and Applications Workshop*, 2009.

Archibald, C. and Shoham, Y. Modeling billiards games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Budapest, Hungary, 2009.

Azuma, K. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967.

Bartlett, P. L., Dani, V., Hayes, T., Kakade, S., Rakhlin, A., and Tewari, A. High-probability regret bounds for bandit online linear optimization. In *Conference on Learning Theory (COLT)*, 2008.

Blackwell, D. and Freedman, D. On the amount of variance needed to escape from a strip. *The Annals of Probability*, pp. 772–787, 1973.

Bošanskỳ, B. and Čermák, J. Sequence-form algorithm for computing Stackelberg equilibria in extensive-form games. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Bošanskỳ, B., Kiekintveld, C., Lisý, V., and Pěchouček, M. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research*, pp. 829–866, 2014.

Brown, N. and Sandholm, T. Regret-based pruning in extensive-form games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

Brown, N. and Sandholm, T. Reduced space and faster convergence in imperfect-information games via pruning. In *International Conference on Machine Learning (ICML)*, 2017a.

Brown, N. and Sandholm, T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, pp. eaao1733, Dec. 2017b.

Brown, N. and Sandholm, T. Solving imperfect-information games via discounted regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019a.

Brown, N. and Sandholm, T. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019b.

Brown, N., Kroer, C., and Sandholm, T. Dynamic thresholding and pruning for regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Chen, K. and Bowling, M. Tractable objectives for robust policy optimization. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

Chen, X., Han, Z., Zhang, H., Xue, G., Xiao, Y., and Bennis, M. Wireless resource scheduling in virtualized radio access networks using stochastic learning. *IEEE Transactions on Mobile Computing*, (1):1–1, 2018.

DeBruhl, B., Kroer, C., Datta, A., Sandholm, T., and Tague, P. Power napping with loud neighbors: optimal energy-constrained jamming and anti-jamming. In *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*, pp. 117–128. ACM, 2014.

Farina, G., Kroer, C., and Sandholm, T. Online convex optimization for sequential decision processes and extensive-form games. In *AAAI Conference on Artificial Intelligence*, 2019a.

Farina, G., Kroer, C., and Sandholm, T. Regret circuits: Composability of regret minimizers. In *International Conference on Machine Learning*, pp. 1863–1872, 2019b.

Farina, G., Ling, C. K., Fang, F., and Sandholm, T. Correlation in extensive-form games: Saddle-point formulation and benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019c.

Freedman, D. A. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, 02 1975.

Gibson, R., Lanctot, M., Burch, N., Szafron, D., and Bowling, M. Generalized sampling and variance in counterfactual regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

Hart, S. and Mas-Colell, A. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68: 1127–1150, 2000.

Hoda, S., Gilpin, A., Peña, J., and Sandholm, T. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2), 2010.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Johanson, M., Waugh, K., Bowling, M., and Zinkevich, M. Accelerating best response calculation in large extensive games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

Koller, D., Megiddo, N., and von Stengel, B. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2), 1996.

Kroer, C., Waugh, K., Kılınç-Karzan, F., and Sandholm, T. Faster first-order methods for extensive-form game solving. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2015.

Kroer, C., Farina, G., and Sandholm, T. Robust stackelberg equilibria in extensive-form games and extension to limited lookahead. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Kroer, C., Waugh, K., Kılınç-Karzan, F., and Sandholm, T. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming*, 2020.

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte Carlo sampling for regret minimization in extensive games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.

Lisý, V., Davis, T., and Bowling, M. Counterfactual regret minimization in sequential security games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

McDiarmid, C. *Concentration*, pp. 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-662-12788-9. doi: 10.1007/978-3-662-12788-9_6.

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., and Bowling, M. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, May 2017.

Munoz de Cote, E., Stranders, R., Basilico, N., Gatti, N., and Jennings, N. Introducing alarms in adversarial patrolling games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1275–1276. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

Orabona, F. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Romanovskii, I. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962.

Ross, S. M. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8(3):621–625, 1971.

Sandholm, T. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, 2010. Special issue on Algorithmic Game Theory.

Sandholm, T. Steering evolution strategically: Computational game theory and opponent exploitation for treatment planning, drug design, and synthetic biology. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. Senior Member Track.

Schmid, M., Burch, N., Lanctot, M., Moravcik, M., Kadlec, R., and Bowling, M. 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, pp. 2157–2164, 2019.

Shalev-Shwartz, S. and Singer, Y. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3): 115–142, 2007.

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

Tammelin, O., Burch, N., Johanson, M., and Bowling, M. Solving heads-up limit Texas hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

von Stengel, B. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pp. 928–936, Washington, DC, USA, 2003.

Zinkevich, M., Bowling, M., Johanson, M., and Piccione, C. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

A. Proofs

A.1. Regret Guarantees when Gradient Estimators are Used

For completeness, we show a proof of Proposition 1. As mentioned, it is an application of the Azuma-Hoeffding inequality for martingale difference sequences, which we now state (see, e.g., Theorem 3.14 of McDiarmid (1998) for a proof).

Theorem 1 (Azuma-Hoeffding inequality)**.** *Let $Y_1, \ldots, Y_n$ be a martingale difference sequence with $a_k \le Y_k \le b_k$ for each k, for suitable constants a_k, b_k. Then for any $\tau \ge 0$,*

$$\mathbb{P}\Big[\sum Y_k \ge \tau\Big] \le e^{-2\tau^2/\sum(b_k-a_k)^2}.$$

Proposition 1. *Let M and $\tilde{M}$ be positive constants such that $|(\boldsymbol{\ell}^t)^\top(\boldsymbol{z}-\boldsymbol{z}')| \le M$ and $|(\tilde{\boldsymbol{\ell}})^\top(\boldsymbol{z}-\boldsymbol{z}')| \le \tilde{M}$ for all times $t = 1, \ldots, T$ and all feasible points $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$. Then, for all $p \in (0,1)$ and all $\boldsymbol{u} \in \mathcal{Z}$,*

$$\mathbb{P}\left[R^T(\boldsymbol{u}) \le \tilde{R}^T(\boldsymbol{u}) + (M+\tilde{M})\sqrt{2T\log\frac{1}{p}}\,\right] \ge 1-p.$$

Proof. As observed in the body, $d^t := (\boldsymbol{\ell}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u}) - (\tilde{\boldsymbol{\ell}}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u})$ is a martingale difference sequence. Furthermore, at all times t,

$$\begin{aligned}|d^t| &= |(\boldsymbol{\ell}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u}) - (\tilde{\boldsymbol{\ell}}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u})| \\ &\le |(\boldsymbol{\ell}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u})| + |(\tilde{\boldsymbol{\ell}}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u})| \\ &\le M+\tilde{M},\end{aligned} \tag{8}$$

and therefore $-(M+\tilde{M}) \le d^t \le (M+\tilde{M})$ for each t.

Furthermore,

$$\sum_{t=1}^T d^t = \left(\sum_{t=1}^T (\boldsymbol{\ell}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u})\right) - \left(\sum_{t=1}^T (\tilde{\boldsymbol{\ell}}^t)^\top(\boldsymbol{z}^t-\boldsymbol{u})\right) = R^T(\boldsymbol{u}) - \tilde{R}^T(\boldsymbol{u}).$$

So, using Theorem 1, for all $\tau \ge 0$

$$\begin{aligned}\mathbb{P}\Big[R^T(\boldsymbol{u}) \le \tilde{R}^T(\boldsymbol{u}) + \tau\Big] &= \mathbb{P}\left[\sum_{t=1}^T d^t \le \tau\right] \\ &= 1 - \mathbb{P}\left[\sum_{t=1}^T d^t \ge \tau\right] \\ &\ge 1 - \exp\left\{-\frac{2\tau^2}{\sum_{t=1}^T 4(M+\tilde{M})^2}\right\} \\ &= 1 - \exp\left\{-\frac{2\tau^2}{4T(M+\tilde{M})^2}\right\}.\end{aligned}$$

Finally, substituting $\tau = (M+\tilde{M})\sqrt{2T\log(1/p)}$ yields the statement. □

A.2. Properties of the Outcome Sampling Gradient Estimator

Let $\boldsymbol{w}^t \in \mathcal{X}$ be an arbitrary strategy for Player 1. Furthermore, let $\tilde{z}^t \in Z$ be a random variable such that for all $z \in Z$,

$$\mathbb{P}_t[\tilde{z}^t = z] = w^t[\sigma_1(z)] \cdot y^t[\sigma_2(z)] \cdot c[\sigma_{\mathsf{c}}(z)],$$

and let $\boldsymbol{e}_z$ be defined as the vector such that $e_z[\sigma_1(z)] = 1$ and $e_z[\sigma] = 0$ for all other $\sigma \in \Sigma_1, \sigma \ne \sigma_1(z)$.

Lemma 1. *The random vector*

$$\tilde{\boldsymbol{\ell}}_1^t := \frac{u_2(\tilde{z}^t)}{w^t[\sigma_1(\tilde{z}^t)]}\boldsymbol{e}_{\tilde{z}^t}$$

is such that $\mathbb{E}_t[\tilde{\boldsymbol{\ell}}_1^t] = \boldsymbol{\ell}_1^t$.

Proof. For all $\boldsymbol{x} \in \mathbb{R}^{|\Sigma_1|}$,

$$\begin{aligned}
\mathbb{E}_t[\boldsymbol{\ell}_1^t]^\top \boldsymbol{x} &= \left(\sum_{z\in Z} \mathbb{P}[\tilde{z}^t = z] \cdot \frac{u_1(z)}{w^t[\sigma_1(z)]}\boldsymbol{e}_z\right)^\top \boldsymbol{x} \\
&= \left(\sum_{z\in Z} u_2(z)\cdot y^t[\sigma_2(z)]\cdot c[\sigma_{\mathsf{C}}(z)]\cdot \boldsymbol{e}_z\right)^\top \boldsymbol{x} \\
&= \sum_{z\in Z} u_2(z)\cdot y^t[\sigma_2(z)]\cdot c[\sigma_{\mathsf{C}}(z)]\cdot (\boldsymbol{e}_z^\top \boldsymbol{x}) \\
&= \sum_{z\in Z} u_2(z)\cdot y^t[\sigma_2(z)]\cdot c[\sigma_{\mathsf{C}}(z)]\cdot x[\sigma_1(z)] \\
&= u_2(\boldsymbol{x}, \boldsymbol{y}^t, \boldsymbol{c}) = \boldsymbol{\ell}_1^\top \boldsymbol{x}.
\end{aligned}$$

Since the equality holds for all $\boldsymbol{x} \in \mathbb{R}^{|\Sigma_1|}$, we conclude $\mathbb{E}_t[\tilde{\boldsymbol{\ell}}_1^t] = \boldsymbol{\ell}_1$. □

Furthermore,

Lemma 2. *For all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$,*

$$(\tilde{\boldsymbol{\ell}}_1)^\top(\boldsymbol{x} - \boldsymbol{x}') \le \Delta \cdot \max_{\sigma\in\Sigma_1} \frac{1}{w^t[\sigma]}.$$

Proof. Using the definition of $\tilde{\boldsymbol{\ell}}_1$,

$$(\tilde{\boldsymbol{\ell}}_1)^\top(\boldsymbol{x} - \boldsymbol{x}') = \frac{u_2(\tilde{z}^t)}{w^t[\sigma_1(\tilde{z}^t)]}\Big(x[\sigma_1(\tilde{z}^t)] - x'[\sigma_1(\tilde{z}^t)]\Big).$$

Since each entry of $\boldsymbol{x}$ and $\boldsymbol{x}'$ is in the interval $[0, 1]$, the quantity $x[\sigma_1(\tilde{z}^t)] - x'[\sigma_1(\tilde{z}^t)]$ has absolute value in $[0, 1]$ as well. Hence,

$$\left|(\tilde{\boldsymbol{\ell}}_1)^\top(\boldsymbol{x} - \boldsymbol{x}')\right| \le \max_{z\in Z}\left|\frac{u_2(z)}{w^t[\sigma_1(z)])}\right| \le \Delta \cdot \max_{\sigma\in\Sigma_1} \frac{1}{w^t[\sigma]}$$

as we wanted to show. □

A.3. Exploration-Balanced Strategy

We now describe the construction of the *exploration-balanced strategy* $\boldsymbol{w}^*$. Given $\sigma \in \Sigma_1$, we let $\mathcal{C}_\sigma$ be the set of information sets $I in \mathcal{I}_1$ such that $\sigma_1(I) = \sigma$. Furthermore, let m_σ, for $\sigma \in \Sigma_1$, be the number of terminal sequences in the subtree rooted under σ; formally, m_σ is defined recursively as

$$m_\sigma = \begin{cases} 1 & \text{if } \mathcal{C}_\sigma = \varnothing; \\ \displaystyle\sum_{I\in\mathcal{C}_\sigma}\sum_{a\in A_I} m_{(I,a)} & \text{otherwise.} \end{cases}$$

Clearly, $m_\sigma \le |\Sigma_1| - 1$, since the empty sequence is never terminal (assuming Player 1 acts at least once). With that, we define $\boldsymbol{w}^*$ such that $w^*[\varnothing] = 1$ and that for all $\sigma = (I, a) \in \Sigma_1$,

$$w^*[\sigma] = \frac{m_\sigma}{\sum_{a'\in A_I} m_{(I,a')}} w^*[\sigma_1(I)].$$

It is immediate to verify that $\boldsymbol{w}^*$ is indeed a valid sequence-form strategy. Furthermore, since for all $I \in \mathcal{I}_1$, $I \in \mathcal{C}_{\sigma_1(I)}$, we have

$$\sum_{a'\in A_I} m_{(I,a')} \le m_{\sigma(I)}.$$

So,

$$w^*[\sigma] \ge \frac{m_\sigma}{m_{\sigma_1(I)}} w^*[\sigma_1(I)].$$

By recursively expanding the definition of $w^*[\sigma_1(I)]$ on the right-hand side until $\sigma_1(I) = \varnothing$, we ultimately obtain

$$w^*[\sigma] \ge \frac{1}{m_\varnothing} \ge \frac{1}{|\Sigma_1| - 1}$$

for all σ, as we wanted to show.

A.4. Proposition 3

As mentioned in the body of the paper, Proposition 3 is a direct consequence of the concentration result for martingale difference sequences of Bartlett et al. (2008), which we state next.

Lemma 3 (Lemma 2 of Bartlett et al. (2008))**.** *Suppose* $X^1, \dots, X^T$ *is a martingale difference sequence with* $|X^t| \le b$*. Let*

$$\mathrm{Var}_t\, X^t := \mathrm{Var}[X^t \mid X^1, \dots, X^{t-1}].$$

Let $V := \sum_{t=1}^T \mathrm{Var}_t\, X^t$ *be the sum of conditional variances of* X^t*'s. Further, let* $\sigma := \sqrt{V}$*. Then we have, for any* $\delta < 1/e$ *and* $T \ge 4$*,*

$$\mathbb{P}\left[\sum_{t=1}^T X^t > 2\max\{2\sigma, b\sqrt{\log(1/\delta)}\}\sqrt{\log(1/\delta)}\right] \le \log(T)\delta.$$

Proposition 3. *Let* $T \ge 4$*, and let* M *and* $\tilde{M}$ *be positive constants such that* $|(\boldsymbol{\ell}^t)^\top(\boldsymbol{z} - \boldsymbol{u})| \le M$ *and* $|(\tilde{\boldsymbol{\ell}})^\top(\boldsymbol{z} - \boldsymbol{u})| \le \tilde{M}$ *for all times* $t = 1, \dots, T$ *and all feasible points* $\boldsymbol{z}, \boldsymbol{u} \in \mathcal{X}$*. Furthermore, let* $\sigma := \sqrt{\sum_{t=1}^T \mathrm{Var}[d^t \mid \tilde{\boldsymbol{\ell}}^1, \dots, \tilde{\boldsymbol{\ell}}^{t-1}]}$ *be the square root of the sum of conditional variances of the random variables* d^t *introduced in* (5)*. Then, for all* $p \in (0, 1/2]$ *and all* $\boldsymbol{u} \in \mathcal{X}$*,*

$$\mathbb{P}\Big[R^T(\boldsymbol{u}) \le \tilde{R}^T(\boldsymbol{u}) + 4\max\{\sigma\beta, (M + \tilde{M})\beta^2\}\Big] \ge 1 - p,$$

where

$$\beta := \sqrt{\log\left(\frac{\log T}{p}\right)}.$$

Proof. We apply Lemma 3 to the martinagle difference sequence $X^t = d_t$. As argued in (8), $|X^t| \le (M + \tilde{M})$ at all times t, so the constant $b = M + \tilde{M}$ satisfies the requirements of Lemma 3. Finally, we set $\delta = p/\log(T)$ in Lemma 3, so that

$$\sqrt{\log(1/\delta)} = \sqrt{\log\left(\frac{\log T}{p}\right)} = \beta.$$

Furthermore, since by hypothesis $T \ge 4$ and $p \le 1/2$, $\delta = p/\log(T) \le 1/(2\log 4) \le 1/e$, so all hypotheses of Lemma 3 are satisfied. Hence, we have

$$\begin{aligned}
\mathbb{P}\Big[R^T(\boldsymbol{u}) - \tilde{R}^T(\boldsymbol{u}) \le 4\max\{\sigma\beta, (M + \tilde{M})\beta^2\}\Big] &= \mathbb{P}\left[\sum_{t=1}^T X^t \le 4\max\{\sigma\beta, b\beta^2\}\right] \\
&= \mathbb{P}\left[\sum_{t=1}^T X^t \le 4\max\{\sigma\sqrt{\log(1/\delta)}, b\log(1/\delta)\}\right] \\
&= \mathbb{P}\left[\sum_{t=1}^T X^t \le 2\max\{2\sigma, 2b\sqrt{\log(1/\delta)}\}\sqrt{\log(1/\delta)}\right] \\
&\ge \mathbb{P}\left[\sum_{t=1}^T X^t \le 2\max\{2\sigma, b\sqrt{\log(1/\delta)}\}\sqrt{\log(1/\delta)}\right] \\
&\ge 1 - \log(T)\delta = 1 - p,
\end{aligned}$$

where the last inequality follows from Lemma 3. □

B. Description of the Game Instances Used in the Experiments

We run our experiments on four different games, each described below.

Leduc poker is a standard benchmark in the EFG-solving community (Southey et al., 2005). Our variant, Leduc 13, has a deck of 13 unique cards, with two copies of each card. The game consists of two rounds. In the first round, each player places an ante of 1 in the pot and receives a single private card. A round of betting then takes place with a two-bet maximum, with Player 1 going first. A public shared card is then dealt face up and another round of betting takes place. Again, Player 1 goes first, and there is a two-bet maximum. If one of the players has a pair with the public card, that player wins. Otherwise, the player with the higher card wins. All bets in the first round are 1, while all bets in the second round are 2. This game has 166336 nodes and 6007 sequences per player.

Goofspiel The variant of Goofspiel (Ross, 1971) that we use in our experiments is a two-player card game, employing three identical decks of 4 cards each. At the beginning of the game, each player receives one of the decks to use it as its own hand, while the last deck is put face down between the players, with cards in increasing order of rank from top to bottom. Cards from this deck will be the prizes of the game. In each round, the players privately select a card from their hand as a bet to win the topmost card in the prize deck. The selected cards are simultaneously revealed, and the highest one wins the prize card. In case of a tie, the prize card is discarded. Each prize card's value is equal to its face value, and at the end of the game the players' score are computed as the sum of the values of the prize cards they have won. This game has 54421 nodes and 21329 sequences per player.

Search is a security-inspired pursuit-evasion game. The game is played on the graph shown in Figure 5.

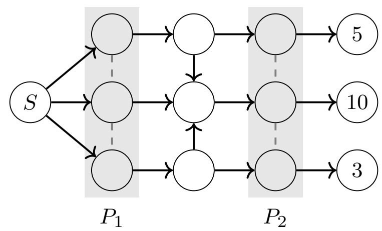

Figure 5. The graph on which the search game is played.

It is a simultaneous-move game (which can be modeled as a turn-taking EFG with appropriately chosen information sets). The defender controls two patrols that can each move within their respective shaded areas (labeled P1 and P2). At each time step the controller chooses a move for both patrols. The attacker is always at a single node on the graph, initially the leftmost node labeled S. The attacker can move freely to any adjacent node (except at patrolled nodes, the attacker cannot move from a patrolled node to another patrolled node). The attacker can also choose to wait in place for a time step in order to clean up their traces. If a patrol visits a node that was previously visited by the attacker, and the attacker did not wait to clean up their traces, they can see that the attacker was there. If the attacker reaches any of the rightmost nodes they receive the respective payoff at the node (5, 10, or 3, respectively). If the attacker and any patrol are on the same node at any time step, the attacker is captured, which leads to a payoff of -1 for the attacker and a payoff of 1 for the defender. Finally, the game times out after k simultaneous moves, in which case both players defender receive payoffs 0. Search-4 (Search-5) has 21613 (87,927) nodes, 2029 (11,830) defender sequences, and 52 (69) attacker sequences.

Our search game is a zero-sum variant of the one used by Kroer et al. (2018). A similar search game considered by Bošanskỳ et al. (2014) and Bošanskỳ & Čermák (2015).

Battleship is a parametric version of a classic board game, where two competing fleets take turns shooting at each other (Farina et al., 2019c). At the beginning of the game, the players take turns at secretly placing a set of ships on separate grids (one for each player) of size 3×2. Each ship has size 2 (measured in terms of contiguous grid cells) and a value of 1, and must be placed so that all the cells that make up the ship are fully contained within each player's grids and do not overlap with any other ship that the player has already positioned on the grid. After all ships have been placed. the players take turns at firing at their opponent. Ships that have been hit at all their cells are considered sunk. The game continues until either one player has sunk all of the opponent's ships, or each player has completed r shots. At the end of the game, each player's payoff is calculated as the sum of the values of the opponent's ships that were sunk, minus the sum of the values of ships which that player has lost. The game has 732607 nodes, 73130 sequences for player 1, and 253940 sequences for player 2.

C. Additional Experimental Results

C.1. External Sampling

The Search-5 plot omitted from the main paper is shown here.

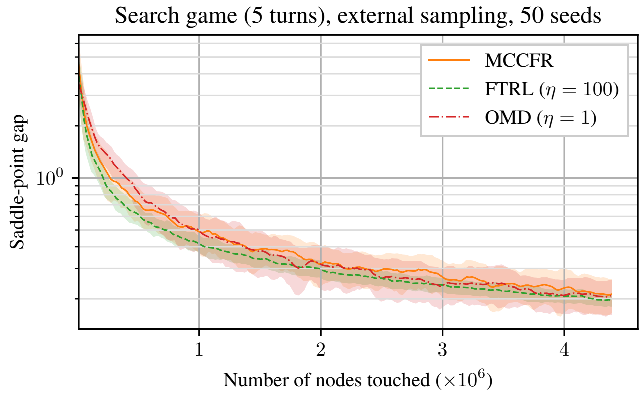

Figure 6. Performance of MCCFR, FTRL, and OMD with external sampling on Search-5.

Figures 7 through 11 show the performance of FTRL and OMD for all four stepsizes that we tried on each game: $\eta = 0.1, 1, 10, 100$.

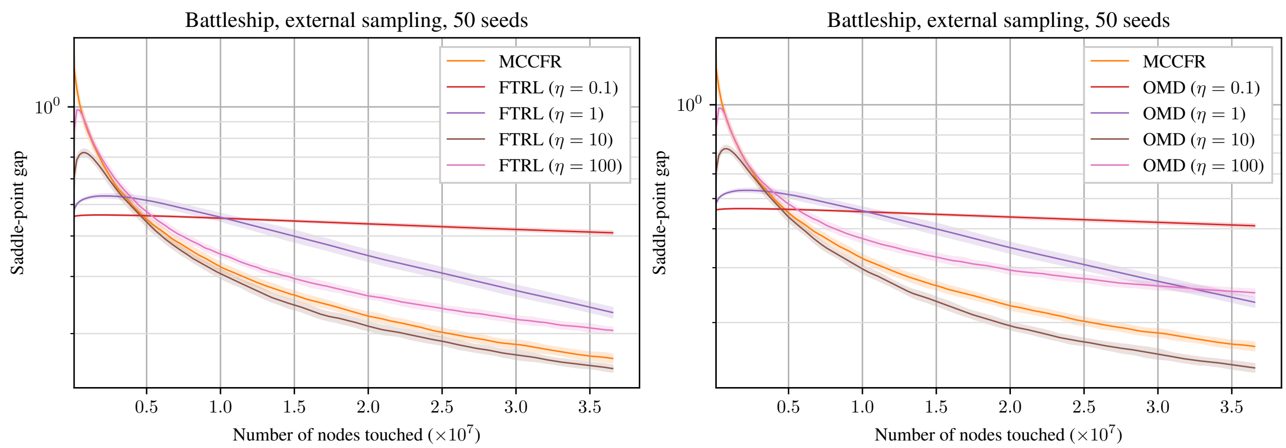

Figure 7. Performance of FTRL and OMD with four stepsizes on Battleship with external sampling. MCCFR shown for reference

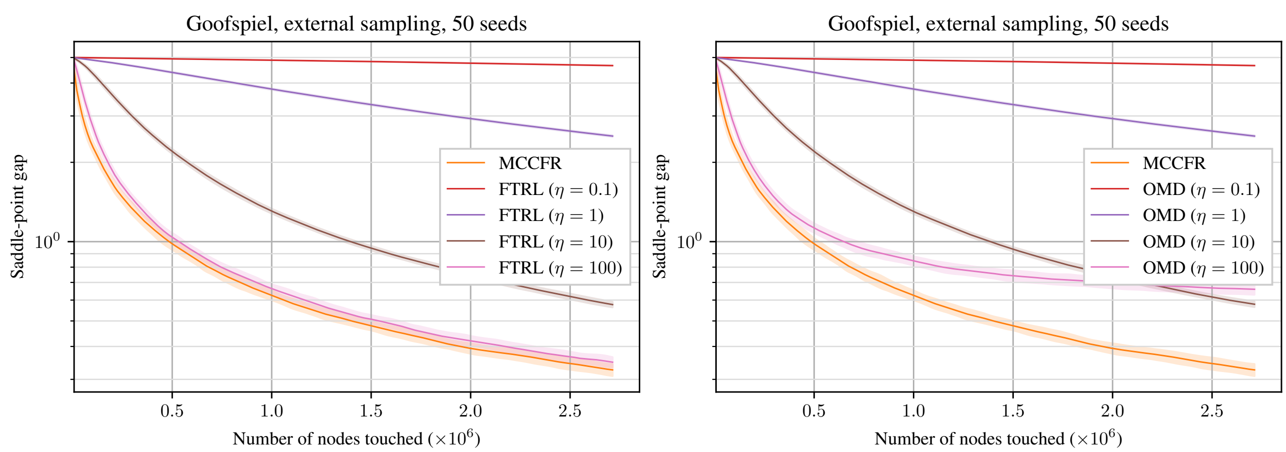

Figure 8. Performance of FTRL and OMD with four stepsizes on Goofspiel with external sampling. MCCFR shown for reference

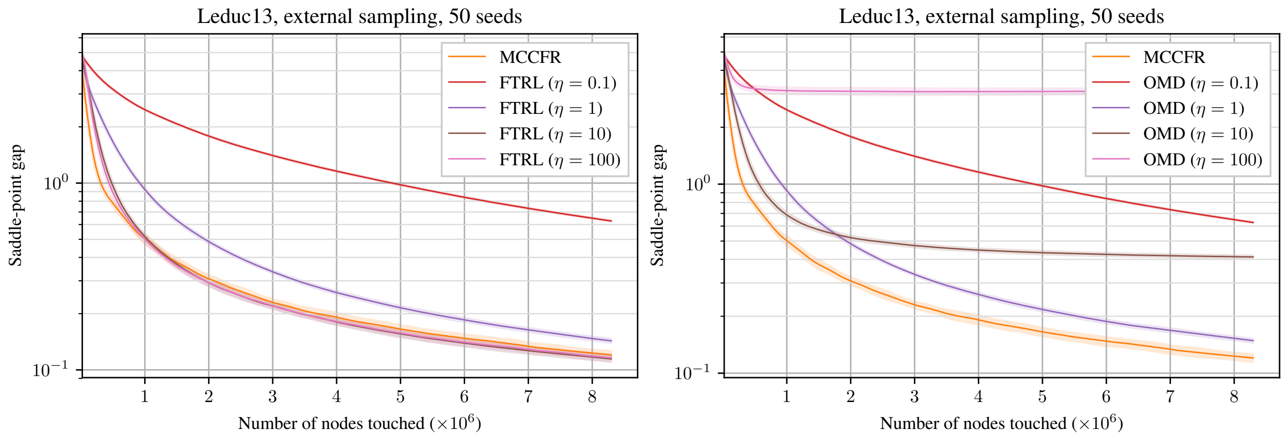

Figure 9. Performance of FTRL and OMD with four stepsizes on Leduc 13 with external sampling. MCCFR shown for reference

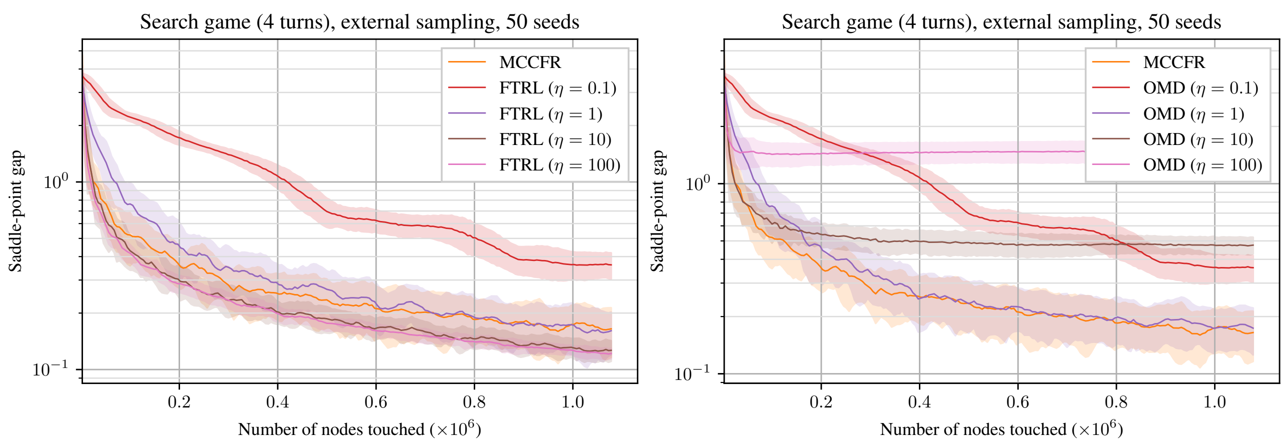

Figure 10. Performance of FTRL and OMD with four stepsizes on Search-4 with external sampling. MCCFR shown for reference

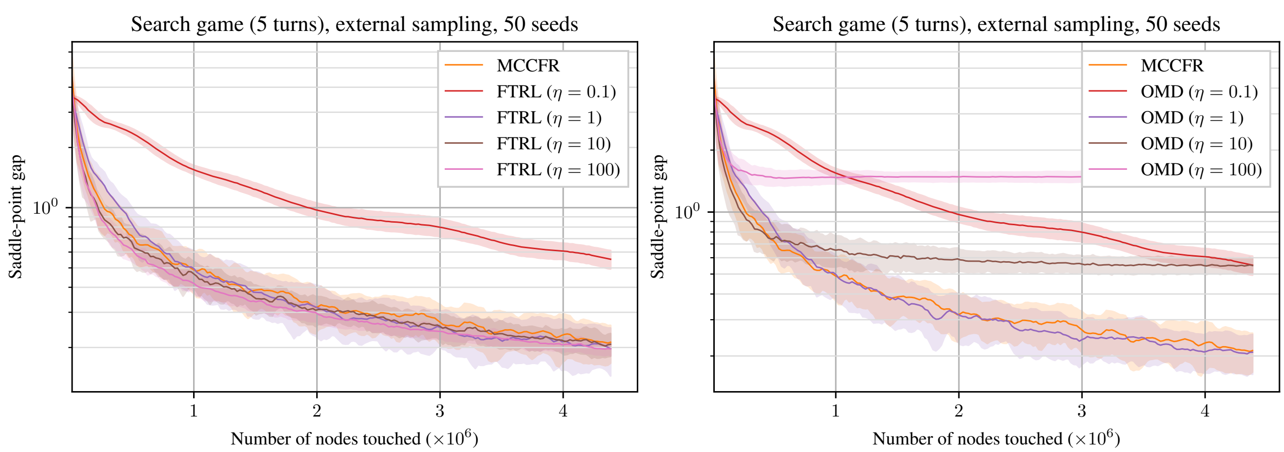

Figure 11. Performance of FTRL and OMD with four stepsizes on Search-5 with external sampling. MCCFR shown for reference

C.2. Exploration-Balanced Outcome Sampling

The Search-4 plot omitted from the main paper is shown here.

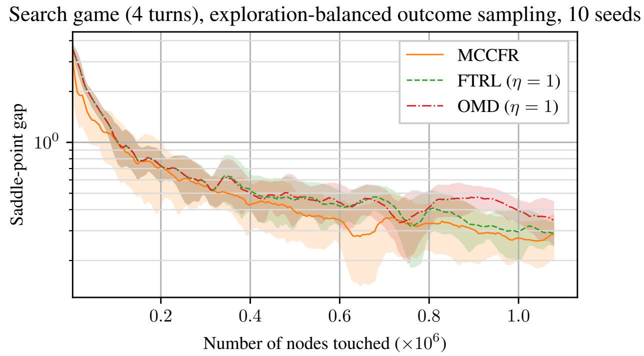

Figure 12. Performance of MCCFR, FTRL, and OMD with outcome sampling on Search-4.

Figure 12 shows the performance on Search-4 and Search-5 with outcome sampling. In Search-4 we find that MCCFR performs better than FTRL and OMD, though FTRL is comparable at later iterations.

Figures 13 through 17 show the performance of FTRL and OMD with outcome sampling for all four stepsizes that we tried on each game: $\eta = 0.1, 1, 10, 100$.

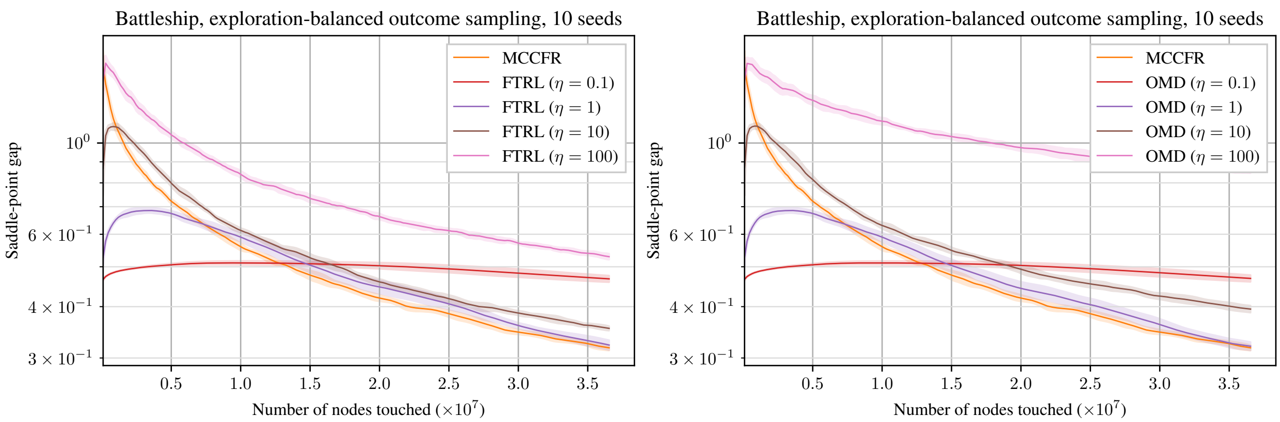

Figure 13. Performance of FTRL and OMD with four stepsizes on Battleship with outcome sampling. MCCFR shown for reference

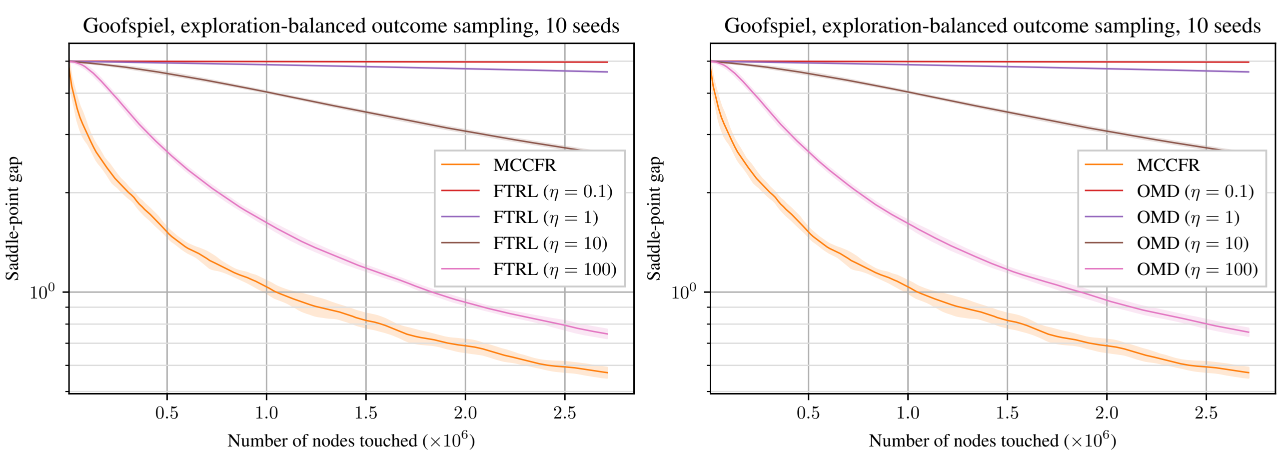

Figure 14. Performance of FTRL and OMD with four stepsizes on Goofspiel with outcome sampling. MCCFR shown for reference

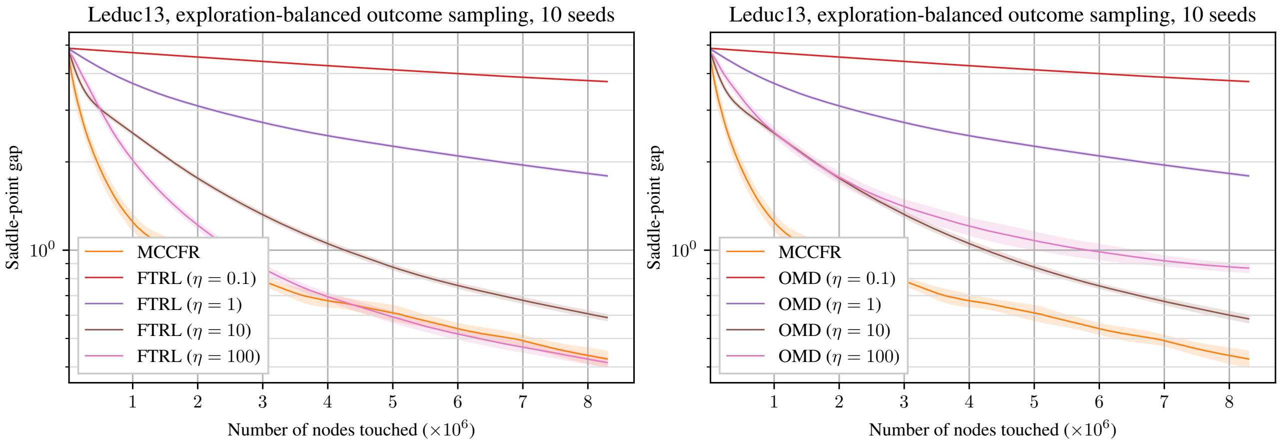

Figure 15. Performance of FTRL and OMD with four stepsizes on Leduc 13 with outcome sampling. MCCFR shown for reference

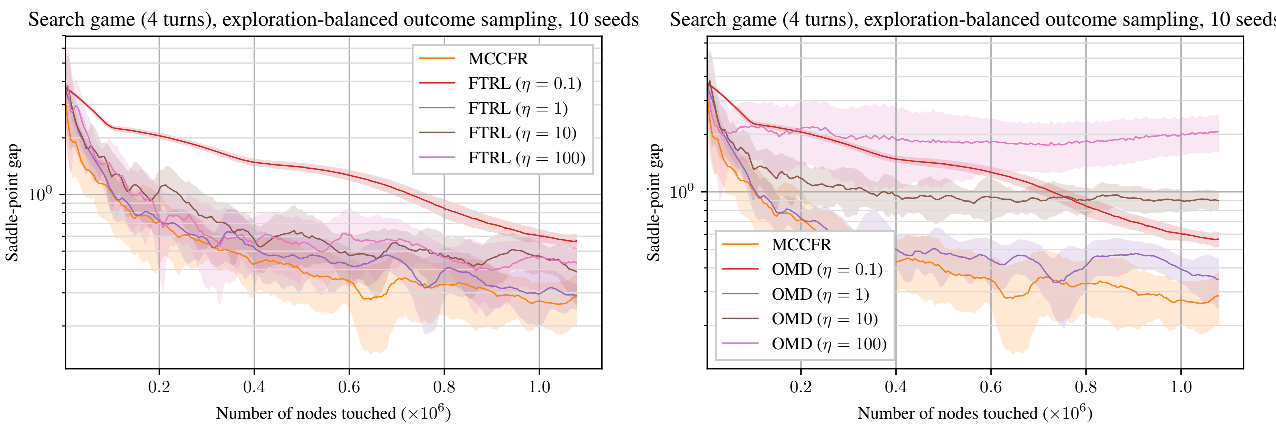

Figure 16. Performance of FTRL and OMD with four stepsizes on Search-4 with outcome sampling. MCCFR shown for reference

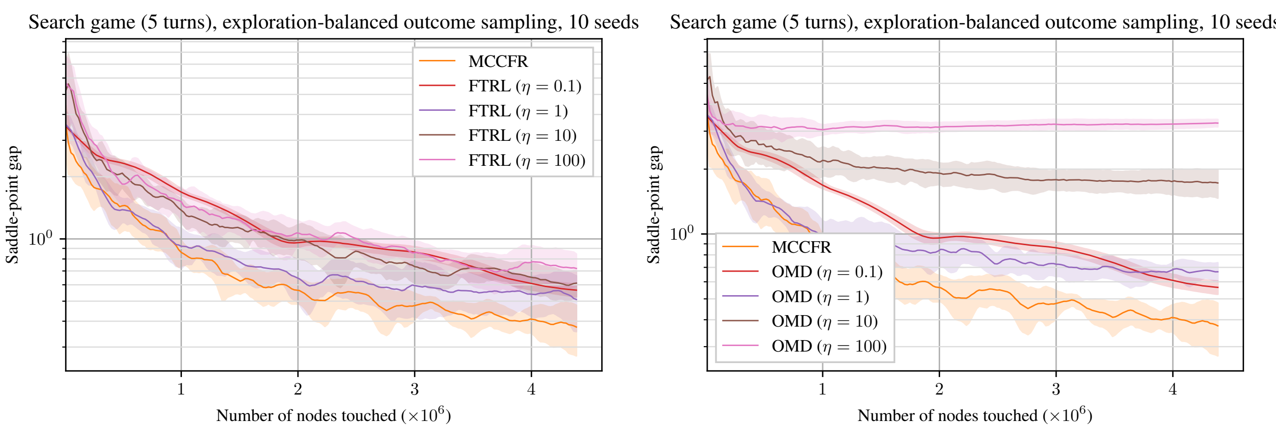

Figure 17. Performance of FTRL and OMD with four stepsizes on Search-5 with outcome sampling. MCCFR shown for reference