Stable-Predictive Optimistic Counterfactual Regret Minimization

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

Abstract

The CFR framework has been a powerful tool for solving large-scale extensive-form games in practice. However, the theoretical rate at which past CFR-based algorithms converge to the Nash equilibrium is on the order of $O(T^{-1/2})$, where T is the number of iterations. In contrast, first-order methods can be used to achieve a $O(T^{-1})$ dependence on iterations, yet these methods have been less successful in practice. In this work we present the first CFR variant that breaks the square-root dependence on iterations. By combining and extending recent advances on predictive and stable regret minimizers for the matrix-game setting we show that it is possible to leverage "optimistic" regret minimizers to achieve a $O(T^{-3/4})$ convergence rate within CFR. This is achieved by introducing a new notion of stable-predictivity, and by setting the stability of each counterfactual regret minimizer relative to its location in the decision tree. Experiments show that this method is faster than the original CFR algorithm, although not as fast as newer variants, in spite of their worst-case $O(T^{-1/2})$ dependence on iterations.

1. Introduction

Counterfactual regret minimization (CFR) (Zinkevich et al., 2007) and later variants such as *Monte-Carlo CFR* (Lanctot et al., 2009), CFR$^+$ (Tammelin et al., 2015), and *Discounted CFR* (Brown & Sandholm, 2019), have been the practical state-of-the-art in solving large-scale zero-sum *extensive-form games* (EFGs) for the last decade. These algorithms were used as an essential ingredient for all recent milestones in the benchmark domain of poker (Bowling et al., 2015; Moravčík et al., 2017; Brown & Sandholm, 2017b). Despite this practical success all known CFR variants have a significant theoretical drawback: their worst-case convergence rate is on the order of $O(T^{-1/2})$, where T is the number of iterations. In contrast to this, there exist first-order methods that converge at a rate of $O(T^{-1})$ (Hoda et al., 2010; Kroer et al., 2015; 2018b). However, these methods have been found to perform worse than newer CFR algorithms such as CFR$^+$, in spite of their theoretical advantage (Kroer et al., 2018b;a).

In this paper we present the first CFR variant which breaks the square-root dependence on the number of iterations. By leveraging recent theoretical breakthroughs on "optimistic" regret minimizers for the matrix-game setting, we show how to set up optimistic counterfactual regret minimizers at each information set such that the overall algorithm retains the properties needed in order to accelerate convergence. In particular, this leads to a *predictive* and *stable* variant of CFR that converges at a rate of $O(T^{-3/4})$.

Typical analysis of regret-minimization leads to a convergence rate of $O(T^{-1/2})$ for solving zero-sum matrix games. However, by leveraging the idea of *optimistic learning* (Chiang et al., 2012; Rakhlin & Sridharan, 2013a;b; Syrgkanis et al., 2015; Wang & Abernethy, 2018), Rakhlin and Sridharan show in a series of papers that it is possible to converge at a rate of $O(T^{-1})$ when leveraging cancellations that occur due to the *optimistic mirror descent* (OMD) algorithm (Rakhlin & Sridharan, 2013a;b). Syrgkanis et al. (2015) build on this idea, and introduce the *optimistic follow-the-regularized-leader* (OFTRL) algorithm; they show that even when the players do not employ the same algorithm, a rate of $O(T^{-3/4})$ can be achieved as long as each algorithm belongs to a class of algorithms that satisfy a stability criterion and leverage predictability of loss inputs. We build on this latter generalization. Because we can only perform the optimistic updates locally with respect to counterfactual regrets we cannot achieve the cancellations that leads to a rate of $O(T^{-1})$; instead we show that by carefully instantiating each counterfactual regret minimizer it is possible to maintain predictability and stability with respect to the overall decision-tree structure, thus leading to a convergence rate of $O(T^{-3/4})$. In order to achieve these results we introduce a

[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>, Noam Brown <noamb@cs.cmu.edu>, Tuomas Sandholm <sandholm@cs.cmu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

new variant of stable-predictivity, and show that each local counterfactual regret minimizer must have its stability set relative to its location in the overall strategy space, with regret minimizers deeper in the decision tree requiring more stability.

In addition to our theoretical results we investigate the practical performance of our algorithm on several poker subgames from the *Libratus* AI which beat top poker professionals (Brown & Sandholm, 2017b). We find that our CFR variant coupled with the OFTRL algorithm and the entropy regularizer leads to better convergence rate than the vanilla CFR algorithm with regret matching, while it does not outperform the newer state-of-the-art algorithm *Discounted CFR* (DCFR) (Brown & Sandholm, 2019). This latter fact is not too surprising, as it has repeatedly been observed that CFR$^+$, and the newer and faster DCFR, converges at a rate *better* than $O(T^{-1})$ for many practical games of interest, in spite of the worst-case rate of $O(T^{-1/2})$.

The reader may wonder why we care about breaking the square-root barrier within the CFR framework. It is well-known that a convergence rate of $O(T^{-1})$ can be achieved outside the CFR framework. As mentioned previously, this can be done with first-order methods such as the *excessive gap technique* (Nesterov, 2005) or *mirror prox* (Nemirovski, 2004) combined with a dilated distance-generating function (Hoda et al., 2010; Kroer et al., 2015; 2018b). Despite this, there has been repeated interest in optimistic regret minimization within the CFR framework, due to the strong practical performance of CFR algorithms. Burch (2017) tries to implement CFR-like features in the context of $O(T^{-1})$ FOMs and regret minimizers, while Brown & Sandholm (2019) experimentally tries optimistic variants of regret minimizers in CFR. We stress that these prior results are only experimental; our results are the first to rigorously incorporate optimistic regret minimization in CFR, and the first to achieve a theoretical speedup.

Notation. Throughout the paper, we use the following notation when dealing with $\mathbb{R}^n$. We use $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ to denote the *dot product* $\boldsymbol{x}^\top \boldsymbol{y}$ of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. We assume that a pair of dual norms $\|\cdot\|, \|\cdot\|_*$ has been chosen. These norms need not be induced by inner products. Common examples of such norm pairs are the ℓ_2 norm which is self dual, and the ℓ_1, ℓ_∞ norms, which are are dual to each other. We will make explicit use of the 2-norm: $\|\boldsymbol{x}\|_2 := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$.

2. Sequential Decision Making and EFG Strategy Spaces

A sequential decision process can be thought of as a tree consisting of two types of nodes: *decision nodes* and *observation nodes*. The set of all decision nodes is denoted as $\mathcal{J}$, and the set of all observation nodes with $\mathcal{K}$. At each decision node $j \in \mathcal{J}$, the agent chooses a strategy from the simplex Δ^{n_j} of all probability distributions over the set A_j of $n_j = |A_j|$ actions available at that decision node. An action is sampled according to the chosen distribution, and the agent then waits to play again. While waiting, the agent might receive a signal (observation) from the process; this possibility is represented with an observation node. At a generic observation point $k \in \mathcal{K}$, the agent might receive n_k signals; the set of signals that the agent can observe is denoted as S_k. The observation node that is reached by the agent after picking action $a \in A_j$ at decision point $j \in \mathcal{J}$ is denoted by $\rho(j, a)$. Likewise, the decision node reached by the agent after observing signal $s \in S_k$ at observation point $k \in \mathcal{K}$ is denoted by $\rho(k, s)$. The set of all observation points reachable from $j \in \mathcal{J}$ is denoted as $\mathcal{C}_j := \{\rho(j, a) : a \in A_j\}$. Similarly, the set of all decision points reachable from $k \in \mathcal{K}$ is denoted as $\mathcal{C}_k := \{\rho(k, s) : s \in S_k\}$. To ease the notation, sometimes we will use the notation $\mathcal{C}_{ja}$ to mean $\mathcal{C}_{\rho(j,a)}$. A concrete example of a decision process is given in the next subsection.

At each decision point $j \in \mathcal{J}$ in a sequential decision process, the decision $\hat{\boldsymbol{x}}_j \in \Delta^{n_j}$ of the agent incurs an (expected) linear loss $\langle \boldsymbol{\ell}_j, \hat{\boldsymbol{x}}_j \rangle$. The expected loss throughout the whole process is therefore $\sum_{j \in \mathcal{J}} \pi_j \langle \boldsymbol{\ell}_j, \hat{\boldsymbol{x}}_j \rangle$, where π_j is the probability of the agent reaching decision point j, defined as the product of the probability with which the agent plays each action on the path from the root of the process to j.

In extensive-form games where all players have *perfect recall* (that is, they never forget about their past moves or their observations), all players face a sequential decision process. The loss vectors $\{\boldsymbol{\ell}_j\}$ are defined based on the strategies of the opponent(s) as well as the chance player. However, as already observed by Farina et al. (2019), sequential decision processes are more general and can model other settings as well, such as POMDPs and MDPs when the decision maker conditions on the entire history of observations and actions.

2.1. Example: Sequential Decision Process for the First Player in Kuhn Poker

As an illustration, consider the game of Kuhn poker (Kuhn, 1950). Kuhn poker consists of a three-card deck: king, queen, and jack. Each player first has to put a payment of 1 into the pot. Each player is then dealt one of the three cards, and the third is put aside unseen. A single round of betting then occurs. The sequential decision process the Player 1 is shown in Figure 1, where $\otimes$ denotes an observation point. In that example, we have: $\mathcal{J} = \{X_0, X_1, X_2, X_3, X_4, X_5, X_6\}$; $n_0 = 1$; $n_j = 2$ for all $j \in \mathcal{J} \setminus \{X_0\}$; $A_{X_0} = \{\text{start}\}$, $A_{X_1} = A_{X_2} = A_{X_3} = \{\text{check}, \text{raise}\}$, $A_{X_4} = A_{X_5} = A_{X_6} = \{\text{fold}, \text{call}\}$; $\mathcal{C}_{\rho(X_0,\text{start})} = \{X_1, X_2, X_3\}$, $\mathcal{C}_{\rho(X_1,\text{raise})} = \emptyset$, $\mathcal{C}_{\rho(X_3,\text{check})} = \{X_6\}$; etc.

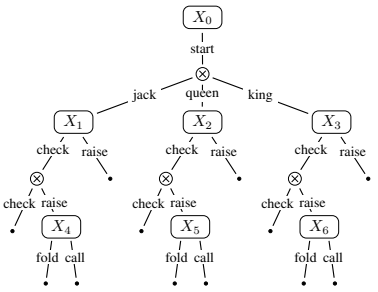

Figure 1. The sequential decision process for the first player in the game of Kuhn poker. $\otimes$ denotes an observation point; small dots represents the end of the decision process.

2.2. Sequence Form for Sequential Decision Processes

The expected loss for a given strategy, as defined in Section 2, is non-linear in the vector of decisions variables $(\hat{\boldsymbol{x}}_j)_{j\in\mathcal{J}}$. This non-linearity is due to the product π_j of probabilities of all actions on the path to from the root to j. We now present a well-known alternative representation of this decision space which preserves linearity.

The alternative formulation is called the *sequence form*. In the sequence-form representation, the simplex strategy space at a generic decision point $j \in \mathcal{J}$ is scaled by the decision variable leading of the last action in the path from the root of the process to j. In this formulation, the value of a particular action represents the probability of playing the whole *sequence* of actions from the root to that action. This allows each term in the expected loss to be weighted only by the sequence ending in the corresponding action. The sequence form has been used to instantiate linear programming (von Stengel, 1996) and first-order methods (Hoda et al., 2010; Kroer et al., 2015; 2018b) for computing Nash equilibria of zero-sum EFGs. There is a straightforward mapping between a vector of decisions $(\hat{\boldsymbol{x}}_j)_{j\in\mathcal{J}}$, one for each decision point, and its corresponding sequence form: simply assign each sequence the product of probabilities in the sequence. We will let $X^\triangle$ denote the sequence-form representation of a vector of decisions $(\hat{\boldsymbol{x}}_j)_{j\in\mathcal{J}}$. Likewise, going from a sequence-form strategy $\boldsymbol{x}^\triangle \in X^\triangle$ to a corresponding vector of decisions $(\hat{\boldsymbol{x}}_j)_{j\in\mathcal{J}}$ can be done by dividing each entry (sequence) in $\boldsymbol{x}^\triangle$ by the value $\boldsymbol{x}^\triangle_{p_j}$ where p_j is the entry in $\boldsymbol{x}^\triangle$ corresponding to the unique last action that the agent took before reaching j.

Formally, the sequence-form representation $X^\triangle$ of a sequential decision process can be obtained recursively, as follows:

- At every observation point $k \in \mathcal{K}$, we let
$$X^\triangle_k := X^\triangle_{j_1} \times X^\triangle_{j_2} \times \cdots \times X^\triangle_{j_{n_k}}, \tag{1}$$
where $\{j_1, j_2, \ldots, j_{n_k}\} = \mathcal{C}_k$ are the children decision points of k.
- At every decision point $j \in \mathcal{J}$, we let
$$X^\triangle_j := \Big\{(\lambda_1, \ldots, \lambda_{n_j}, \lambda_1 \boldsymbol{x}_{k_1}, \ldots, \lambda_{n_j}\boldsymbol{x}_{k_{n_j}}) : (\lambda_1, \ldots, \lambda_n) \in \Delta^{n_j}, \boldsymbol{x}_{k_1} \in X^\triangle_{k_1}, \boldsymbol{x}_{k_2} \in X^\triangle_{k_2}, \ldots, \boldsymbol{x}_{k_{n_j}} \in X^\triangle_{k_{n_j}}\Big\}, \tag{2}$$
where $\{k_1, k_2, \ldots, k_{n_j}\} = \mathcal{C}_j$ are the children observation points of j.

The sequence form strategy space for the whole sequential decision process is then $X^\triangle_r$, where r is the root of the process. Crucially, $X^\triangle$ is a convex and compact set, and the expected loss of the process is a linear function over $X^\triangle$.

With the sequence-form representation the problem of computing a Nash equilibriun in an EFG can be formulated as a *bilinear saddle-point problem* (BSPP). A BSPP has the form
$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}, \tag{3}$$
where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact sets. In the case of extensive-form games, $\mathcal{X} = X^\triangle$ and $\mathcal{Y} = Y^\triangle$ are the sequence-form strategy spaces of the sequential decision processes faced by the two players, and A is a sparse matrix encoding the leaf payoffs of the game.

2.3. Notation when Dealing with the Extensive Form

In the rest of the paper, we will make heavy use of the sequence form and its inductive construction given in (12) and (13). We will consistently denote sequence-form strategies with a triangle superscript. As we have already observed, vectors that pertain to the sequence-form have one entry for each sequence of the decision process, that is one entry for pair (j, a) where $j \in \mathcal{J}, a \in A_j$. Sometimes, we will need to *slice* a vector v and isolate only those entries that refer to all decision points j' and actions $a' \in A_{j'}$ that are at or below some $j \in \mathcal{J}$; we will denote such operation as $[\boldsymbol{v}]_{\downarrow j}$. Similarly, we introduce the syntax $[\boldsymbol{v}]_j$ to denote the subset of $n_j = |A_j|$ entries of v that pertain to all actions $a \in A_j$ at decision point $j \in \mathcal{J}$.

3. Stable-Predictive Regret Minimizers

In this paper, we operate within the online learning framework called *online convex optimization* (Zinkevich, 2003). In particular, we restrict our attention to a modern subtopic: *predictive* (also often called *optimistic*) regret minimization (Chiang et al., 2012; Rakhlin & Sridharan, 2013a;b).

As usual in this setting, a decision maker repeatedly plays against an unknown environment by making a sequence of decisions $\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots \in \mathcal{X} \subseteq \mathbb{R}^n$, where the set $\mathcal{X}$ of feasible decisions for the decision maker is convex and compact. The evaluation of the outcome of each decision $\boldsymbol{x}^t$ is $\langle \boldsymbol{\ell}^t, \boldsymbol{x}^t \rangle$, where $\boldsymbol{\ell}^t \in \mathcal{X}$ is a convex *loss vector*, unknown

to the decision maker until after the decision is made. The peculiarity of *predictive* regret minimization is that we also assume that the decision maker has access to *predictions* $\boldsymbol{m}^1, \boldsymbol{m}^2, \dots$ of what the loss vectors $\boldsymbol{\ell}^1, \boldsymbol{\ell}^2, \dots$ will be. In summary, by *predictive regret minimizer* we mean a device that supports the following two operations:

- it provides the next decision $\boldsymbol{x}^{t+1} \in \mathcal{X}$ given a prediction $\boldsymbol{m}^{t+1}$ of the next loss vector and
- it receives/observes the convex loss vectors $\boldsymbol{\ell}^t$ used to evaluate decision $\boldsymbol{x}^t$.

The learning is *online* in the sense that the decision maker's (that is, device's) next decision, $\boldsymbol{x}^{t+1}$, is based only on the previous decisions $\boldsymbol{x}^1, \dots, \boldsymbol{x}^t$, observed loss vectors $\boldsymbol{\ell}^1, \dots, \boldsymbol{\ell}^t$, and the prediction of the past loss vectors as well as the next one $\boldsymbol{m}^1, \dots, \boldsymbol{m}^{t+1}$.

Just as in the case of a regular (that is, non-predictive) regret minimizer, the quality metric for the predictive regret minimizer is its *cumulative regret*, which is the difference between the loss cumulated by the sequence of decisions $\boldsymbol{x}^1, \dots, \boldsymbol{x}^T$ and the loss that would have been cumulated by playing the best-in-hindsight time-independent decision $\tilde{\boldsymbol{x}}$. Formally, the cumulative regret up to time T is

$$R^T := \sum_{t=1}^{T} \langle \boldsymbol{\ell}^t, \boldsymbol{x}^t \rangle - \min_{\tilde{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} \langle \boldsymbol{\ell}^t, \tilde{\boldsymbol{x}} \rangle \right\}. \tag{4}$$

We introduce a new class of predictive regret minimizers whose cumulative regret decomposes into a constant term plus a measure of the prediction quality, while maintaining stability in the sense that the iterates $\boldsymbol{x}^1, \dots, \boldsymbol{x}^T$ change slowly.

Definition 1 (Stable-predictive regret minimizer)**.** *A predictive regret minimizer is (κ, α, β)-stable-predictive if the following two conditions are met:*

- Stability. *The decisions produced change slowly:*
$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\| \leq \kappa \quad \forall\, t \geq 1. \tag{5}$$
- Prediction bound. *For all T, the cumulative regret up to time T is bounded according to*
$$R^T \leq \frac{\alpha}{\kappa} + \beta\kappa \sum_{t=1}^{T} \|\boldsymbol{\ell}^t - \boldsymbol{m}^t\|_*^2. \tag{6}$$
In other words, small prediction errors only minimally affect the regret accumulated by the device. If, in particular, the prediction $\boldsymbol{m}^t$ matches the loss vector $\boldsymbol{\ell}^t$ perfectly for all t, the cumulative regret remains asymptotically constant.

Our notion of stable-predictivity is similar to the *Regret bounded by Variation in Utilities* (RVU) property given by Syrgkanis et al. (2015), which asserts that

$$R^T \leq \alpha' + \beta' \sum_{t=1}^{T} \|\boldsymbol{\ell}^t - \boldsymbol{\ell}^{t-1}\|_*^2 - \gamma' \sum_{t=1}^{T} \|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2. \tag{RVU}$$

However, there are several important differences:

- Syrgkanis et al. (2015) assume that $\boldsymbol{m}^t = \boldsymbol{\ell}^{t-1}$; this explains the term $\|\boldsymbol{\ell}^t - \boldsymbol{\ell}^{t-1}\|_*^2$ in (RVU) instead of $\|\boldsymbol{\ell}^t - \boldsymbol{m}^t\|_*^2$ in (6). One of the reason why we do not make assumptions on $\boldsymbol{m}^t$ is that, unlike in matrix games, we will need to use modified predictions for each local regret minimizer, since we need to predict the local counterfactual loss.
- Our notion ignores the cancellation term $-\gamma' \sum \|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2$; instead, we require the *stabilty* property (5).
- The coefficients in the regret bound (6) are forced to be inversely proportional, and tied to the choice of the stability parameter κ. Syrgkanis et al. (2015) show that same correlation holds for the optimistic follow-the-regularized leader, but they don't require it in their definition of the RVU property.

Syrgkanis et al. (2015) show that their optimistic follow-the-regularized-leader (OFTRL) algorithm, as well as the variant of the mirror descent algorithm presented by Rakhlin & Sridharan (2013a), satisfy (RVU). In Section 3.2 we show that OFTRL also satisfies stable-predictivity.

3.1. Relationship with Bilinear Saddle-Point Problems

In this subsection we show how stable-predictive regret minimization can be used to solve a BSPP such as a Nash equilibrium problem in two-player zero-sum extensive-form games with perfect recall (Sections 2 and 2.2). The solutions of (3) are called *saddle points*. The *saddle-point residual* (or *gap*) ξ of a point $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \in \mathcal{X} \times \mathcal{Y}$, defined as

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

measures how close $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is to being a saddle point (the lower the residual, the closer).

It is known that regular (non-predictive) regret minimization yields an anytime algorithm that produces a sequence of points $(\bar{\boldsymbol{x}}^T, \bar{\boldsymbol{y}}^T) \in \mathcal{X} \times \mathcal{Y}$ whose residuals are $\xi^T = O(T^{-1/2})$. Syrgkanis et al. (2015) observe that in the context of matrix games (i.e., when $\mathcal{X}$ and $\mathcal{Y}$ are simplexes), RVU minimizers that also satisfy the stability condition (5) can be used in place of regular regret minimizers to improve the convergence rate to $O(T^{-3/4})$. In what follows, we show how to extend the argument to stable-predictive regret minimizers and general bilinear saddle-point problems beyond Nash equilibria in two-player zero-sum matrix games.

A folk theorem explains the tight connections between low regret and low residual (Cesa-Bianchi & Lugosi, 2006). Specifically, by setting up two regret minimizers (one for $\mathcal{X}$ and one for $\mathcal{Y}$) that observe loss vectors given by $\boldsymbol{\ell}_{\mathcal{X}}^t :=$

$-\boldsymbol{A}\boldsymbol{y}^t, \boldsymbol{\ell}_{\mathcal{Y}}^t := \boldsymbol{A}^\top \boldsymbol{x}^t$, the profile of average decisions

$$\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}^t, \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}^t\right) \in \mathcal{X}\times\mathcal{Y} \tag{7}$$

has residual ξ bounded from above according to

$$\xi \le \frac{1}{T}(R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T).$$

Hence, by letting the predictions be defined as $\boldsymbol{m}_{\mathcal{X}}^t := \boldsymbol{\ell}_{\mathcal{X}}^{t-1}, \boldsymbol{m}_{\mathcal{Y}}^t := \boldsymbol{\ell}_{\mathcal{Y}}^{t-1}$, and assuming that the predictive regret minimizers are (κ, α, β)-stable-predictive, we obtain that the residual ξ of the average decisions (7) satisfies

$$\begin{aligned} T\xi &\le \frac{2\alpha}{\kappa} + \beta\kappa\sum_{t=1}^{T}\|-\boldsymbol{A}\boldsymbol{y}^t + \boldsymbol{A}\boldsymbol{y}^{t-1}\|_*^2 \\ &\qquad + \beta\kappa\sum_{t=1}^{T}\|\boldsymbol{A}^\top\boldsymbol{x}^t - \boldsymbol{A}^\top\boldsymbol{x}^{t-1}\|_*^2 \\ &\le \frac{2\alpha}{\kappa} + \beta\|\boldsymbol{A}\|_{\text{op}}^2\kappa\left(\sum_{t=1}^{T}\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2 + \sum_{t=1}^{T}\|\boldsymbol{y}^t - \boldsymbol{y}^{t-1}\|^2\right) \\ &\le \frac{2\alpha}{\kappa} + 2\beta T\|\boldsymbol{A}\|_{\text{op}}^2\kappa^3, \end{aligned}$$

where the first inequality holds by (6), the second by noting that the operator norm $\|\cdot\|_{\text{op}}$ of a linear function is equal to the operator norm of its transpose, and the third inequality by the stability condition (5). This shows that if the stability parameter κ of the two stable-predictive regret minimizers is $\Theta(T^{-1/4})$, then the saddle point residual is $\xi = O(T^{-3/4})$, an improvement over the bound $\xi = O(T^{-1/2})$ obtained with regular (that is, non-predictive) regret minimizers.

3.2. Optimistic Follow the Regularized Leader

Optimistic follow-the-regularized-leader (OFTRL) is a regret minimizer introduced by Syrgkanis et al. (2015). At each time t, OFTRL outputs the decision

$$\boldsymbol{x}^t = \operatorname*{argmin}_{\tilde{\boldsymbol{x}}\in\mathcal{X}}\left\{\left\langle \tilde{\boldsymbol{x}}, \boldsymbol{m}^t + \sum_{t=1}^{T-1}\boldsymbol{\ell}^t\right\rangle + \frac{1}{\eta}R(\tilde{\boldsymbol{x}})\right\}, \tag{8}$$

where $\eta > 0$ is a free constant and $R(\cdot)$ is a 1-strongly convex regularizer with respect to the norm $\|\cdot\|$. Furthermore, let $\Delta_R := \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{X}}\{R(\boldsymbol{x}) - R(\boldsymbol{y})\}$ denote the diameter of the range of R, and let $\Delta_\ell := \max_t \max\{\|\boldsymbol{\ell}^t\|_*, \|\boldsymbol{m}^t\|_*\}$ be the maximum (dual) norm of any loss vector or prediction thereof.

A theorem similar to that of Syrgkanis et al. (2015, Proposition 7), which was obtained in the context of the RVU property, can be shown for the stable-predictive framework:

Theorem 1. *OFTRL is a $3\Delta_\ell(\eta, \Delta_R, 1)$-stable-predictive regret minimizer.*

We give a proof of Theorem 1 the appendix. When the loss vectors are further assumed to be non-negative, it can be shown that OFTRL is $2\Delta_\ell(\eta, \Delta_R, 1)$-stable-predictive, where we have substituted a factor of 2 rather than the factor of 3 in Theorem 1.

4. CFR as Regret Decomposition

In this section we offer some insights into CFR, and discuss what changes need to be made in order to leverage the power of predictive regret minimization. CFR is a framework for constructing a (non-predictive) regret minimizer $\mathcal{R}^\triangle$ that operates over the sequence-form strategy space $X^\triangle$ of a sequential decision process. In accordance with Section 2.3, we denote the decision produced by $\mathcal{R}^\triangle$ at time t as $\boldsymbol{x}^{\triangle,t}$; the corresponding loss functions is denoted as $\boldsymbol{\ell}^{\triangle,t}$.

One central idea in CFR is to define a localized notion of loss: for all $j \in \mathcal{J}$, CFR constructs the following linear *counterfactual loss function* $\hat{\boldsymbol{\ell}}_j^{t,\circ} : \Delta^{n_j} \to \mathbb{R}$. Intuitively, the counterfactual loss $\hat{\boldsymbol{\ell}}_j^{t,\circ}(\boldsymbol{x}_j)$ of a *local* strategy $\boldsymbol{x}_j \in \Delta^{n_j}$ measures the loss that the agent would face were the agent allowed to change the strategy at decision point j *only*. In particular, $\hat{\boldsymbol{\ell}}_j^{t,\circ}(\boldsymbol{x}_j)$ is the loss of an agent that follows the strategy $\boldsymbol{x}_j$ instead of $\boldsymbol{x}^{\triangle,t}$ at decision point j, but otherwise follows the strategy $\boldsymbol{x}^{\triangle,t}$ everywhere else. Formally,

$$\begin{aligned} \hat{\boldsymbol{\ell}}_j^{t,\circ} : \boldsymbol{x}_j = (x_{ja_1}, \dots x_{ja_{n_j}}) &\mapsto \langle [\boldsymbol{\ell}^{\triangle,t}]_j, \boldsymbol{x}_j\rangle \\ &\quad + \sum_{a\in A_j}\left(x_{ja}\sum_{j'\in\mathcal{C}_{ja}}\langle [\boldsymbol{\ell}^{\triangle,t}]_{\downarrow j'}, [\boldsymbol{x}^{\triangle,t}]_{\downarrow j'}\rangle\right). \end{aligned} \tag{9}$$

Since $\hat{\boldsymbol{\ell}}_j^{t,\circ}$ is a linear function, it has a unique representation as a *counterfactual loss vector* $\hat{\boldsymbol{\ell}}_j^t$, defined as

$$\hat{\boldsymbol{\ell}}_j^{t,\circ}(\boldsymbol{x}_j) = \langle \hat{\boldsymbol{\ell}}_j^t, \boldsymbol{x}_j\rangle \quad \forall\, \boldsymbol{x}_j \in \Delta^{n_j}. \tag{10}$$

With this local notion of loss function, a corresponding local notion of regret for a sequence of decisions $\hat{\boldsymbol{x}}_j^1, \dots, \hat{\boldsymbol{x}}_j^T$, called the *counterfactual regret*, is defined for each decision point $j \in \mathcal{J}$:

$$\hat{R}_j^T := \sum_{t=1}^{T}\langle \hat{\boldsymbol{\ell}}_j^t, \hat{\boldsymbol{x}}_j^t\rangle - \min_{\tilde{\boldsymbol{x}}_j\in\Delta^{n_j}}\sum_{t=1}^{T}\langle \hat{\boldsymbol{\ell}}_j^t, \tilde{\boldsymbol{x}}_j\rangle.$$

Intuitively, $\hat{R}_j^T$ represents the difference between the loss that was suffered for picking $\hat{\boldsymbol{x}}_j^t \in \Delta^{n_j}$ and the minimum loss that could be secured by choosing a different strategy *at decision point j only*. This is conceptually different from the definition of regret of $\mathcal{R}^\triangle$, which instead measures the difference between the loss suffered and the best loss that could have been obtained, in hindsight, by picking *any* strategy from the whole strategy space, with no extra constraints.

With this notion of regret, CFR instantiates one (non-stable-predictive) regret minimizer $\hat{\mathcal{R}}_j$ for each decision point $j \in \mathcal{J}$. Each local regret minimizer $\hat{\mathcal{R}}_j$ operates on the domain Δ^{n_j}, that is, the space of strategies at decision point j only. At each time t, $\mathcal{R}^\triangle$ prescribes the strategy that, at each information set j, behaves according to the decision of $\hat{\mathcal{R}}_j$. Similarly, any loss vector $\boldsymbol{\ell}^{\triangle,t}$ input to $\mathcal{R}^\triangle$ is processed as follows: (i) first, the counterfactual loss vectors

$\{\hat{\boldsymbol{\ell}}_j^t\}_{j\in\mathcal{J}}$, one for each decision point $j \in \mathcal{J}$, are computed; (ii) then, each $\hat{\mathcal{R}}_j$ observes its corresponding counterfactual loss vector $\hat{\boldsymbol{\ell}}_j^t$.

Another way to look at CFR and counterfactual losses is as an inductive construction over subtrees. When a loss function relative to the whole sequential decision process is received by the root node, inductively each node of the sequential decision process does the following:

- If the node receiving the loss vector is an observation node, the incoming loss vector is partitioned and forwarded to each child decision node. The partition of the loss vector is done so as to ensure that only entries relevant to each subtree are received down the tree.
- If the node receiving the loss vector is a decision node, the incoming loss vector is first forwarded as-is to each of the child observation points, and then it is used to construct the counterfactual loss vector $\hat{\boldsymbol{\ell}}_j^t$ which is input into $\hat{\mathcal{R}}_j$.

This alternative point of view differs from the original one, but has been recently used by Farina et al. (2018; 2019) to simplify the analysis of the algorithm. When viewed from the above point of view, CFR is recursively building—in a bottom-up fashion—regret minimizers for each subtree starting from child subtrees.

In accordance with our convention (Section 2.3), we denote $\mathcal{R}_v^\triangle$, for $v \in \mathcal{J}\cup\mathcal{K}$, the regret minimizer that operates on $X_v^\triangle$ obtained by only considering the local regret minimizers in the subtree rooted at vertex v of the sequential decision process. Analogously, we will denote with $R_v^{\triangle,T}$ the regret of $\mathcal{R}_v^\triangle$ up to time T, and with $\boldsymbol{\ell}_v^{\triangle,t}$ the loss function entering $\mathcal{R}_v^\triangle$ at time t. In accordance with the above construction, we have that

$$\boldsymbol{\ell}_k^{\triangle,t} = [\boldsymbol{\ell}_j^{\triangle,t}]_{\downarrow k} \;\forall k \in \mathcal{C}_j,\;\; \boldsymbol{\ell}_j^{\triangle,t} = [\boldsymbol{\ell}_k^{\triangle,t}]_{\downarrow j} \;\forall j \in \mathcal{C}_k. \quad (11)$$

Finally, we denote the decisions produced by $\mathcal{R}_v^\triangle$ at time t as $\boldsymbol{x}_v^{\triangle,t}$. As per our discussion above, the decisions produced by $\mathcal{R}^\triangle$ are tied together inductively according to

$$\boldsymbol{x}_k^{\triangle,t} = (\boldsymbol{x}_{j_1}^{\triangle,t}, \dots, \boldsymbol{x}_{j_{n_k}}^{\triangle,t}) \quad \forall k \in \mathcal{K}, \quad (12)$$

where $\{j_1, \dots, j_{n_k}\} = \mathcal{C}_k$, and

$$\boldsymbol{x}_j^{\triangle,t} = \left(\hat{\boldsymbol{x}}_j^t, \hat{\boldsymbol{x}}_{ja_1}^t \boldsymbol{x}_{\rho(j,a_1)}^{\triangle,t}, ..., \hat{\boldsymbol{x}}_{ja_{n_j}}^t \boldsymbol{x}_{\rho(j,a_{n_j})}^{\triangle,t}\right) \;\forall j \in \mathcal{J}, \quad (13)$$

where $\{a_1, \dots, a_{n_j}\} = A_j$. The following two lemmas can be easily extracted from Farina et al. (2018). A proof is presented in the appendix.

Lemma 1. *For all $k \in \mathcal{K}$, $R_k^{\triangle,T} = \sum_{j\in\mathcal{C}_k} R_j^{\triangle,T}$.*

Lemma 2. *For all $j \in \mathcal{J}$, $R_j^{\triangle,T} \le \hat{R}_j^T + \max_{k\in\mathcal{C}_j} R_k^{\triangle,T}$.*

The two lemmas above do not make any assumption about the nature of the (localized) regret minimizers $\hat{\mathcal{R}}_j$, and therefore they are applicable even when the $\hat{\mathcal{R}}_j$ are predictive or, specifically, stable-predictive.

5. Stable-Predictive Counterfactual Regret Minimization

Our proposed algorithm behaves exactly like CFR, with the notable difference that our local regret minimizers $\hat{\mathcal{R}}_j$ are stable-predictive and chosen to have specific stability parameters. Furthermore, the predictions m_j^t for each local regret minimizer $\hat{\mathcal{R}}_j$ are chosen so as to leverage the predictivity property of the regret minimizers. Given a desired value of $\kappa^* > 0$, by choosing the stability parameters and predictions as we will detail later, we can guarantee that $\mathcal{R}^\triangle$ is a $(\kappa^*, O(1), O(1))$-stable-predictive regret minimizer.[1]

5.1. Choice of Stability Parameters

We use the following scheme to pick the stability parameter of $\hat{\mathcal{R}}_j$. First, we associate a scalar γ_v to each node $v \in \mathcal{J}\cup\mathcal{K}$ of the sequential decision process. The value γ_r of the root decision node is set to κ^*, and the value for each other node v is set relative to the value γ_u of their parent

$$\gamma_v := \frac{\gamma_u}{2\sqrt{n_u}} \text{ if } u \in \mathcal{J}, \quad \gamma_v := \frac{\gamma_u}{\sqrt{n_u}} \text{ if } u \in \mathcal{K}. \quad (14)$$

The stability parameter of each decision point $j \in \mathcal{J}$ is

$$\kappa_j := \frac{\gamma_j}{2\sqrt{n_j} B_j}, \quad (15)$$

where B_j is an upper bound on the 2-norm of any vector in $X_j^\triangle$. A suitable value of B_j can be found by recursively using the following rules: for all $k \in \mathcal{K}$ and $j \in \mathcal{J}$,

$$B_k = \sqrt{\sum_{j'\in\mathcal{C}_k} B_{j'}^2}, \quad B_j = \sqrt{1 + \max_{k'\in\mathcal{C}_j} B_{k'}^2} \quad (16)$$

At each decision point j, any stable-predictive regret minimizer that is able to guarantee the above stability parameter can be used. For example, one can use OFTRL where the stepsize η is chosen appropriately. As an examples, assuming that all loss vectors involved have (dual) norm bounded by $1/3$, we can simply set the stepsize η of the local OFTRL regret minimizer $\hat{\mathcal{R}}_j$ at decision point j to be $\eta = \kappa_j$.

5.2. Prediction of Counterfactual Loss Vectors

Let $\boldsymbol{m}^{\triangle,t}$ be the prediction received by $\mathcal{R}^\triangle$, concerning the future loss vector $\boldsymbol{\ell}^{\triangle,t}$. We will show how to process the prediction and produce *counterfactual* prediction vectors $\hat{\boldsymbol{m}}_j^t$ (one for each decision point $j \in \mathcal{J}$) for each local stable-predictive regret minimizer $\hat{\mathcal{R}}_j$.

[1] Throughout the paper, our asymptotic notation is always with respect to the number of iterations T.

Following the construction of the counterfactual loss functions defined in (9), for each decision point $j \in \mathcal{J}$ we define the *counterfactual prediction function* $\hat{\boldsymbol{m}}_j^{t,\circ} : \Delta^{n_j} \to \mathbb{R}$ as

$$\hat{\boldsymbol{m}}_j^{t,\circ} : \Delta^{n_j} \ni \boldsymbol{x}_j = (x_{ja_1}, \dots, x_{ja_{n_j}}) \mapsto \left\langle [\boldsymbol{m}^{\triangle,t}]_j, \boldsymbol{x}_j \right\rangle + \sum_{a \in A_j} \left(x_{ja} \sum_{j' \in \mathcal{C}_{ja}} \left\langle [\boldsymbol{m}^{\triangle,t}]_{\downarrow j'}, [\boldsymbol{x}^{\triangle,t}]_{\downarrow j'} \right\rangle \right).$$

Observation. *It important to observe that the counterfactual prediction function $\hat{\boldsymbol{m}}_j^t$ depends on the decisions produced at time t in the subtree rooted at j. In other words, in order to construct the prediction for what loss $\hat{\mathcal{R}}_j$ will observe* after *producing the decision $\boldsymbol{x}_j^t$, we use the "future" decisions $\boldsymbol{x}_{ja}^t$ from the subtrees below $j \in J$.*

Similarly to what is done for the counterfactual loss function, we define the *counterfactual loss prediction vector* $\hat{\boldsymbol{m}}_j^t$, as the (unique) vector in $\mathbb{R}_j^n$ such that

$$\hat{\boldsymbol{m}}_j^{t,\circ}(\boldsymbol{x}_j) = \langle \hat{\boldsymbol{m}}_j^t, \boldsymbol{x}_j \rangle \quad \forall\, \boldsymbol{x}_j \in \Delta^{n_j}. \tag{17}$$

5.3. Proof of Correctness

We will prove that our choice of stability parameters (14) and (localized) counterfactual loss predictions (17) guarantee that $\mathcal{R}^{\triangle}$ is a $(\kappa^*, O(1), O(1))$-stable-predictive regret minimizer. Our proof is by induction on the sequential decision process structure: we prove that our choices yield a $(\gamma_v, O(1), O(1))$-stable-predictive regret minimizer in the sub-sequential decision process rooted at each possible node $v \in \mathcal{J} \cup \mathcal{K}$. For observation nodes $v \in \mathcal{K}$ the inductive step is performed via Lemma 3, while for decision nodes $v \in \mathcal{J}$ the inductive step is performed via Lemma 4. The proof of both lemmas can be found in the appendix.

Lemma 3. *Let $k \in \mathcal{K}$ be an observation node, and assume that $\mathcal{R}_j^{\triangle}$ is a $(\gamma_j, O(1), O(1))$-stable-predictive regret minimizer over the sequence-form strategy space $X_j^{\triangle}$ for each $j \in \mathcal{C}_k$. Then, $\mathcal{R}_k^{\triangle}$ is a $(\gamma_k, O(1), O(1))$-stable-predictive regret minimizer over the sequence-form strategy space $X_k^{\triangle}$.*

Lemma 4. *Let $j \in \mathcal{J}$ be a decision node, and assume that $\mathcal{R}_k^{\triangle}$ is a $(\gamma_k, O(1), O(1))$-stable-predictive regret minimizer over the sequence-form strategy space $X_k^{\triangle}$ for each $k \in \mathcal{C}_j$. Suppose further that $\hat{\mathcal{R}}_j$ is a $(\kappa_j, O(1), O(1))$-stable-predictive regret minimizer over the simplex Δ^{n_j}. Then, $\mathcal{R}_j^{\triangle}$ is a $(\gamma_k, O(1), O(1))$-stable-predictive regret minimizer over the sequence-form strategy space $X_j^{\triangle}$.*

Putting together Lemma 3 and Lemma 4, and using induction on the sequential decision process structure, we obtain the following formal statement.

Corollary 1. *Let $\kappa^* > 0$. If:*

1. *Each localized regret minimizer $\hat{\mathcal{R}}_j$ is $(\kappa_j, O(1), O(1))$-stable-predictive and produces decisions over the local (simplex) action space Δ^{n_j}, where κ_j is as in* (15)*; and*
2. *$\hat{\mathcal{R}}_j$ observes the counterfactual loss prediction $\hat{\boldsymbol{m}}_j^t$ as defined in* (17)*; and*
3. *$\hat{\mathcal{R}}_j$ observes the counterfactual loss vectors $\hat{\boldsymbol{\ell}}_j^t$ as defined in* (10)*,*

then $\mathcal{R}^{\triangle}$ is a $(\kappa^*, O(1), O(1))$-stable-predictive regret minimizer that acts over the sequence-form strategy space $\tilde{X}$.

By combining the above result with the arguments of Section 3.1, we conclude that by constructing two $(\Theta(T^{1/4}), O(1), O(1))$-stable-predictive regret minimizers, one per player, using the construction above, we obtain an algorithm that can approximate a Nash equilibrium and at time T the average strategy produces an $O(T^{-3/4})$-Nash equilibrium in a two-player zero-sum game.

6. Experiments

Our techniques are evaluated in the benchmark domain of heads-up no-limit Texas hold'em poker (HUNL) subgames. In HUNL, two players P_1 and P_2 each start the game with \$20,000. The position of the players switches after each hand. The players alternate taking turns and may choose to either fold, call, or raise on their turn. Folding results in the player losing and the money in the pot being awarded to the other player. Calling means the player places a number of chips in the pot equal to the opponent's share. Raising means the player adds more chips to the pot than the opponent's share. There are four betting rounds in the game. A round ends when both players have acted at least once and the most recent player has called. Players cannot raise beyond the \$20,000 they start with. All raises must be at least \$100 and at least as large as any previous raise in that round.

At the start of the game P_1 must place \$100 in the pot and P_2 must place \$50 in the pot. Both players are then dealt two cards that only they observe from a 52-card deck. A round of betting then occurs starting with P_2. P_1 will be the first to act in all subsequent betting rounds. Upon completion of the first betting round, three *community* cards are dealt face up. After the second betting round is over, another community card is dealt face up. Finally, after that betting round one more community card is revealed and a final betting round occurs. If no player has folded then the player with the best five-card poker hand, out of their two private cards and the five community cards wins the pot. The pot is split evenly if there is a tie.

The most competitive agents for HUNL solve portions of the game (referred to as subgames) in real time during play (Brown & Sandholm, 2017a; Moravčík et al., 2017; Brown & Sandholm, 2017b; Brown et al., 2018). For example, *Libratus* solved in real time the remainder of HUNL starting on the third betting round. We conduct our experiments on four open-source subgames solved by *Libratus*

in real time during its competition against top humans in HUNL.[2] Following prior convention, we use the bet sizes of 0.5× the size of the pot, 1× the size of the pot, and the all-in bet for the first bet of each round. For subsequent bets in a round, we consider 1× the pot and the all-in bet.

Subgames 1 and 2 occur over the third and fourth betting round. Subgame 1 has $500 in the pot at the start of the game while Subgame 2 has $4,780. Subgames 3 and 4 occur over only the fourth betting round. Subgame 1 has $500 in the pot at the start of the game while Subgame 4 has $3,750. We measure exploitability in terms of the standard metric: milli big blinds per game (mbb/g), which is the number of big blinds (P_1's original contribution to the pot) lost per hand of poker multiplied by 1,000 and is the standard measurement of win rate in the related literature.

We compare the performance of three algorithms: vanilla CFR (i.e. CFR with regret matching; labeled CFR in plots), the current state-of-the-art algorithm in practice, Discounted CFR (Brown & Sandholm, 2019) (labeled DCFR in plots), and our stable-predictive variant of CFR with OFTRL at each decision point (labeled OFTRL in plots). DCFR was set up with parameters $(\alpha, \beta, \gamma) = (1.5, 0, 2)$, as recommended in the original DCFR paper. For OFTRL we use the stepsize that the theory suggests in our experiments on subgames 3 and 4 (labeled OFTRL theory). For subgames 1 and 2 we found that the theoretically-correct stepsize is much too conservative, so we also show results with a less-conservative parameter found through dividing the stepsize by 10, 100, and 1000, and picking the best among those (labeled OFTRL tuned). For all games we show two plots: one where all algorithms use simultaneous updates, as CFR traditionally uses, and one where all algorithms use alternating updates, a practical change that usually leads to better performance.

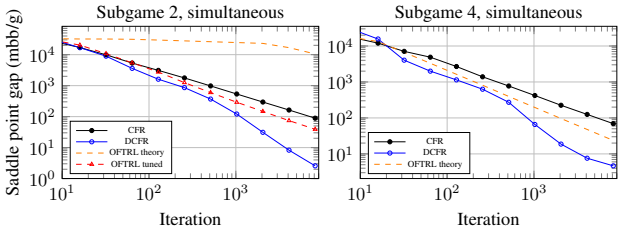

Figure 2. Convergence rate with iterations on the x-axis, and the exploitability in mbb. All algorithms use simultaneous updates.

Figure 2 shows the results for simultaneous updates on subgames 2 and 4, while Figure 4 in the appendix for subgames 1 and 3. In the smaller subgames 3 and 4 we find that OFTRL with the stepsize set according to our theory outperforms CFR: in subgame 4 almost immediately and significantly, in subgame 3 only after roughly 800 iterations. In contrast to this we find that in the larger subgames 1 and 2 the OFTRL stepsize is much too conversative, and the algorithm barely starts to make progress within the number of iterations that we run. With a moderately-hand-tuned stepsize OFTRL beats CFR somewhat significantly. In all games DCFR performs better than OFTRL, and also significantly better than its theory predicts. This is not too surprising, as both CFR$^+$ and the improved DCFR are known to significantly outperform their theoretical convergence rate in practice.

Figure 3 shows the results for alternating updates on subgames 2 and 4, while subgames 1 and 3 are given in the appendix in Figure 5. In the alternating-updates setting OFTRL performs worse relative to CFR and DCFR. In subgame 1 OFTRL with stepsizes set according to the theory slightly outperforms CFR, but in subgame 2 they have near-identical performance. In subgames 3 and 4 even the manually-tuned variant performs worse than CFR, although we suspect that it is possible to improve on this with a better choice of stepsize parameter. In the alternating setting DCFR performs significantly better than all other algorithms.

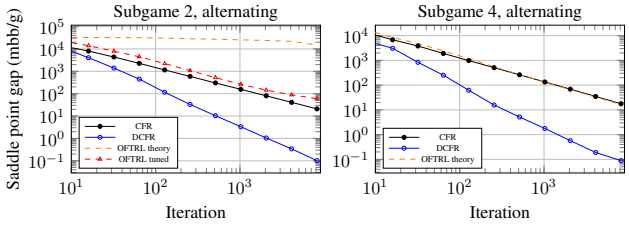

Figure 3. Convergence rate with iterations on the x-axis, and the exploitability in mbb. All algorithms use alternating updates.

7. Conclusions

We developed the first variant of CFR that converges at a rate better than $T^{-1/2}$. In particular we extend the ideas of predictability and stability for optimistic regret minimization on matrix games to the setting of EFGs. In doing so we showed that stable-predictive simplex regret minimizers can be aggregated to form a stable-predictive variant of CFR for sequential decision making, and we showed that this leads to a convergence rate of $O(T^{-3/4})$ for solving two-player zero-sum EFGs. Our result makes the first step towards reconciling the gap between the theoretical rate at which CFR converges, and the rate at which $O(T^{-1})$ first-order methods converge.

Experimentally we showed that our CFR variant can outperform CFR on some games, but that the choice of stepsize is important, while we find that DCFR is faster in practice. An important direction for future work is to find variants of our algorithm that still satisfy the theoretical guarantee and perform even better in practice.

[2] https://github.com/CMU-EM/LibratusEndgames

Acknowledgments

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1617590, and CCF-1733556, and the ARO under award W911NF-171-0082. Noam Brown is also sponsored by an Open Philanthropy Project AI Fellowship and a Tencent AI Lab Fellowship.

References

Bowling, M., Burch, N., Johanson, M., and Tammelin, O. Heads-up limit hold'em poker is solved. *Science*, 347 (6218), January 2015.

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 689–699, 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)*, 2019.

Brown, N., Sandholm, T., and Amos, B. Depth-limited solving for imperfect-information games. In *Neural Information Processing Systems, NeurIPS 2018*, 2018.

Burch, N. *Time and Space: Why Imperfect Information Games are Hard*. PhD thesis, University of Alberta, 2017.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Chiang, C.-K., Yang, T., Lee, C.-J., Mahdavi, M., Lu, C.-J., Jin, R., and Zhu, S. Online optimization with gradual variations. In *Conference on Learning Theory*, pp. 6–1, 2012.

Farina, G., Kroer, C., and Sandholm, T. Composability of Regret Minimizers. *arXiv e-prints*, art. arXiv:1811.02540, November 2018.

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

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.

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. Solving large sequential games with the excessive gap technique. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2018a.

Kroer, C., Waugh, K., Kılınç-Karzan, F., and Sandholm, T. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming*, pp. 1–33, 2018b.

Kuhn, H. W. A simplified two-person poker. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pp. 97–103. Princeton University Press, Princeton, New Jersey, 1950.

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte Carlo sampling for regret minimization in extensive games. In *COLT (Conference on Learning Theory) workshop on Online Learning with Limited Feedback*, 2009.

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*, 356(6337), May 2017.

Nemirovski, A. Prox-method with rate of convergence O(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1), 2004.

Nesterov, Y. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal of Optimization*, 16(1), 2005.

Rakhlin, A. and Sridharan, K. Online learning with predictable sequences. In *Conference on Learning Theory*, pp. 993–1019, 2013a.

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

Syrgkanis, V., Agarwal, A., Luo, H., and Schapire, R. E. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, pp. 2989–2997, 2015.

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.

Wang, J.-K. and Abernethy, J. D. Acceleration through optimistic no-regret dynamics. In *Advances in Neural Information Processing Systems*, pp. 3828–3838, 2018.

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.