Regret-Based Pruning in Extensive-Form Games

Noam Brown
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15217
noamb@cmu.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15217
sandholm@cs.cmu.edu

Abstract

Counterfactual Regret Minimization (CFR) is a leading algorithm for finding a Nash equilibrium in large zero-sum imperfect-information games. CFR is an iterative algorithm that repeatedly traverses the game tree, updating regrets at each information set. We introduce an improvement to CFR that prunes any path of play in the tree, and its descendants, that has negative regret. It revisits that sequence at the earliest subsequent CFR iteration where the regret could have become positive, had that path been explored on every iteration. The new algorithm maintains CFR's convergence guarantees while making iterations significantly faster—even if previously known pruning techniques are used in the comparison. This improvement carries over to CFR+, a recent variant of CFR. Experiments show an order of magnitude speed improvement, and the relative speed improvement increases with the size of the game.

1 Introduction

Extensive-form imperfect-information games are a general model for strategic interaction. The last ten years have witnessed a leap of several orders of magnitude in the size of two-player zero-sum extensive-form imperfect-information games that can be solved to (near-)equilibrium [11][2][6]. This is the game class that this paper focuses on. For small games, a linear program (LP) can find a solution (that is, a Nash equilibrium) to the game in polynomial time, even in the presence of imperfect information. However, today's leading LP solvers only scale to games with around 10^8 nodes in the game tree [4]. Instead, iterative algorithms are used to approximate solutions for larger games. There are a variety of such iterative algorithms that are guaranteed to converge to a solution [5, 3, 10]. Among these, *Counterfactual Regret Minimization (CFR)* [16] has emerged as the most popular, and *CFR+* as the state-of-the-art variant thereof [13, 14].

CFR begins by exploring the entire game tree (though sampling variants exist as well [9]) and calculating regret for every hypothetical situation in which the player could be. A key improvement that makes CFR practical in large games is *pruning*. At a high level, pruning allows the algorithm to avoid traversing the entire game tree while still maintaining the same convergence guarantees. The classic version of pruning, which we will refer to as *partial pruning*, allows the algorithm to skip updates for a player in a sequence if the other player's current strategy does not reach the sequence with positive probability. This dramatically reduces the cost of each iteration. The magnitude of this reduction varies considerably depending on the game, but can easily be higher than 90% [9], which improves the convergence speed of the algorithm by a factor of 10. Moreover, the benefit of partial pruning empirically seems to be more significant as the size of the game increases.

While partial pruning leads to a large gain in speed, we observe that there is still room for much larger speed improvement. Partial pruning only skips updates for a player if an *opponent's* action in the path leading to that point has zero probability. This can fail to prune paths that are actually prunable. Consider a game where the first player to act (Player 1) has hundreds of actions to choose

from, and where, over several iterations, the reward received from many of them is extremely poor. Intuitively, we should be able to spend less time updating the strategy for Player 1 following these poor actions, and more time on the actions that proved worthwhile so far. However, here, partial pruning will continue to update Player 1's strategy following each action in every iteration.

In this paper we introduce a better version of pruning, *regret-based pruning (RBP)*, in which CFR can avoid traversing a path in the game tree if *either* player takes actions leading to that path with zero probability. This pruning needs to be temporary, because the probabilities may change later in the CFR iterations, so the reach probability may turn positive later on. The number of CFR iterations during which a sequence can be skipped depends on how poorly the sequence has performed in previous CFR iterations. More specifically, the number of iterations that an action can be pruned is proportional to how negative the regret is for that action. We will detail these topics in this paper.

RBP can lead to a dramatic improvement depending on the game. As a rough example, consider a game in which each player has very negative regret for actions leading to 90% of nodes. Partial pruning, which skips updates for a player when the opponent does not reach the node, would traverse 10% of the game tree per iteration. In contrast, regret-based pruning, which skips updates when either player does not reach the node, would traverse only $0.1 \cdot 0.1 = 1\%$ of the game tree. In general, RBP roughly squares the performance gain of partial pruning.

We test RBP with CFR and CFR+. Experiments show that it leads to more than an order of magnitude speed improvement over partial pruning. The benefit increases with the size of the game.

2 Background

In this section we present the notation used in the rest of the paper. In an imperfect-information extensive-form game there is a finite set of players, $\mathcal{P}$. H is the set of all possible histories (nodes) in the game tree, represented as a sequence of actions, and includes the empty history. $A(h)$ is the actions available in a history and $P(h) \in \mathcal{P} \cup c$ is the player who acts at that history, where c denotes chance. Chance plays an action $a \in A(h)$ with a fixed probability $\sigma_c(h, a)$ that is known to all players. The history h' reached after an action is taken in h is a child of h, represented by $h \cdot a = h'$, while h is the parent of h'. More generally, h' is an ancestor of h (and h is a descendant of h'), represented by $h' \sqsubset h$, if there exists a sequence of actions from h' to h. $Z \subseteq H$ are terminal histories for which no actions are available. For each player $i \in \mathcal{P}$, there is a payoff function $u_i : Z \to \Re$. If $P = \{1, 2\}$ and $u_1 = -u_2$, the game is two-player zero-sum. We define $\Delta_i = \max_{z \in Z} u_i(z) - \min_{z \in Z} u_i(z)$ and $\Delta = \max_i \Delta_i$.

Imperfect information is represented by *information sets* for each player $i \in \mathcal{P}$ by a partition $\mathcal{I}_i$ of $h \in H : P(h) = i$. For any information set $I \in \mathcal{I}_i$, all histories $h, h' \in I$ are indistinguishable to player i, so $A(h) = A(h')$. $I(h)$ is the information set I where $h \in I$. $P(I)$ is the player i such that $I \in \mathcal{I}_i$. $A(I)$ is the set of actions such that for all $h \in I$, $A(I) = A(h)$. $|A_i| = \max_{I \in \mathcal{I}_i} |A(I)|$ and $|A| = \max_i |A_i|$. We define $U(I)$ to be the maximum payoff reachable from a history in I, and $L(I)$ to be the minimum. That is, $U(I) = \max_{z \in Z, h \in I : h \sqsubseteq z} u_{P(I)}(z)$ and $L(I) = \min_{z \in Z, h \in I : h \sqsubseteq z} u_{P(I)}(z)$. We define $\Delta(I) = U(I) - L(I)$ to be the range of payoffs reachable from a history in I. We similarly define $U(I, a)$, $L(I, a)$, and $\Delta(I, a)$ as the maximum, minimum, and range of payoffs (respectively) reachable from a history in I after taking action a. We define $D(I, a)$ to be the set of information sets reachable by player $P(I)$ after taking action a. Formally, $I' \in D(I, a)$ if for some history $h \in I$ and $h' \in I'$, $h \cdot a \sqsubseteq h'$ and $P(I) = P(I')$.

A strategy $\sigma_i(I)$ is a probability vector over $A(I)$ for player i in information set I. The probability of a particular action a is denoted by $\sigma_i(I, a)$. Since all histories in an information set belonging to player i are indistinguishable, the strategies in each of them must be identical. That is, for all $h \in I$, $\sigma_i(h) = \sigma_i(I)$ and $\sigma_i(h, a) = \sigma_i(I, a)$. We define σ_i to be a probability vector for player i over all available strategies Σ_i in the game. A strategy profile σ is a tuple of strategies, one for each player. $u_i(\sigma_i, \sigma_{-i})$ is the expected payoff for player i if all players play according to the strategy profile $\langle \sigma_i, \sigma_{-i} \rangle$. If a series of strategies are played over T iterations, then $\bar{\sigma}_i^T = \frac{\sum_{t \in T} \sigma_i^t}{T}$.

$\pi^\sigma(h) = \Pi_{h' \to a \sqsubseteq h} \sigma_{P(h)}(h, a)$ is the joint probability of reaching h if all players play according to σ. $\pi_i^\sigma(h)$ is the contribution of player i to this probability (that is, the probability of reaching h if all players other than i, and chance, always chose actions leading to h). $\pi_{-i}^\sigma(h)$ is the contribution of

all players other than i, and chance. $\pi^\sigma(h, h')$ is the probability of reaching h' given that h has been reached, and 0 if $h \not\sqsubset h'$. In a *perfect-recall* game, $\forall h, h' \in I \in \mathcal{I}_i$, $\pi_i(h) = \pi_i(h')$. In this paper we focus on perfect-recall games. Therefore, for $i = P(I)$ we define $\pi_i(I) = \pi_i(h)$ for $h \in I$. We define the average strategy $\bar{\sigma}_i^T(I)$ for an information set I to be

$$\bar{\sigma}_i^T(I) = \frac{\sum_{t \in T} \pi_i^{\sigma_i^t} \sigma_i^t(I)}{\sum_{t \in T} \pi_i^{\sigma^t}(I)} \tag{1}$$

2.1 Nash Equilibrium

A *best response* to σ_{-i} is a strategy σ_i^* such that $u_i(\sigma_i^*, \sigma_{-i}) = \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i})$. A *Nash equilibrium*, is a strategy profile where every player plays a best response. Formally, it is a strategy profile σ^* such that $\forall i$, $u_i(\sigma_i^*, \sigma_{-i}^*) = \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i}^*)$. We define a *Nash equilibrium strategy* for player i as a strategy σ_i that is part of any Nash equilibrium. In two-player zero-sum games, if σ_i and σ_{-i} are both Nash equilibrium strategies, then $\langle \sigma_i, \sigma_{-i} \rangle$ is a Nash equilibrium. An *ϵ-equilibrium* is a strategy profile σ^* such that $\forall i$, $u_i(\sigma_i^*, \sigma_{-i}^*) + \epsilon \geq \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i}^*)$.

2.2 Counterfactual Regret Minimization

Counterfactual Regret Minimization (CFR) is a popular regret-minimization algorithm for extensive-form games [16]. Our analysis of CFR makes frequent use of *counterfactual value*. Informally, this is the expected utility of an information set given that player i tries to reach it. For player i at information set I given a strategy profile σ, this is defined as

$$v_i^\sigma(I) = \sum_{h \in I} \left(\pi_{-i}^\sigma(h) \sum_{z \in Z} \left(\pi^\sigma(h, z) u_i(z) \right) \right) \tag{2}$$

The counterfactual value of an action a is

$$v_i^\sigma(I, a) = \sum_{h \in I} \left(\pi_{-i}^\sigma(h) \sum_{z \in Z} \left(\pi^\sigma(h \cdot a, z) u_i(z) \right) \right) \tag{3}$$

Let σ^t be the strategy profile used on iteration t. The *instantaneous regret* on iteration t for action a in information set I is

$$r^t(I, a) = v_{P(I)}^{\sigma^t}(I, a) - v_{P(I)}^{\sigma^t}(I) \tag{4}$$

and the *regret* for action a in I on iteration T is

$$R^T(I, a) = \sum_{t \in T} r^t(I, a) \tag{5}$$

Additionally, $R_+^T(I, a) = \max\{R^T(I, a), 0\}$ and $R^T(I) = \max_a \{R_+^T(I, a)\}$. Regret for player i in the entire game is

$$R_i^T = \max_{\sigma_i' \in \Sigma_i} \sum_{t \in T} \left(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma_i^t, \sigma_{-i}^t) \right) \tag{6}$$

In CFR, a player in an information set picks an action among the actions with positive regret in proportion to his positive regret on that action. Formally, on each iteration $T + 1$, player i selects actions $a \in A(I)$ according to probabilities

$$\sigma_i^{T+1}(I, a) = \begin{cases} \frac{R_+^T(I,a)}{\sum_{a' \in A(I)} R_+^T(I,a')}, & \text{if } \sum_{a' \in A_i} R_+^T(I, a') > 0 \\ \frac{1}{|A(I)|}, & \text{otherwise} \end{cases} \tag{7}$$

If a player plays according to CFR in every iteration, then on iteration T, $R^T(I) \leq \Delta_i \sqrt{|A(I)|} \sqrt{T}$. Moreover,

$$R_i^T \leq \sum_{I \in \mathcal{I}_i} R^T(I) \leq |\mathcal{I}_i| \Delta_i \sqrt{|A_i|} \sqrt{T} \tag{8}$$

So, as $T \to \infty$, $\frac{R_i^T}{T} \to 0$. In two-player zero-sum games, if both players' average regret $\frac{R_i^T}{T} \leq \epsilon$, their average strategies $\langle \bar{\sigma}_1^T, \bar{\sigma}_2^T \rangle$ form a 2ϵ-equilibrium [15]. Thus, CFR constitutes an anytime algorithm for finding an ϵ-Nash equilibrium in zero-sum games.

3 Applying Best Response to Zero-Reach Sequences

In Section 2 it was explained that if both players' average regret approaches zero, then their average strategies approach a Nash equilibrium. CFR provides one way to compute strategies that have bounded regret, but it is not the only way. CFR-BR [7] is a variant of CFR in which one player plays CFR and the other player plays a best response to the opponent's strategy in every iteration. Calculating a best response to a fixed strategy is computationally cheap (in games of perfect recall), costing only a single traversal of the game tree. By playing a best response in every iteration, the best-responder is guaranteed to have at most zero regret. Moreover, the CFR player's regret is still bounded according to (8). However, in practice the CFR player's regret in CFR-BR tends to be higher than when both players play vanilla CFR (since the opponent is clairvoyantly maximizing the CFR player's regret). For this reason, empirical results show that CFR-BR converges slower than CFR, even though the best-responder's regret is always at most zero.

We now discuss a modification of CFR that will motivate the main contribution of this paper, which, in turn, is described in Section 4. The idea is that *by applying a best response only in certain situations* (and CFR in others), we can lower regret for one player without increasing it for the opponent. Without loss of generality, we discuss how to reduce regret for Player 1. Specifically, consider an information set $I \in \mathcal{I}_1$ and action a where $\sigma_1^t(I, a) = 0$ and any history $h \in I$. Then for any ancestor history h' such that $h' \sqsubset h \cdot a$, we know $\pi_1^{\sigma^t}(h', h \cdot a) = 0$. Likewise, for any descendant history h' such that $h \cdot a \sqsubseteq h'$, we know $\pi_1^{\sigma^t}(h') = 0$. Thus, from (4) we see that Player 1's strategy on iteration t in any information set following action a has no effect on Player 2's regret for that iteration. Moreover, it also has no effect on Player 1's regret for any information set except $R(I, a)$ and information sets that follow action a. Therefore, by playing a best response *only* in information sets following action a (and playing vanilla CFR elsewhere), Player 1 guarantees zero regret for himself in all information sets following action a, without the practical cost of increasing his regret in information sets before I or of increasing Player 2's regret. This may increase regret for action a itself, but if we only do this when $R(I, a) \leq -\Delta(I)$, we can guarantee $R(I, a) \leq 0$ even after the iteration. Similarly, Player 2 can simultaneously play a best response in information sets following an action a' where $\sigma_2^t(I', a') = 0$ for $I' \in \mathcal{I}_2$. This approach leads to lower regret for both players.

(In situations where *both* players' sequences of reaching an information set have zero probability ($\pi_1(h) = \pi_2(h) = 0$) the strategies chosen have no impact on the regret or average strategy for either player, so there is no need to compute what strategies should be played from then on.)

Our experiments showed that this technique leads to a dramatic improvement over CFR in terms of the number of iterations needed—though the theoretical convergence bound remains the same. However, each iteration touches more nodes—because negative-regret actions more quickly become positive and are not skipped with partial pruning—and thus takes longer. It depends on the game whether CFR or this technique is faster overall; see experiments in Appendix A. Regret-based pruning, introduced in the next section, outperforms both of these approaches significantly.

4 Regret-Based Pruning (RBP)

In this section we present the main contribution of this paper, a technique for soundly pruning—on a temporary basis—negative-regret actions from the tree traversal in order to speed it up significantly. In Section 3 we proposed a variant of CFR where a player plays a best response in information sets that the player reaches with zero probability. In this section, we show that these information sets and their descendants need not be traversed in every iteration. Rather, the frequency that they must be traversed is proportional to how negative regret is for the action leading to them. This less-frequent traversal does not hurt the regret bound (8). Consider an information set $I \in \mathcal{I}_1$ and action a where $R^t(I, a) = -1000$ and regret for at least one other action in I is positive, and assume $\Delta(I) = 1$. From (7), we see that $\sigma_1^{t+1}(I, a) = 0$. As described in Section 3, the strategy played by Player 1 on iteration $t + 1$ in any information set following action a has no effect on Player 2. Moreover, it has no immediate effect on what Player 1 will do in the next iteration (other than in information sets following action a), because we know regret for action a will still be at most -999 on iteration $t + 2$ (since $\Delta(I) = 1$) and will continue to not be played. So rather than traverse the game tree following action a, we could "procrastinate" in deciding what Player 1 did on iteration $t+1, t+2, ..., t+1000$

in that branch until after iteration $t + 1000$ (at which point regret for that action may be positive). That is, we could (in principle) store Player 2's strategy for each iteration between $t+1$ and $t+1000$, and on iteration $t+1000$ calculate a best response to each of them and announce that Player 1 played those best responses following action a on iterations $t + 1$ to $t + 1000$ (and update the regrets to match this). Obviously this itself would not be an improvement, but performance would be identical to the algorithm described in Section 3.

However, rather than have Player 1 calculate and play a best response for each iteration between $t+1$ and $t+1000$ separately, we could simply calculate a best response against the average strategy that Player 2 played in those iterations. This can be accomplished in a single traversal of the game tree. We can then announce that Player 1 played this best response on each iteration between $t + 1$ and $t + 1000$. This provides benefits similar to the algorithm described in Section 3, but allows us to do the work of 1000 iterations in a single traversal! We coin this *regret-based pruning (RBP)*.

We now present a theorem that guarantees that when $R(I, a) \leq 0$, we can prune $D(I, a)$ through regret-based pruning for $\lfloor \frac{|R(I,a)|}{U(I,a)-L(I)} \rfloor$ iterations.

Theorem 1. *Consider a two-player zero-sum game. Let $a \in A(I)$ be an action such that on iteration T_0, $R^{T_0}(I, a) \leq 0$. Let I' be an information set for any player such that $I' \notin D(I, a)$ and let $a' \in A(I')$. Let $m = \lfloor \frac{|R(I,a)|}{U(I,a)-L(I)} \rfloor$. If $\sigma(I, a) = 0$ when $R(I, a) \leq 0$, then regardless of what is played in $D(I, a)$ during $\{T_0, ..., T_0 + m\}$, $R^T_+(I', a')$ is identical for $T \leq T_0 + m$.*

Proof. Since $v^\sigma_i(I) \geq L(I)$ and $v^\sigma_i(I, a) \leq U(I, a)$, so from (4) we get $r^t(I, a) \leq U(I, a) - L(I)$. Thus, for iteration $T_0 \leq T \leq T_0 + m$, $R^T(I, a) \leq 0$. Clearly the theorem is true for $T < T_0$. We prove the theorem continues to hold inductively for $T \leq T_0 + m$. Assume the theorem holds for iteration T and consider iteration $T + 1$. Suppose $I' \in \mathcal{I}_{P(I)}$ and either $I' \neq I$ or $a' \neq a$. Then for any $h' \in I'$, there is no ancestor of h' in an information set in $D(I, a)$. Thus, $\pi^{\sigma^{T+1}}_{-i}(h')$ does not depend on the strategy in $D(I, a)$. Moreover, for any $z \in Z$, if $h' \sqsubset h \sqsubset z$ for some $h \in I^* \in D(I, a)$, then $\pi^{\sigma^{T+1}}(h', z) = 0$ because $\sigma^{T+1}(I, a) = 0$. Since $I' \neq I$ or $a' \neq a$, it similarly holds that $\pi^{\sigma^{T+1}}(h' \cdot a', z) = 0$. Then from (4), $r^{T+1}(I, a)$ does not depend on the strategy in $D(I, a)$.

Now suppose $I' \in \mathcal{I}_i$ for $i \neq P(I)$. Consider some $h' \in I'$ and some $h \in I$. First suppose that $h \cdot a \sqsubseteq h'$. Since $\pi^{\sigma^{T+1}}_i(h \cdot a) = 0$, so $\pi^{\sigma^{T+1}}_i(h') = 0$ and h' contributes nothing to the regret of I'. Now suppose $h' \sqsubset h$. Then for any $z \in Z$, if $h' \sqsubset h \sqsubset z$ then $\pi^{\sigma^{T+1}}(h', z) = 0$ and does not depend on the strategy in $D(I, a)$. Finally, suppose $h' \not\sqsubset h$ and $h \cdot a \not\sqsubseteq h'$. Then for any $z \in Z$ such that $h' \sqsubset z$, we know $h \not\sqsubset z$ and therefore $\pi^{\sigma^{T+1}}(h', z) = 0$ does not depend on the strategy in $D(I, a)$.

Now suppose $I' = I$ and $a' = a$. We proved $R^T(I, a) \leq 0$ for $T_0 \leq T \leq T_0 + m$, so $R^T_+(I, a) = 0$. Thus, for all $T \leq T_0 + m$, $R^T(I', a')$ is identical regardless of what is played in $D(I, a)$. □

We can improve this approach significantly by not requiring knowledge *beforehand* of exactly how many iterations can be skipped. Rather, we will *decide in light of what happens during the intervening CFR iterations when an action needs to be revisited.* From (4) we know that $r^T(I, a) \propto \pi^{\sigma^T}_{-i}(I)$. Moreover, $v^{\sigma^T}_{P(I)}(I)$ does not depend on $D(I, a)$. Thus, we can prune $D(I, a)$ from iteration T_0 until iteration T_1 so long as

$$\sum_{t=1}^{T_0} v^{\sigma^t}_{P(I)}(I, a) + \sum_{t=T_0+1}^{T_1} \pi^{\sigma^t}_{-i}(I) U(I, a) \leq \sum_{t=1}^{T_1} v^{\sigma^t}_{P(I)}(I) \tag{9}$$

In the worst case, this allows us to skip only $\lfloor \frac{R(I,a)}{U(I,a)-L(I)} \rfloor$ iterations. However, in practice it performs significantly better, though we cannot know on iteration T_0 how many iterations it will skip because it depends on what is played in $T_0 \leq t \leq T_1$. Our exploratory experiments showed that in practice performance also improves by replacing $U(I, a)$ with a more accurate upper bound on reward in (9). CFR will still converge if $D(I, a)$ is pruned for too many iterations; however, that hurts convergence speed. In the experiments included in this paper, we conservatively use $U(I, a)$ as the upper bound.

4.1 Best Response Calculation for Regret-Based Pruning

In this section we discuss how one can efficiently compute the best responses as called for in regret-based pruning. The advantage of Theorem 1 is that we can wait until after pruning has finished—that is, until we revisit an action—to decide what strategies were played in $D(I, a)$ during the intervening iterations. We can then calculate a single best response to the average strategy that the opponent played, and say that that best response was played in $D(I, a)$ in each of the intervening iterations. This results in zero regret over those iterations for information sets in $D(I, a)$. We now describe how this best response can be calculated efficiently.

Typically, when playing CFR one stores $\sum_{t=1}^{T} \pi_i^t(I)\sigma_i^t(I)$ for each information set I. This allows one to immediately calculate the average strategy defined in (1) in any particular iteration. If we start pruning on iteration T_0 and revisit on iteration T_1, we wish to calculate a best response to $\bar{\sigma}_i^{T_1-T_0}$ where $\bar{\sigma}_i^{T_1-T_0}(I) = \frac{\sum_{t=T_0}^{T_1} \pi_i^t(I)\sigma_i^t(I)}{\sum_{t=T_0}^{T_1} \pi_i^t(I)}$. An easy approach would be to store the opponent's cumulative strategy before pruning begins and subtract it from the current cumulative strategy when pruning ends. In fact, we only need to store the opponent's strategy in information sets that follow action a. However, this could potentially use $O(H)$ memory because the same information set I belonging to Player 2 may be reached from multiple information sets belonging to Player 1. In contrast, CFR only requires $O(|\mathcal{I}||A|)$ memory, and we want to maintain this desirable property. We accomplish that as follows.

To calculate a best response against $\bar{\sigma}_2^T$, we traverse the game tree and calculate the counterfactual value, defined in (3), for every action for every information set belonging to Player 1 that does not lead to any further Player 1 information sets. Specifically, we calculate $v_1^{\bar{\sigma}^{T_0-1}}(I, a)$ for every action a in I such that $D(I, a) = \emptyset$. Since we calculate this only for actions where $D(I, a) = \emptyset$, so $v_1^{\bar{\sigma}^{T_0-1}}(I, a)$ does not depend on $\bar{\sigma}_1$. Then, starting from the bottom information sets, we set the best-response strategy $\sigma_1^{BR}(I)$ to always play the action with the highest counterfactual value (ties can be broken arbitrarily), and pass this value up as the payoff for reaching I, repeating the process up the tree. In order to calculate a best response to $\bar{\sigma}_2^{T_1-T_0}$, we first store, before pruning begins, the counterfactual values for Player 1 against Player 2's average strategy for every action a in each information set I where $D(I, a) = \emptyset$. When we revisit the action on iteration T_1, we calculate a best response to $\bar{\sigma}_2^{T_1}$ except that we set the counterfactual value for every action a in information set I where $D(I, a) = \emptyset$ to be $T_1 v_1^{\bar{\sigma}^{T_1}}(I, a) - (T_0 - 1)v_1^{\bar{\sigma}^{T_0-1}}(I, a)$. The latter term was stored, and the former term can be calculated from the current average strategy profile. As before, we set $\sigma_1^{BR}(I)$ to always play whichever action has the highest counterfactual value, and pass this term up.

A slight complication arises when we are pruning an action a in information set I and wish to start pruning an earlier action a' from information set I' such that $I \in D(I', a')$. In this case, it is necessary to explore action a in order to calculate the best response in $D(I', a')$. However, if such traversals happen frequently, then this would defeat the purpose of pruning action a. One way to address this is to only prune an action a' when the number of iterations guaranteed (or estimated) to be skipped exceeds some threshold. This ensures that the overhead is worthwhile, and that we are not frequently traversing an action a farther down the tree that is already being pruned. Another option is to add some upper bound to how long we will prune an action. If the lower bound for how long we will prune a exceeds the upper bound for how long we will prune a', then we need not traverse a in the best response calculation for a' because a will still be pruned when we are finished with pruning a'. In our experiments, we use the former approach. Experiments to determine a good parameter for this are presented in Appendix B.

4.2 Regret-Based Pruning with CFR+

CFR+ [13] is a variant of CFR where the regret is never allowed to go below 0. Formally, $R^T(I, a) = \max\{R^{T-1}(I, a) + r^T(I, a), 0\}$ for $T \geq 1$ and $R^T(I, a) = 0$ for $T = 0$. Although this change appears small, and does not improve the bound on regret, it leads to faster empirical convergence. CFR+ was a key advancement that allowed Limit Texas Hold'em poker to be essentially solved [1].

At first glance, it would seem that CFR+ and RBP are incompatible. RBP allows actions to be traversed with decreasing frequency as regret decreases below zero. However, CFR+ sets a floor

for regret at zero. Nevertheless, it is possible to combine the two, as we now show. We modify the definition of regret in CFR+ so that it can drop below zero, but *immediately returns to being positive as soon as regret begins increasing*. Formally, we modify the definition of regret in CFR+ for $T > 0$ to be as follows: $R^T(I,a) = r^T(I,a)$ if $r^T(I,a) > 0$ and $R^{T-1}(I,a) \leq 0$, and $R^T(I,a) = R^{T-1}(I,a) + r^T(I,a)$ otherwise. This leads to identical behavior in CFR+, and also allows regret to drop below zero so actions can be pruned.

When using RBP with CFR+, regret does not strictly follow the rules for CFR+. CFR+ calls for an action to be played with positive probability whenever instantaneous regret for it is positive in the previous iteration. Since RBP only checks the regret for an action after potentially several iterations have been skipped, there may be a delay between the iteration when an action would return to play in CFR+ and the iteration when it returns to play in RBP. This does not pose a theoretical problem: CFR's convergence rate still applies.

However, this difference is noticeable when combined with *linear averaging*. Linear averaging weighs each iteration σ^t in the average strategy by t. It does not affect regret or influence the selection of strategies on an iteration. That is, with linear averaging the new definition for average strategy becomes $\bar{\sigma}_i^T(I) = \frac{\sum_{t \in T}(t\pi_i^{\sigma_i^t}\sigma_i^t)}{\sum_{t \in T}(t\pi_i^{\sigma_i^t})}$. Linear averaging still maintains the asymptotic convergence rate of constant averaging (where each iteration is weighed equally) in CFR+ [14]. Empirically it causes CFR+ to converge to a Nash equilibrium much faster. However, in vanilla CFR it results in worse performance and there is no proof guaranteeing convergence. Since RBP with CFR+ results in behavior that does not strictly conform to CFR+, linear averaging results in somewhat noisier convergence. This can be mitigated by reporting the strategy profile found so far that is closest to a Nash equilibrium rather than the current average strategy profile, and we do this in the experiments.

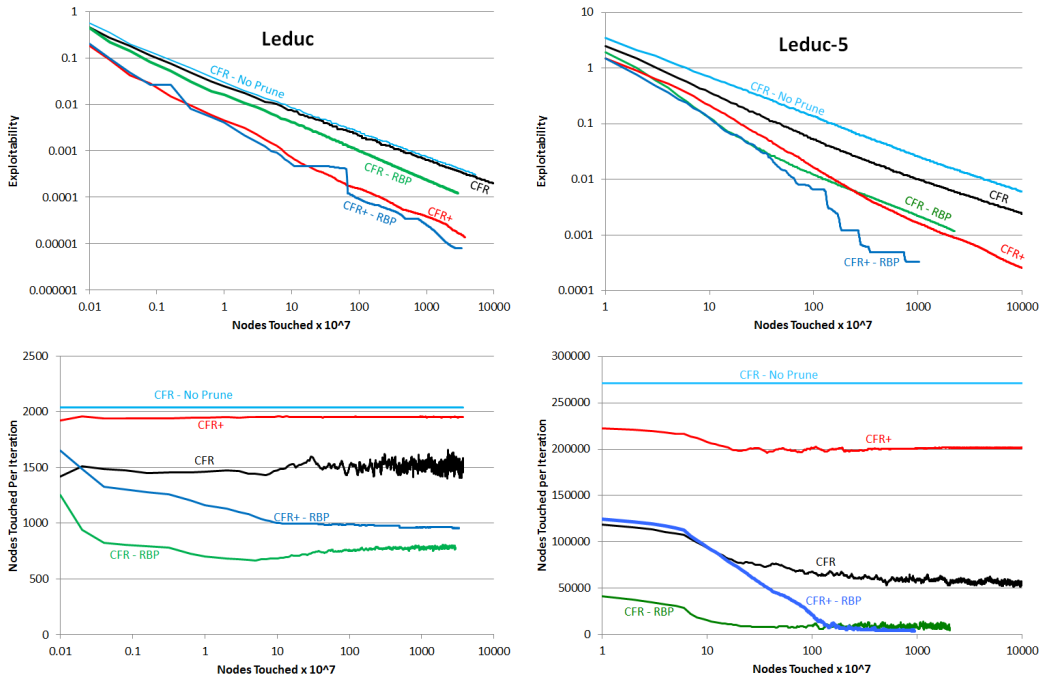
5 Experiments

We tested regret-based pruning in both CFR and CFR+ against partial pruning, as well as against CFR with no pruning. Our implementation traverses the game tree once each iteration.[1] We tested our algorithm on standard Leduc Hold'em [12] and a scaled-up variant of it featuring more actions. Leduc Hold'em is a popular benchmark problem for imperfect-information game solving due to its size (large enough to be highly nontrivial but small enough to be solvable) and strategic complexity. In Leduc Hold'em, there is a deck consisting of six cards: two each of Jack, Queen, and King. There are two rounds. In the first round, each player places an ante of 1 chip in the pot and receives a single private card. A round of betting then takes place with a two-bet maximum, with Player 1 going first. A public shared card is then dealt face up and another round of betting takes place. Again, Player 1 goes first, and there is a two-bet maximum. If one of the players has a pair with the public card, that players wins. Otherwise, the player with the higher card wins. In standard Leduc Hold'em, the bet size in the first round is 2 chips, and 4 chips in the second round. In our scaled-up variant, which we call *Leduc-5*, there are 5 bet sizes to choose from: in the first round a player may bet 0.5, 1, 2, 4, or 8 chips, while in the second round a player may bet 1, 2, 4, 8, or 16 chips.

We measure the quality of a strategy profile by its *exploitability*, which is the summed ϵ distance of both players from a Nash equilibrium strategy. Formally, exploitability of a strategy profile σ is $\max_{\sigma_1^* \in \Sigma_1} u_1(\sigma_1^*, \sigma_2) + \max_{\sigma_2^* \in \Sigma_2} u_2(\sigma_1, \sigma_2^*)$. We measure exploitability against the number of nodes touched over all CFR traversals. As shown in Figure 1, RBP leads to a substantial improvement over vanilla CFR with partial pruning in Leduc Hold'em, increasing the speed of convergence by more than a factor of 8. This is partially due to the game tree being traversed twice as fast, and partially due to the use of a best response in sequences that are pruned (the benefit of which was described in Section 3). The improvement when added on top of CFR+ is smaller, increasing the speed of convergence by about a factor of 2. This matches the reduction in game tree traversal size.

The benefit from RBP is more substantial in the larger benchmark game, Leduc-5. RBP increases convergence speed of CFR by a factor of 12, and reduces the per-iteration game tree traversal cost by about a factor of 7. In CFR+, RBP improves the rate of convergence by about an order of magnitude. RBP also decreases the number of nodes touched per iteration in CFR+ by about a factor of 40.

[1] Canonical CFR+ traverses the game tree twice each iteration, updating the regrets for each player in separate traversals [13]. This difference does not, however, affect the error measure (y-axis) in the experiments.



(a) Leduc Hold'em

(b) Leduc-5 Hold'em

Figure 1: Top: Exploitability. Bottom: Nodes touched per iteration.

The results imply that larger games benefit more from RBP than smaller games. This is not universally true, since it is possible to have a large game where every action is part of the Nash equilibrium. Nevertheless, there are many games with very large action spaces where the vast majority of those actions are suboptimal, but players do not know beforehand which are suboptimal. In such games, RBP would improve convergence tremendously.

6 Conclusions and Future Research

In this paper we introduced a new method of pruning that allows CFR to avoid traversing high-regret actions in every iteration. Our regret-based pruning (RBP) temporarily ceases their traversal in a sound way without compromising the overall convergence rate. Experiments show an order of magnitude speed improvement over partial pruning, and suggest that the benefit of RBP increases with game size. Thus RBP is particularly useful in large games where many actions are suboptimal, but where it is not known beforehand which actions those are.

In future research, it would be worth examining whether similar forms of pruning can be applied to other equilibrium-finding algorithms as well. RBP, as presented in this paper, is for CFR using regret matching to determine what strategies to use on each iteration based on the regrets. RBP does not directly apply to other strategy selection techniques that could be used within CFR such as exponential weights, because the latter always puts positive probability on actions. Also, it would be interesting to see whether RBP-like pruning could be applied to first-order methods for equilibrium-finding [5, 3, 10, 8]. The results in this paper suggest that for any equilibrium-finding algorithm to be efficient in large games, effective pruning is essential.

6.1 Acknowledgement

This material is based on work supported by the National Science Foundation under grants IIS-1320620 and IIS-1546752, as well as XSEDE computing resources provided by the Pittsburgh Supercomputing Center.

References

[1] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold-dem poker is solved. *Science*, 347(6218):145–149, 2015.

[2] Noam Brown, Sam Ganzfried, and Tuomas Sandholm. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit texas hold'em agent. In *Proceedings of the 2015 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[3] Andrew Gilpin, Javier Peña, and Tuomas Sandholm. First-order algorithm with $\mathcal{O}(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games. *Mathematical Programming*, 133(1–2):279–298, 2012. Conference version appeared in AAAI-08.

[4] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM*, 54(5), 2007. Early version 'Finding equilibria in large sequential games of imperfect information' appeared in the Proceedings of the ACM Conference on Electronic Commerce (EC), pages 160–169, 2006.

[5] Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010. Conference version appeared in WINE-07.

[6] Eric Griffin Jackson. A time and space efficient algorithm for approximately solving large imperfect information games. In *AAAI Workshop on Computer Poker and Imperfect Information*, 2014.

[7] Michael Johanson, Nolan Bard, Neil Burch, and Michael Bowling. Finding optimal abstract strategies in extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[8] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. Faster first-order methods for extensive-form game solving. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2015.

[9] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1078–1086, 2009.

[10] François Pays. An interior point approach to large games of incomplete information. In *AAAI Computer Poker Workshop*, 2014.

[11] Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, pages 13–32, Winter 2010. Special issue on Algorithmic Game Theory.

[12] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 550–558, July 2005.

[13] Oskari Tammelin. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042*, 2014.

[14] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit texas holdem. In *IJCAI*, volume 2015, 2015.

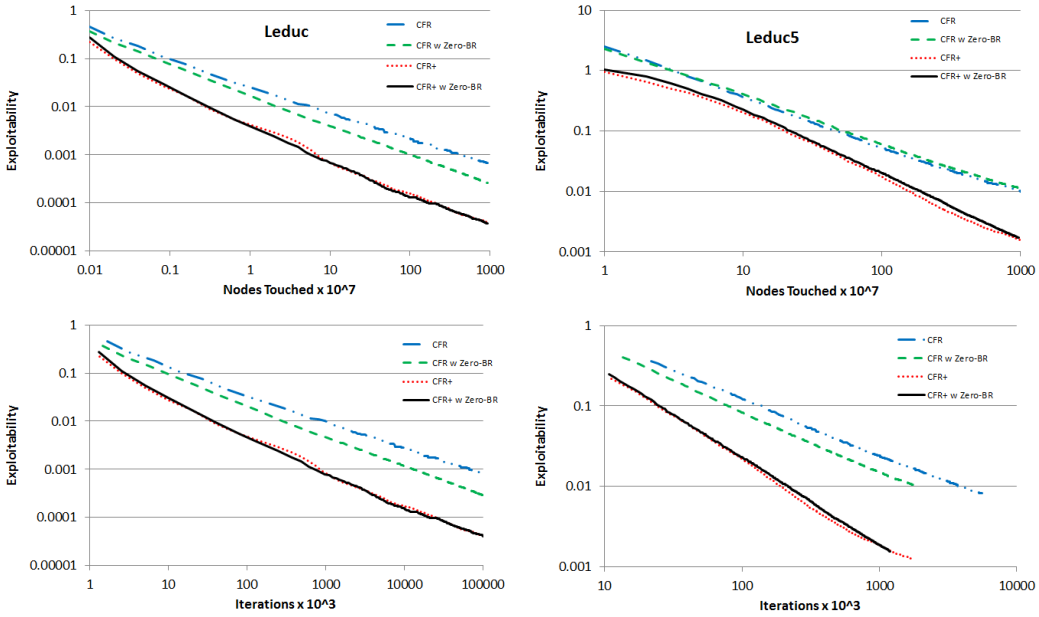
[15] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2009.

[16] Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

Appendix

A Experiments on Zero-Reach Best Response

Section 3 introduces a variant of CFR in which players play a best response only in sequences that they reach with zero probability. Figure 2 shows experimental results comparing this variant to CFR and CFR+. The figure shows that it reduces the number of iterations needed for convergence. However, in Leduc-5, the number of nodes touched (which is a better measure of time required), is actually slightly higher for a given level of convergence. In both games, playing a best response in zero-reach sequences led to no substantial difference when combined with CFR+.



(a) Leduc Hold'em

(b) Leduc-5 Hold'em

Figure 2: Top: Exploitability vs Nodes Touched. Bottom: Exploitability vs Iterations.

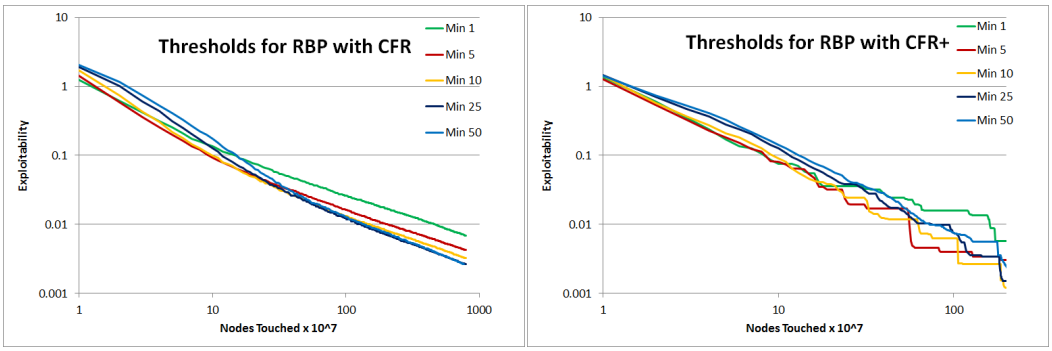
B Comparison of Minimum Skip Thresholds

As discussed in Section 4.1, it may be worthwhile to establish some minimum threshold for the anticipated number of iterations to be skipped in order to prune an action. This ensures that the overhead of pruning is worth the gain, and also prevents descendant actions that are already pruned from being repeatedly traversed. Setting such a threshold does not affect the theoretical guarantees of the algorithm, and may lead to better empirical performance.

We now describe how to set such a threshold. We estimate the number of iterations that will be skipped if we prune an action by modifying (9) to assume the information set will continue to be reached by the opponent as often as it has, on average, in the past. We also modify the equation to assume that $v^{\sigma^t}_{P(I)}(I)$ will be the average that has been received in the past. This allows us to solve the equation for the number of iterations we estimate will be skipped. If this estimated number of iterations exceeds some minimum threshold, then we proceed with pruning the action. Formally, we calculate the estimate as

$$\hat{T}_1 - T_0 = \frac{R^{T_0}(I,a)}{\frac{\sum_{t=1}^{T_0} v^{\sigma^t}(I)}{T_0} - \frac{\sum_{t=1}^{T_0} \pi^{\sigma^t}_{-i} U(I,a)}{T_0}} \tag{10}$$

In Figure 3, we compare 1, 5, 10, 25, and 50 as possible thresholds in Leduc-5 for regret-based pruning in CFR and CFR+. All of the options performed similarly, suggesting that the algorithm is not very sensitive to the parameter chosen. The experiments show that a low threshold is preferable early on, while a higher threshold leads to better performance later. The long-term gain of increasing the threshold appears to quickly diminish however, as there is only a slight long-term difference between a threshold of 10 iterations and a threshold of 50 iterations. In the experiments presented in Section 5, we use a threshold of 25 iterations for RBP in all cases.



(a) Leduc Hold'em

(b) Leduc-5 Hold'em

Figure 3: A comparison of minimum thresholds for estimated number of iterations pruned for RBP in CFR (left) and CFR+ (right)