Multiplicative Weight Updates for Extensive Form Games

Chirag Chhablani
University of Illinois at Chicago
Chicago, Illinois, USA
cchhab2@uic.edu

Michael Sullins
University of Illinois at Chicago
Chicago, Illinois, USA
sullins2@uic.edu

Ian A. Kash
University of Illinois at Chicago
Chicago, Illinois, USA
iankash@uic.edu

ABSTRACT

Recent research in Nash equilibrium (NE) computation in extensive forms games (EFGs), such as poker, show that it is possible to compute strong solutions for two-player zero-sum games via regret minimization in theory and practice. Regret minimization is less well-understood in other classes of EFGs, even with perfect information. We introduce an approach based on converting the EFG into its corresponding normal form game (NFG). This faces two challenges. First, the exponential increase in the size of the NFG representation makes the straightforward use of regret minimization algorithms, like Multiplicative weights update (MWU) variants, on the resulting game impractical. Second, it is not clear how the updates in the normal form version of the game translate to the update in the behavioral strategies of the extensive form. We address these two challenges by introducing Extensive-form Implementation of Normal-form Regret minimization (EINR). Like CFR, it can be applied locally and recursively to the decision nodes in extensive form version. Further, we show a way to extend the EINR implementation to simultaneous move games where each agent knows the state of the game only when all the other players have acted in the game. Experiments on a zero-sum extensive form game and a cooperative simultaneous move game provide a comparison to CFR.

KEYWORDS

Extensive form games; Simultaneous move games; Multiplicative weights update; Online learning

ACM Reference Format:
Chirag Chhablani, Michael Sullins, and Ian A. Kash. 2023. Multiplicative Weight Updates for Extensive Form Games. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 8 pages.

1 INTRODUCTION

Extensive-form games (EFGs) are a framework to model sequential decision-making among multiple agents. Counterfactual Regret Minimization (CFR) algorithm and its variants [3, 8, 39], have been developed to compute Nash Equilibrium in zero-sum EFGs. This has led to outperforming top human level poker professionals in No-Limit Texas Hold 'Em via agents such as DeepStack [29] and Libratus [5]. CFR minimizes regret locally in each state (information set) of the game tree using counterfactual values, which, in turn, are used to compute counterfactual regret for each action. This iterative process minimizes total regret due to the main CFR theorem [39], thus giving a local regret minimizing algorithm to minimize total regret.

For non-zero-sum EFGs, CFR has much weaker convergence guarantees and we have relatively few positive results even turning to other classes of games like cooperative games. On the other hand, there has been extensive exploration of convergence behavior of a variety of regret minimization algorithms such as Multiplicative Weights Update (MWU) and its variants in Normal Form Games (NFGs). For example, variants of MWU have last-iterate convergence and polylogarithmic regret bounds in general sum games and last iterate convergence in zero-sum games which is stronger than CFR (see Table 1 from Farina et al. [13]'s work for detailed comparison)

A desirable approach would be use the well-known conversion of EFGs to NFGs and then enjoy the benefits of the positive results for regret minimization in NFGs. However, since the number of deterministic strategies in an EFG is exponential in the size of the game, this conversion is intractable in general. We address this by developing EINR, an algorithm for EFGs with perfect information which produces the same updates as this conversion yet can be directly applied to the nodes of the EFG representation of game tree and hence doesn't suffer from the exponential blow-up of strategies. *Thus EINR simultaneously enjoys the convergence properties of NFG regret minimization and the computational efficiency of EFG algorithms like CFR (though restricted to perfect information).*

The key to EINR is the observation that tracking a certain version of cumulative utility at the leaves of the game provides a sufficient statistic to reconstruct the current probability of each strategy in the NFG according to MWU. At each iteration these statistics can be updated in a single top-down pass over the tree. They can then be used to calculate the extensive-form version of the current behavioral strategy. This approach mirrors the simple and efficient structure of CFR and uses many of the same concepts such as reach probabilities and counterfactual utilities. However, it differs in the details of the recursive calculation which we show eliminates the need to "unlearn" that an action early in the tree is bad when this is due to a poor initial strategy in the subtree following that action. Further, while EFGs usually assume one player acts at a time, we show that EINR naturally extends to settings with simultaneous moves. As simultaneous moves are typically modeled using imperfect information, this shows that EINR can be extended to at least some imperfect information EFGs. We leave the extension of EINR-like update to general imperfect information games as an interesting future direction.

Apart from zero-sum settings, an interesting domain on which to test EINR is simultaenous move-identical interest games (SM-IIGs), cooperative settings where all the agents act simultaneous and observe the state and reward of the game only after playing the action. In this setting, the players don't have any information about the

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

actions of the other players. This setting can be commonly observed in many real-world scenarios such as genetics [28], cooperation among a team of agents [27], and coordination amongst different functions in robotic control [31]. Our experiments on variants of an N-step SM-IIG introduced by Yang et al. [38] show that EINR achieves such last iterate convergence to a single leaf node and perhaps more surprisingly CFR does as well, albeit in some cases suboptimally.

In summary, our main contributions are:

(1) We establish that the counterfactual utility at terminal nodes in the EFG is a sufficient statistic for the mixed strategy of an agent following MWU in the NFG (Lemma 2).
(2) We introduce EINR, a simple, linear time, CFR-style algorithm for updating these statistics and reconstructing the current strategy in the EFG representation from them. We also show the implementation of EINR in simultaneous move games where at every step agents take actions without knowledge of the actions of the other agents. (Theorem 1 and Corollary 1).
(3) We provide comparison of the update rule for EINR and CFR and illustrate through a simple example that EINR avoids the need for "unlearning" that CFR faces (Section 4).
(4) While our primary contribution is theoretical, a small set of experiments demonstrates last iterate convergence of EINR in EFGs with perfect information and also simultaneous move games. In harder versions of the latter, EINR finds the optimal solution while CFR finds a suboptimal equilibrium (Section 5).

1.1 Related Work

Two recent works have independently developed algorithms to achieve the goal of extensive form implementations of normal form regret minimization. Farina et al. [13] develop a kernel-based approach for MWU which works in any polyhedral convex game (which includes EFGs). In general their algorithm tracks one number for each history of the game and they provide a generic algorithm which is quadratic this size and an optimized implementation for EFGs which is linear. Bai et al. [2] provide an approach for a generalization of MWU called Φ-Hedge in EFGs and show that the Farina et al. [13] results for EFGs are a special case of their framework and the linear time implementation is equivalent to the standard Online Mirror Descent algorithm. In contrast, EINR achieves the same linear time performance in EFGs with perfect information while tracking counterfactual utilities *only at the leaves of the tree.* This results in a simpler algorithms which keeps the CFR-like structure of a *single pass down and then up the tree.* In this sense our work is complementary because both rely on identifying structures where relevant computations can be performed efficiently and we identify a new version of this structure. Additionally our version is naturally phrased in terms of familiar concepts from CFR such as reach probabilities.

There are several works that study online no-regret learning algorithms for NFGs and achieve good convergence for many classes of games. In general in a NFG, it is known that any no-regret learning algorithm can achieve convergence to coarse correlated equilibrium. Further, no-regret algorithms like MWU can achieve Nash Equilibrium convergence for zero-sum games [14], congestion games [20, 23, 37], and cooperative games [28]. Optimistic variants of MWU [32, 35] also enables faster NE and last iterate convergence for zero-sum games and polylogarithmic regret bounds for general-sum NFGs [12, 25]. This work is complementary to ours because by effectively converting EFGs to NFGs we can leverage these convergence results as long as the induced NFG satisfies the assumptions under which the corresponding guarantees hold (see discussion by Farina et al. [13] in their Appendix A.1 and Section 5.3).

Recent work on team games [7] has show that the joint normal form strategies of team players against an adversary can be used to solve for an Team Maxmin equilibrium. However such an approach is not scalable as the number of players and decision points increase [6]. Due to the lack of scalability of the induced NFG, most of EFG solutions have been limited to finding algorithms directly for the EFGs. For example, for zero-sum EFG Nash equilibrium can be computed based on linear programs [33], double oracle methods [22] and most popularly by CFR [39] and its variants. However, as pointed out by [13], many of the normal form guarantees like last-iterate convergence, tight regret convergence, etc. from NFGs are still not easily extendable by algorithms designed mainly for EFG. Moreover, CFR-based learning algorithms lack theoretical guarantees for some settings like cooperative games [27].

More recently Markov potential games (MPGs) [16, 26], which like EFG add a stateful structure, have been studied as they can provide a more general framework for multi-step cooperative games. A Markov game is an MPG if there exists a potential function such that if one agent changes their policy, the difference in their value function is the same as the difference in the potential function in all states. It was recently shown that Independent policy gradient [26] and Independent Natural Policy gradient [16] have updates identical to applying MWU in each state and that they converge to Nash equilibrium in MPGs. Relative to these our work represents an alternative approach to extending MWU to stateful settings.

Apart from game-theoretic approaches, decentralized multiagent reinforcement learning (MARL) techniques have also achieved success in the multi-agent cooperative[9, 18] and competitive settings [1, 11, 34]. Some of these MARL that use centralized training and decentralized execution techniques [15, 27] use Regret or Regret-like functions to update the policy of all the agents. However, almost all of these MARL techniques lack the generic theoretical guarantees MWU and its variants have in NFG. Moreover, MWU has been shown to be essentially equivalent to the tabular version of many Deep-RL algorithms [16, 21, 26] which shows the promise of MWU in stateful RL settings.

2 BACKGROUND

2.1 Normal Form Games

Definition 1. *A normal form game (NFG) is a single step game defined by tuple* $(N, (\mathbb{S}_i)_{i\in N}, (u_i)_{i\in N})$ *where* N *is the number of players and each player has a set of pure strategies* $\mathbb{S}_i = \{s_{i1}, s_{i2}, .., s_{in_i}\}$. *For a strategy profile denoted by* $\boldsymbol{s} = (s_1, s_2, .., s_N)$ *the utility of player* i *is given by* $u_i(\boldsymbol{s})$ *where* u_i *is payoff function defined over* $\mathbb{S}_1 \times \mathbb{S}_2 \times .. \times \mathbb{S}_N \to \mathbb{R}$.

We use $P_i = \Delta(\mathbb{S}_i)$ to denote the set of player i's mixed (randomized) strategies and $P = (P_1, P_2 ..., P_N)$ the set of mixed strategies of all players.

Definition 2. *An identical interest game (NFG-IIG) is a N-player game defined by tuple $G = (N, (\mathbb{S}_i)_{i \in N}, u))$. In an IIG, all the agents receive the same shared reward u for a particular joint action $\mathbf{s} = (s_1, s_2 ... s_N) \in \boldsymbol{S}, i, j \in N$ i.e*

$$u_i(\boldsymbol{s}) = u_j(\boldsymbol{s}) = u(\boldsymbol{s}) \tag{1}$$

All NFG-IIGs have at least one joint action $\mathbf{s}^*$ which is a maximizer of the shared reward $u(\mathbf{s}^*)$. With such a joint action, indeed for any local optimum of u, for a player i playing strategy s_i and rest of the players playing strategy $\mathbf{s}_{-i}$, where $-i$ subscript is used to indicate the strategy of all the players except i,

$$u_i(\mathbf{s}^*) \geq u_i(s_i, \mathbf{s}^*_{-i}) \forall s_i \in \mathbb{S}_i \tag{2}$$

Such a point is known as a pure Nash equilibrium.

Finally,the expected utility for strategy s_i denoted by $l_i(s_i)$ for the player i in an NFG where the over players follow a mixed strategy p_{-i} is given by -

$$l_i(s_i) = \sum_{\mathbf{s}_{-i} \in \mathbb{S}_{-i}} p_{\mathbf{s}_{-i}} * u_i(s_i, \mathbf{s}_{-i}) \tag{3}$$

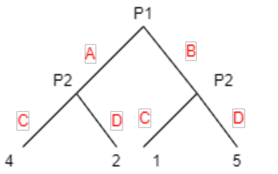
2.2 Extensive form game

An extensive form game (EFG), unlike a NFG, is a game with sequential interaction and is generally represented by a game tree. The non-terminal nodes of the tree represent a single player taking action at that decision node and a terminal node represent the end of the game and the utilities received by each player after the terminal node is reached. The edges of the tree represent the player's strategies for the node. An example of such a game is shown in figure 1.

A perfection information[1] EFG is formulated as a tuple (N, H, A, ρ, U). $N = \{1, ..., n\}$ is a set of players. H is a set of histories (i.e., the possible action sequences) where each history $h \in H$ is the sequence of actions taken by all the players according to history $h = (a^0, a^1, ... a^t)$. The empty sequence ϕ which is the root node of the game tree is in H, and for every prefix h' of a sequence $h \in H$, $h' \in H$. The prefix relationship between the history h' and h is denoted by $h' \sqsubseteq h$. $Z \subset H$ is the set of the terminal histories. ρ is the player function. $\rho(h)$ is the player who takes an action at the history h, i.e. For a non-terminal history $h \in H$, $\rho(h) \to N$. For the player $\rho(h) = i$, let $A(h)$ be the set of available actions at history h. For all players $i \in N$, the utility function is a mapping $u_i : Z \to R$.

A player's behavior strategy σ_i is a function mapping every history h where $\rho(h) = i$ to a probability distribution over $A(h)$. A strategy profile σ consists of a strategy for each player $\sigma_1, \sigma_2, ..., \sigma_n$ with σ_{-i} referring to all the strategies in σ except σ_i. Let $\pi^\sigma(h)$ be the probability of reaching history h if players choose actions according to σ. The reach probability $\pi^{\sigma_i}(h)$ of player i denotes the probability with which player i will play the actions required to trace the history h and is given by $\pi^{\sigma_i}(h) = \Pi_{\substack{(h',a) \sqsubseteq h \\ \rho(h)=i}} \sigma_i(a|h')$. For any player i, the counterfactual utility is $\lambda_i : Z \to R$ and for $z \in Z$, $\lambda_i(z) = \pi^{\sigma_{-i}}(z) u_i(z)$. We also make use of the cumulative counterfactual utility $\Lambda^\tau_i(z) = \sum_{t=0}^{\tau} \lambda^t_i(z)$

To leverage the equilibrium guarantees from NFG in EFG, we use the standard fact that all EFG games can be transformed into equivalent NFGs (see Figure 1 for a simple example with an IIG) by converting the pure behavioral strategies of the EFG to pure strategies in the NFG. We call the resulting induced NFG the INFG for general case or IIIG for identical interest games.



(a) Extensive form shared reward game

	CC	CD	DC	DD
A	4	4	2	2
B	1	5	1	5

(b) Normal form transformation of EFG

Figure 1: Normal and Extensive form version of an IIG

To make our analysis of the connection between extensive form and normal form strategies easier, we define a few custom notations in addition to the standard notations for our perfect information EFG. Let r^h be the action chosen by player i according to the normal form strategy $r \in \mathbb{S}_i$ of the INFG after history h such that $\rho(h) = i$. $\mathbb{S}^h_i$ denotes the set of normal form strategies for player i that can lead to history h. That is, $\mathbb{S}^h_i = \{r \in \mathbb{S}_i \mid \exists (h', a) \sqsubseteq h, r^{h'}_i = a, \rho(h') = i\}$. When h is a terminal history such that $h \in Z$, we use the z for the superscript as $\mathbb{S}^z_i$. Further, let $\mathbb{Z}^r$ denote the set of terminal histories reachable through the normal form strategy r. That is, $\mathbb{Z}^r = \{z \in Z \mid r \in \mathbb{S}^z_i\}$. Similarly, let $\mathbb{Z}^h$ denote the set of terminal histories reachable from history h. That is, $\mathbb{Z}^h = \{z \in Z \mid h \sqsubseteq z\}$.

2.3 Multiplicative Weights Update (MWU)

In this paper we study the class of MWU algorithms termed aggregate monotonic selection (AMS) which was studied by Kleinberg et al. [23]. This class notably includes the Hedge algorithm of Freund and Schapire [17]. These algorithms maintain a vector p_i of probabilities for the available actions, so that p_{ir} is the probability of player i choosing action r. In each round each player samples an action according to this distribution. Afterwards, the weights are updated multiplicatively, based on the realized cost.

$$p_{ir} = \frac{p_{ir}(1 - \epsilon\beta)^{u_i(p)}}{\sum_{r' \in s_i} p_{ir'}(1 - \epsilon\beta)^{u_i(p)}} \tag{4}$$

In the update, ϵ is a base learning rate and $\beta(i, p)$ is a parameter allowing it to be state-dependent. Following Kleinberg et al. [23], we assume that $\forall i, p, 0 < \beta(i, p) \leq 1$. Hedge, which we will use later, results from taking $\beta = 1$.

3 EXTENSIVE FORM IMPLEMENTATION OF NORMAL FORM REGRET MINIMIZATION

In this section we show how address the exponential number of normal-form strategies. We do so by developing an algorithm we

[1] Perfect information means that all players can observe the full history of play. In imperfect information games players may have uncertainty about the past which is captured via information sets.

call EINR, for Extensive-form Implementation of Normal-form Regret Minimization. As the name suggests, it allows us to compute the same behavior strategies which result from MWU dynamics on the NFG via computations performed efficiently on the EFG. For notational simplicity we present our results for one particular MWU dynamic (Hedge, with ϵ chosen to make the base of the exponent e), but as this only affects the weighting of terms in the exponent our results apply to arbitrary MWU dynamics. Similarly, we assume there are only two players, i and $-i$, as when performing updates for i the other players can be treated as a unit. We begin by stating the Hedge update rule in a convenient form.

Definition 3. *Let p_i^t be the normal form mixed strategy of player i at time step t. The Hedge update is*

$$p_i^{t+1}(r) = \frac{p_i^t(r) exp(l_i^t(r))}{\sum_{r' \in \mathbb{S}_i} p_i^t(r) exp(l^t(r'))} \tag{5}$$

where $r \in \mathbb{S}_i$ is a normal form pure strategy and $l_i^t(r)$ is the expected utility of r at time t per Equation (3).

Assuming all players start with uniform initial probabilities, applying Equation (5) recursively yields

$$p_i^{t+1}(r) = \frac{exp(L_i^t(r))}{\sum_{r_i \in \mathbb{S}_i} exp(L_i^t(r_i))} \tag{6}$$

where $L_i^\tau(r) = \sum_{t=0}^{\tau} l_i^t(r)$ denotes the accumulated expected utilities for player i using pure strategy r through time τ.

This update rule for INFG strategies induces an update rule for extensive form behavior strategies through the standard transformation.

Definition 4. *Let p_i, a mixed strategy for player i, be given. The behavior strategy induced by p_i is defined as follows. Let h be a history such that $\rho(h) = i$ and let $a \in A(h)$. Then*

$$\sigma_i(a|h) = \frac{\sum\limits_{r \in \mathbb{S}_i^h, r_i^h = a} p_i(r)}{\sum\limits_{b \in A(h)} \sum\limits_{r \in \mathbb{S}_i^h, r_i^h = b} p_i(r)} \tag{7}$$

In other words, the behavior strategy can be given as the summation of normal form probabilities of strategies of player i that can reach h and play a at history h, normalized by the probability of all the strategies that can reach h. Given this connection, the reach probabilities satisfy the obvious property that they can be computed as the summation of normal form probabilities of strategies of player i that can reach h

Lemma 1. *Let p_i be a mixed strategy and σ_i be the induced behavior strategy. Then the reach probabilities satisfy*

$$\pi^{\sigma_i}(h) = \sum_{r \in \mathbb{S}_i^h} p_i(r) \tag{8}$$

The proof is deferred to the appendix. Since counterfactual utilities are defined in terms of reach probabilities, this allows us to give a concise characterization of the expected utility of a normal-form pure strategy.

Lemma 2. *Let $r \in \mathbb{S}_i$ be a normal form strategy for the player i and recall that $\mathbb{Z}^r = \{z \in Z \mid r \in \mathbb{S}_i^z\}$ is the set of terminals reachable when playing r. Then*

$$l_i^t(r) = \sum_{z \in \mathbb{Z}^r} \lambda_i^t(z) \tag{9}$$

Proof. By definition

$$l_i^t(r) = \sum_{r' \in \mathbb{S}_{-i}} p_{-i}^t(r') u_i(r, r')$$

Let $\mathbb{S}_{-i}^z$ denote the subset of normal form strategies of player $-i$ leading to terminal history z given player i is playing r.

Then

$$l_i^t(r) = \sum_{z \in Z^r} \Big(\sum_{r' \in \mathbb{S}_{-i}^z} p_{-i}^t(r') \Big) u_i(z)$$

By Lemma 1 and the definition of counterfactual utilities,

$$l_i^t(r) = \sum_{z \in Z^r} \pi^{\sigma_{-i}^t}(z) u_i(z) = \sum_{z \in \mathbb{Z}^r} \lambda_i^t(z) \tag{10}$$

□

Lemma 2 is key to the design of EINR. It shows that the cumulative counterfactual utilities of terminal histories are a sufficient statistic for the probabilities of the normal form strategies, and thus via Equation (7) the behavior strategy. All that remains is to show that we can efficiently compute it from the sufficient statistic. The following theorem does so.

Theorem 1. *Let player i be using the Hedge update. Then*

$$\sigma_i^\tau(a|h) = \frac{V_i^\tau(h.a)}{\sum_{b \in A(h)} V_i^\tau(h.b)} \tag{11}$$

where $V_i^\tau(h)$ is defined recursively as

$$V_i^\tau(h) = \begin{cases} exp(\Lambda_i^t(h)), & \text{if } h \in Z \\ \prod_{a' \in A(h)} V_i^\tau(h.a'), & \text{if } \rho(h) = -i \\ 1/|A(h)| \sum_{a' \in A(h)} V_i^\tau(h.a'), & \text{if } \rho(h) = i \end{cases} \tag{12}$$

The full proof is deferred to the appendix, but we provide a brief sketch here. First, we show that the probability of an action at a given history can be factored into two parts: one which depends only on the subtree rooted at that history and one which depends on the parts of the tree that do not pass through that history. We provide a recursive characterization of the first part while showing that the second is the same for all actions at a history and therefore cancels out. This recursive characterization yields the given form for V.

Thus, like CFR, each iteration of EINR consists of two passes over the extensive form game tree. First, a top-down pass calculates the reach probability of each terminal history according to the current behavior strategy and uses them to update the cumulative counterfactual utilities $\Lambda_i^\tau(z)$. Then a bottom-up pass computes V according to Equation (12) and uses it to update the behavior strategy according to Equation (11). Thus each iteration of EINR is linear in the size of the extensive form game.

As a final remark, we note that it is straightforward to extend EINR to allow simultaneous moves. This allows us to capture, for example, NFGs and (finitely) repeated games. It also demonstrates that

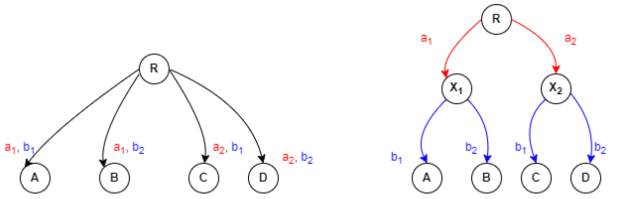


Figure 2: Conversion of 2-Simultaneous Move Game to Extensive form games from player 1's perspective

EINR can handle at least some games with imperfect information (which is one may to model simultaneous moves).

Corollary 1. *EINR extends to EFGs with simultaneous moves by defining V at nodes where both i and another player move as*

$$V_i^\tau(h) = 1/|A_i(h)| \sum_{a'\in A_i(h)} \prod_{a''\in A_{-i}(h)} V_i^\tau(h.(a',a'')) \qquad (13)$$

Proof. Nodes with simultaneous moves can be replaced by a tree where each player acts in turn in an arbitrary order as shown in Figure 2. Putting i at the root yields the given form by combining the resulting 2 cases of (12). While under this transformation we should restrict the actions of the agent(s) $-i$ (in the language of imperfect information the nodes are in the same information set), from the perspective of player i the actions of other agents are arbitrary and used only to calculate the reach probabilities at the terminals. □

4 COMPARISON TO CFR

The analysis in Section 3 highlighed the similarity in the structure of CFR and EINR. We now examine their differences more carefully. As a starting point, it is instructive to compare the V defined in Equation (12) with the counterfactual value used by CFR

$$v_i(\sigma,h) = \sum_{z\in\mathbb{Z}^h} \pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z) \qquad (14)$$

Both ultimately reduce to the counterfactual values at terminals reachable from h, but there are two key differences. First, CFR accumulates the counterfactual values themselves in an expected-value-like calculation. In contrast, EINR accumulates, them in the exponent. While this ends up being equivalent at histories where $\rho(h) \neq i$ (because multiplying exponential sums their exponents), when $\rho(h) = i$ the behavior is quite different and EINR essentially computes the average of the weights associated with the children rather than the values themselves. This enables the second difference: $V_i(h)$ has no dependence on i's current strategy. CFR needs to know the strategy to know how to weight different options. Since EINR directly tracks weights, it simply does an equally weighted average, which in some sense can be thought of as picking a uniform random policy starting from h.

4.1 A single player example

To illustrate the resulting difference in behavior, consider the following simple example with one player shown in Figure 3. The player first chooses left or right. Left results in a terminal. Right results in a further choice between left and right, with both resulting in terminals.

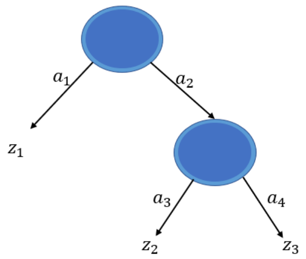


Figure 3: Single player game.

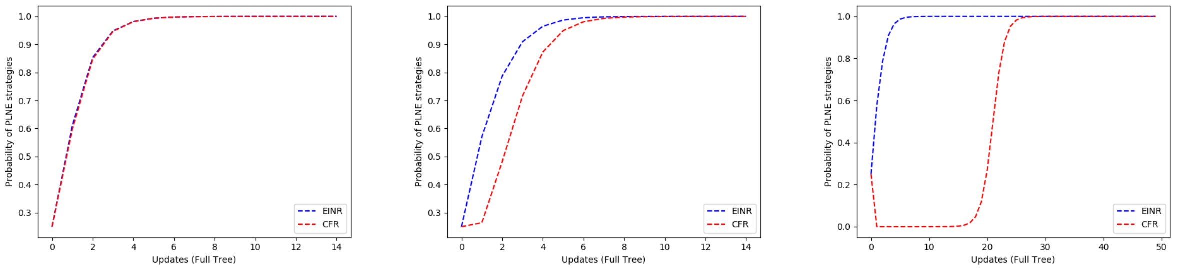
Figure 4 compares the behavior of EINR with CFR with Hedge as its regret minimizer. While in the literature, and in practice, CFR nearly exclusively uses regret matching [19] as its local regret minimization algorithm (due to lack of exponentiation, parameter-freeness, and ease of implementation), we instantiate CFR here with Hedge for as close a comparison as possible. The three subfigures show three different choices for the values of the terminals, differing only in the result of right followed by left. In this game right followed by right is the unique optimal strategy and the y-axis indicates the probability with which this is played.

When going right first is clearly better in Figure 8a (because both 2 and 3 are larger than 1), the two algorithms behave (nearly) identically. However, when going right includes a "bad" option, EINR converges substantially faster. With the uniform random starting strategy, playing right at the root is quite bad (particularly in Figure 8c). This means that CFR needs to unlearn its aversion to playing right, and this aversion grows with the strength of the initial negative experience. In contrast, EINR is implicitly tracking *every* pure strategy, so this negative experience only affects the strategy of right then left leaving nothing to unlearn. Of course, there are optimizations to CFR designed to speed this unlearning, such as CFR+ [36], but the inherent lack of need for such heuristics highlights an inherent advantage of our approach.

5 EXPERIMENTS

We test our implementation of EINR on Tic-Tac-Toe game and a N-step simultaneous move game (both using Openspiel [24]) . Our implementation can be found on the github repository[2] While Tic-Tac-Toe is regarded as a simple game by humans, it has a high branching factor at early histories. In particular, it is substantially larger than common benchmarks such as Kuhn and Leduc poker. Further, the Nash equilibrium of the game is to settle in a tie where both players will receive 0 payoff. To exactly play such a best response is challenging as the breadth and depth of game tree is large. For the SMG, we choose the standard N-step identical interest (SMG-IIG) from the cooperative multi-agent learning literature. The main reason for which we choose SMG-IIG is because, unlike zero-sum games, the results of MWU in EFG-IIG or SMG-IIG

[2] https://github.com/chiggy2402/EINR.git



(a) 1,2,3

(b) 1,-2,2

(c) 1,-40,2

Figure 4: Convergence of CFR depends on the extent to which avoiding right initially needs to be unlearned unlike EINR. Captions of subfigures reflect the terminal utilities in Figure 3

are relatively less well-known although both zero-sum and IIG enjoy good convergence for NFG. For example, Kleinberg et al. [23] show that in potential games (a generalization of NFG-IIGs also known as congestion games) continuous time and noisy versions of Hedge achieve last iterate convergence to Nash equilibrium, Mehta et al. [28] show convergence of a “linear” variant of Hedge converges generically to pure Nash equilibrium in 2 agent NFG-IIGs, Palaiopanos et al. [30] showed a version of Hedge with suitably decreasing learning rates converges in potential games, and Coucheney et al. [10] and Heliou et al. [20] showed similar results with bandit feedback. While none of these result exactly match EINR, they give us a strong reason to expect last iterate convergence to Nash equilibrium in an N-step SM-IIG. Finally, N-step SM-IIG games are specifically used in the MARL literature to test if the algorithm can choose long term high gain instead of a short term small gain.

In general we cannot expect to learn a pure Nash equilibrium because in an SMG-IIG, there is room for arbitrary random behavior at histories which will not be reached. However, we find that in SM-IIG, both EINR and CFR, the joint behavioral strategy converges to a single leaf node. This means that $\pi^{\sigma}(z) = 1$ for some $z \in Z$

The following results use a game specific learning rate for Hedge that was found experimentally. Tic-tac-toe has rewards of +1/-1 so a learning rate effectively scales the rewards up or down. As tic-tac-toe is a large game tree, larger learning rates were found to dramatically speed up convergence. Regret matching, used in CFR, tracks only positive regret sums which has a similar effect to a large learning rate in this context.

5.1 Tic-tac-toe

Tic-tac-toe is a simple two player game with perfect information. On a 3x3 grid, players alternate placing ’X’ and ’O’ in a cell until one player makes three in a row, column, or diagonal, or the game results in a tie. Tic-tac-toe has 255168 unique states. Optimal play results in a draw regardless of the initial player’s move. We run EINR and CFR (with the standard choice of regret matching and the regret minimizer) on tic-tac-toe and measure the optimality of the learned policy. This is done by computing the game value, which uses a copy of the policy for each player in self-play. A game value of 0 indicates a tie. EINR learns rapidly and effectively with a large learning rate of 10^5. Intuitively, this large learning rate allows it to effectively mirror the standard minimax / backward induction approach to solving tic-tac-toe.

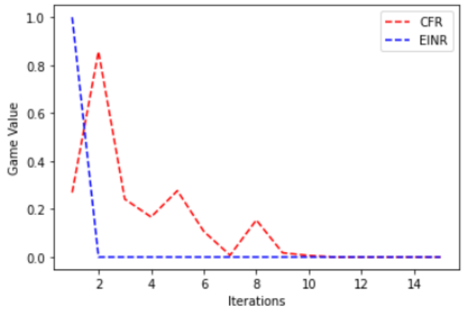


Figure 5: Tic-tac-toe.

5.2 Simultaneous move games

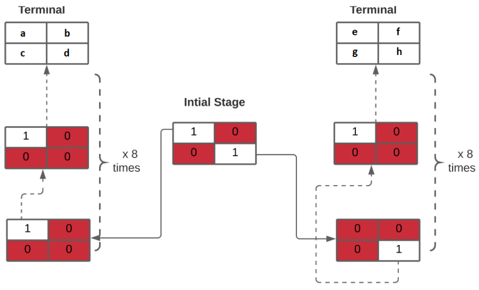
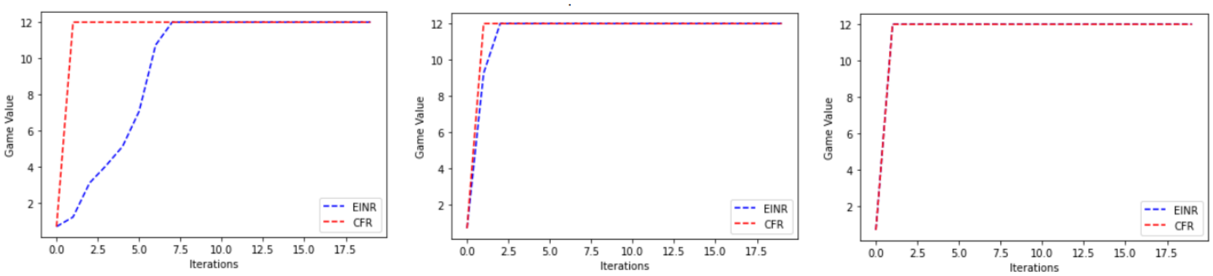


Figure 6: Simultaneous move game.

Here we study the simultaneous move game from Yang et al. [38] shown in Figure 6. Two players play a matrix game at each stage of the full game. By cooperating, the game can continue. The players are presented with an increasingly higher payoff the further into the game they get. If cooperation is maintained, at the terminals each player will receive the highest payoff available. Payoffs are

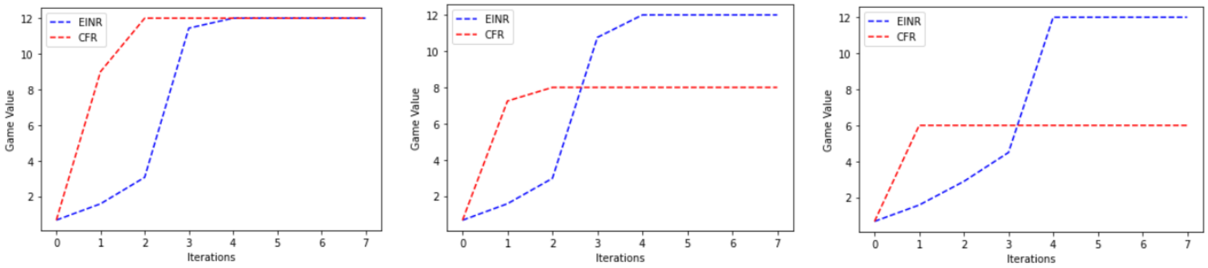


(a) Learning rate of 10.

(b) Learning rate of 100.

(c) Learning rate of 1000.

Figure 7: Learning curves for EINR/CFR in the standard SMG. Increasing the learning rate shows EINR approaching the number of iterations as CFR.



(a) Modified payoffs (a=4, c=4)

(b) Medium difficulty. (a=-4, c=-4)

(c) Hard Difficulty. (a=-40, c=-40)

Figure 8: Modified game

only received at the terminals. In Figure 7, a = 9, b = 9, c = 9, d = 12 and e = 9, f = 9, g = 9, h = 9. In this scenario, cooperation results in a higher payoff of 12. Our results show that EINR performs similarly to CFR. By increasing the learning rate, EINR approaches the same number of iterations for convergence that CFR achieves.

In order to compare to CFR in a more challenging setting, the next example modifies the payoffs. In Figure 8a, a = 4, b = 12, c = 4, d = 12 and e = -40, f = -40, g = 11, h = 11. The payoff for cooperation remains 12. In Figure 8b, a slight change in the rewards is now made where a and c are negated (a = -4, c = -4). As a result, EINR continues to converge to the highest game value. CFR, however, converges suboptimally to 8, corresponding to the players coordinating on terminating before reaching the final step. Finally, in Figure 8c the values for a and c are decreased one order of magnitude (a=-40, c=-40). EINR still converges to the highest game value. CFR performs even worse and converges to a game value of 6 by terminating even earlier. Intuitively, in these harder examples CFR learns to coordinate on the suboptimal equilibrium before it unlearns its initial discovery that the last stage is dangerous under random play.

6 CONCLUSION

We have developed EINR, a way of implementing regret minimization on the normal form representation of an extensive form game with perfect information while efficiently performing computations on the extensive form representation. There are a number of natural directions for future work to extend this. As discussed, our approach to doing these computations differs from those developed by Farina et al. [13] and Bai et al. [2]. Does our approach extend to some of the richer settings they explore? Another direction is whether we can create extensions of EINR by taking inspirations from similar CFR variants like Deep-CFR [4] or [5]. This would allow a richer comparison to CFR and develop faster variants of EINR. In our experiments, We chose an IIG due to their known strong convergence properties, but to our knowledge no prior result exactly covers the exact setting used by EINR, much less our empirical results for CFR. In addition to filling in this gap, many prior results extend to potential games. Do similar properties hold for extensive-form versions of potential games? Recent work has shows that defining potential games in more general settings can be subtle [26]. Finally, while we have described EINR in terms of the traditional Hedge algorithm, it can also be used with the optimistic variant [32, 35] by adjusting the utilities used in the updates appropriately. The benefits of doing so remain to be explored.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2217023

REFERENCES

[1] Yu Bai and Chi Jin. 2020. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*. PMLR, 551–560.

[2] Yu Bai, Chi Jin, Song Mei, Ziang Song, and Tiancheng Yu. 2022. Efficient Φ-Regret Minimization in Extensive-Form Games via Online Mirror Descent. *arXiv preprint arXiv:2205.15294* (2022).

[3] Branislav Bosansky and Jiri Cermak. 2015. Sequence-form algorithm for computing stackelberg equilibria in extensive-form games. In *Twenty-Ninth AAAI Conference on Artificial Intelligence.*

[4] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. Deep counterfactual regret minimization. In *International conference on machine learning*. PMLR, 793–802.

[5] Noam Brown and Tuomas Sandholm. 2017. Libratus: The Superhuman AI for No-Limit Poker.. In *IJCAI*. 5226–5228.

[6] Federico Cacciamani, Andrea Celli, Marco Ciccone, and Nicola Gatti. 2021. Multi-agent coordination in adversarial environments through signal mediated strategies. *arXiv preprint arXiv:2102.05026* (2021).

[7] Andrea Celli and Nicola Gatti. 2018. Computational results for extensive-form adversarial team games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[8] Andrea Celli, Alberto Marchesi, Tommaso Bianchi, and Nicola Gatti. 2019. Learning to correlate in multi-player general-sum sequential games. *Advances in Neural Information Processing Systems* 32 (2019).

[9] Dingyang Chen, Yile Li, and Qi Zhang. 2022. Communication-Efficient Actor-Critic Methods for Homogeneous Markov Games. *arXiv preprint arXiv:2202.09422* (2022).

[10] Pierre Coucheney, Bruno Gaujal, and Panayotis Mertikopoulos. 2015. Penalty-regulated dynamics and robust learning procedures in games. *Mathematics of Operations Research* 40, 3 (2015), 611–633.

[11] Constantinos Daskalakis, Dylan J Foster, and Noah Golowich. 2020. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems* 33 (2020), 5527–5540.

[12] Constantinos Daskalakis and Ioannis Panageas. 2018. Last-iterate convergence: Zero-sum games and constrained min-max optimization. *arXiv preprint arXiv:1807.04252* (2018).

[13] Gabriele Farina, Chung-Wei Lee, Haipeng Luo, and Christian Kroer. 2022. Kernelized Multiplicative Weights for 0/1-Polyhedral Games: Bridging the Gap Between Learning in Extensive-Form and Normal-Form Games. *arXiv preprint arXiv:2202.00237* (2022).

[14] Michail Fasoulakis, Evangelos Markakis, Yannis Pantazis, and Constantinos Varsos. 2022. Forward Looking Best-Response Multiplicative Weights Update Methods for Bilinear Zero-sum Games. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 11096–11117.

[15] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*.

[16] Roy Fox, Stephen M Mcaleer, Will Overman, and Ioannis Panageas. 2022. Independent natural policy gradient always converges in Markov potential games. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4414–4425.

[17] Yoav Freund and Robert E Schapire. 1999. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* 29, 1-2 (1999), 79–103.

[18] FNU Hairi, Jia Liu, and Songtao Lu. 2021. Finite-Time Convergence and Sample Complexity of Multi-Agent Actor-Critic Reinforcement Learning with Average Reward. In *International Conference on Learning Representations*.

[19] Sergiu Hart and Andreu Mas-Colell. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

[20] Amélie Heliou, Johanne Cohen, and Panayotis Mertikopoulos. 2017. Learning with bandit feedback in potential games. *Advances in Neural Information Processing Systems* 30 (2017).

[21] Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Remi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duenez-Guzman, et al. 2019. Neural replicator dynamics. *arXiv preprint arXiv:1906.00190* (2019).

[22] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. 2011. A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 327–334.

[23] Robert Kleinberg, Georgios Piliouras, and Éva Tardos. 2009. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 533–542.

[24] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. 2019. OpenSpiel: A Framework for Reinforcement Learning in Games. *CoRR* abs/1908.09453 (2019). arXiv:1908.09453 [cs.LG] http://arxiv.org/abs/1908.09453

[25] Qi Lei, Sai Ganesh Nagarajan, Ioannis Panageas, et al. 2021. Last iterate convergence in no-regret learning: constrained min-max optimization for convex-concave landscapes. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1441–1449.

[26] Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. 2021. Global convergence of multi-agent policy gradient in markov potential games. *arXiv preprint arXiv:2106.01969* (2021).

[27] Shuxin Li, Youzhi Zhang, Xinrun Wang, Wanqi Xue, and Bo An. 2021. CFR-MIX: Solving imperfect information extensive-form games with combinatorial action space. *arXiv preprint arXiv:2105.08440* (2021).

[28] Ruta Mehta, Ioannis Panageas, and Georgios Piliouras. 2015. Natural selection as an inhibitor of genetic diversity: Multiplicative weights updates algorithm and a conjecture of haploid genetics [working paper abstract]. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. 73–73.

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

[30] Gerasimos Palaiopanos, Ioannis Panageas, and Georgios Piliouras. 2017. Multiplicative weights update with constant step-size in congestion games: Convergence, limit cycles and chaos. *Advances in Neural Information Processing Systems* 30 (2017).

[31] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. 2021. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 12208–12221.

[32] Alexander Rakhlin and Karthik Sridharan. 2013. Online learning with predictable sequences. In *Conference on Learning Theory*. PMLR, 993–1019.

[33] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations.* Cambridge University Press.

[34] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. 2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in neural information processing systems*. 3422–3435.

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

[36] Oskari Tammelin. 2014. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042* (2014).

[37] Dong Quan Vu, Kimon Antonakopoulos, and Panayotis Mertikopoulos. 2021. Fast Routing under Uncertainty: Adaptive Learning in Congestion Games via Exponential Weights. *Advances in Neural Information Processing Systems* 34 (2021), 14708–14720.

[38] Yaodong Yang, Ying Wen, Lihuan Chen, Jun Wang, Kun Shao, David Mguni, and Weinan Zhang. 2020. Multi-Agent Determinantal Q-Learning. *arXiv preprint arXiv:2006.01482* (2020).

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