Finding Optimal Abstract Strategies in Extensive-Form Games

Michael Johanson and **Nolan Bard** and **Neil Burch** and **Michael Bowling**
{johanson,nbard,nburch,mbowling}@ualberta.ca
University of Alberta, Edmonton, Alberta, Canada

Abstract

Extensive-form games are a powerful model for representing interactions between agents. Nash equilibrium strategies are a common solution concept for extensive-form games and, in two-player zero-sum games, there are efficient algorithms for calculating such strategies. In large games, this computation may require too much memory and time to be tractable. A standard approach in such cases is to apply a lossy state-space abstraction technique to produce a smaller abstract game that can be tractably solved, while hoping that the resulting abstract game equilibrium is close to an equilibrium strategy in the unabstracted game. Recent work has shown that this assumption is unreliable, and an arbitrary Nash equilibrium in the abstract game is unlikely to be even near the least suboptimal strategy that can be represented in that space. In this work, we present for the first time an algorithm which efficiently finds optimal abstract strategies — strategies with minimal exploitability in the unabstracted game. We use this technique to find the least exploitable strategy ever reported for two-player limit Texas hold'em.

Introduction

Extensive-form games are a general model of multiagent interaction. They have been used to model a variety of scenarios including game playing (Zinkevich et al. 2008; Lanctot et al. 2009; Hoda et al. 2010; Risk and Szafron 2010), bargaining and negotiation (Lazaric, de Cote, and Gatti 2007; Gatti 2008), argumentation (Procaccia and Rosenschein 2005), and even distributed database management (Mostafa, Lesser, and Miklau 2008). Strategic reasoning in all but the simplest such models has proven computationally challenging beyond certain special cases. Even the most theoretically-straightforward setting of two-player, zero-sum extensive-form games presents obstacles for finding approximate solutions for human-scale interactions (e.g., two-player, limit Texas hold'em with its 10^{18} game states). These obstacles include the recently discovered existence of abstraction pathologies (Waugh et al. 2009a) and a form of abstract game "overfitting" (Johanson et al. 2011). This paper presents the first technique for overcoming these abstraction challenges in the two-player, zero-sum setting.

Abstraction, first suggested by Billings and colleagues (2003), is the dominant approach for handling massive extensive-form imperfect information games and is used by the majority of top competitors in the Annual Computer Poker Competition (Sandholm 2010). The approach involves constructing an abstract game by aggregating each player's states (*i.e.*, information sets) into abstract game states (Gilpin, Sandholm, and Sørensen 2007; Zinkevich et al. 2008). An ϵ-Nash equilibrium is computed in the abstract game, and that strategy is then employed in the original game. As equilibrium computation algorithms improve or computational resources become available, a refined, less abstract but larger, game can be solved instead. This improvement, as larger and larger abstract games are solved, has appeared to drive much of the advancement in the Annual Computer Poker Competitions (Sandholm 2010).

However, recent work by Waugh et al. (2009a) showed that solving more refined abstractions is not always better by presenting examples of **abstraction pathologies** in toy poker games. They showed that even when considering strict refinements of an abstraction (*i.e.*, one capable of representing a strictly larger set of strategies), the equilibria found in this finer-grained abstraction could be dramatically worse approximations than equilibria in the coarser abstraction. Furthermore, their experiments showed that while an abstraction may be able to represent good approximations of real game equilibria, these good abstract strategies may not be abstract game equilibria.

A recent publication presented a technique for efficiently computing best-responses in very large extensive-form games (Johanson et al. 2011). This made it possible to investigate Waugh's findings in the context of full two-player limit Texas hold'em. While abstraction pathologies were not found to be common using typical abstraction techniques, it was discovered that equilibrium learning methods, such as Counterfactual Regret Minimization (CFR) (Zinkevich et al. 2008), can "overfit": as the approximation gets more exact in the abstract game, its approximation of the full-game equilibrium can worsen (see Figure 1).

Combined, these results present a rather bleak picture. It is unclear how to use more computational power to better approximate a Nash equilibrium in massive extensive-form games. Furthermore, our current abstractions are likely able

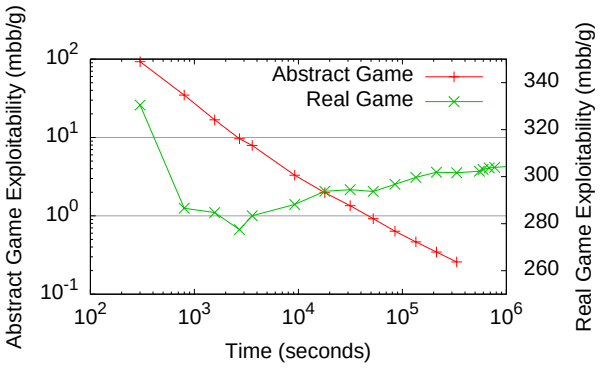


Figure 1: Abstract-game and real-game exploitability of strategies generated by the CFR algorithm.

to represent better approximations than our current methods actually compute. In this paper, we present the first algorithm that avoids abstraction pathologies and overfitting entirely. Essentially, the approach leaves one player unabstracted and finds the best possible abstract strategy for the other player. It avoids the memory requirements for solving for an unabstracted opponent by having the opponent employ a best-response strategy on each iteration rather than a no-regret strategy. It then uses sampling tricks to avoid the computational requirements needed to compute an exact best-response on each iteration. The resulting algorithm, CFR-BR, finds optimal abstract strategies, i.e, the best-approximation to a Nash equilibrium that can be represented within a chosen strategy abstraction. Consequently, it is not subject to abstraction pathologies or overfitting. We demonstrate the approach in two-player limit Texas hold'em, showing that it indeed finds dramatically better Nash equilibrium approximations than CFR with the same abstraction. We use the technique to compute the least exploitable strategy ever reported for this game.

Background

We begin with some formalism for extensive-form games and the counterfactual regret minimization algorithm.

Extensive-Form Games. For a complete description see (Osborne and Rubinstein 1994). Extensive-form games provide a general model for domains with multiple agents making decisions sequentially. They can be viewed as a game tree that consists of nodes corresponding to histories of the game and edges between nodes being actions taken by agents or by the environment. Therefore each **history** $h \in H$ corresponds to a past sequence of actions from the set of **players**, N, and **chance**, c. For each non-terminal history h, the acting player $P(h) \in N \cup \{c\}$ selects an **action** a from $A(h)$, the set of actions available at h. We call h' a **prefix** of h, written as $h' \sqsubseteq h$, if h begins with h'. Each terminal history $z \in Z$ has a **utility** associated with it for each player i, $u_i(z)$. If $\sum_{i \in N} u_i(z) = 0$ then the game is **zero-sum**. This work focuses on two-player, zero-sum games (*i.e.*, $u_1(z) = -u_2(z)$). Let $\Delta_i = \max_{z \in Z} u_i(z) - \min_{z \in Z} u_i(z)$, be the range of utilities for player i. In our case, a two-player zero-sum game, Δ_i is the same for both players and so we refer to it simply as Δ.

In **imperfect information** games, actions taken by the players or by chance may not be observable by all of the other players. Extensive games model imperfect information by partitioning the histories where each player acts into **information sets**. For each information set $I \in \mathcal{I}_i$, player i cannot distinguish between the histories in I. It is required that $A(h)$ must equal $A(h')$ for all $h, h' \in I$, so we can denote the actions available at an information set as $A(I)$. Furthermore, we generally require the information partition to satisfy **perfect recall**, i.e., all players are able to distinguish histories previously distinguishable or in which they took a different sequence of actions. Poker is an example of an imperfect information game since chance acts by dealing cards privately to the players. Since player i cannot see the cards of the other players, histories where only the cards of i's opponents differ are in the same information set.

A **strategy** for player i, $\sigma_i \in \Sigma_i$, maps each information set $I \in \mathcal{I}_i$ to a probability distribution over the actions $A(I)$. The **average strategy**, $\bar{\sigma}_i^t$, of the strategies $\sigma_i^1, \ldots, \sigma_i^t$ defines $\bar{\sigma}_i^t(I)$ as the average of $\sigma_i^1(I), \ldots, \sigma_i^t(I)$ weighted by each strategy's probability of reaching I (Zinkevich et al. 2008, Equation 4). A **strategy profile**, $\sigma \in \Sigma$, is a vector of strategies $(\sigma_1, \ldots, \sigma_{|N|})$. We let σ_{-i} refer to the strategies in σ except for σ_i. Given a strategy profile, we define player i's **expected utility** as $u_i(\sigma)$ or, since we are using two-player games, $u_i(\sigma_1, \sigma_2)$. We define $b_i(\sigma_{-i}) = \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i})$ to be the **best response value** for player i against their opponents σ_{-i} (a **best response** is the argmax). A strategy profile σ is an **ϵ-Nash equilibrium** if no player can gain more than ϵ by unilaterally deviating from σ. That is, if $b_i(\sigma_{-i}) \leq u_i(\sigma_i, \sigma_{-i}) + \epsilon$, for all $i \in N$. If this holds when $\epsilon = 0$, then all players are playing a best response to σ_{-i}, and this is called a **Nash equilibrium**. In two-player zero-sum games, we define the **game value**, v_i, for each player i to be the unique value of $u_i(\sigma^*)$ for any Nash equilibrium profile σ^*. Finally, in two-player zero-sum games we define $\varepsilon_i(\sigma_i) = b_{-i}(\sigma_i) - v_{-i}$ to be the **exploitability** of strategy σ_i, and $\varepsilon(\sigma) = (\varepsilon_1(\sigma_1) + \varepsilon_2(\sigma_2))/2 = (b_1(\sigma_2) + b_2(\sigma_1))/2$ to be the exploitability (or best response value) of the strategy profile σ. This measures the quality of an approximation to a Nash equilibrium profile, as Nash equilibria have an exploitability of 0.

Counterfactual Regret Minimization. CFR (Zinkevich et al. 2008) is a state-of-the-art algorithm for approximating Nash equilibria in two-player, zero-sum, perfect-recall games. It is an iterative algorithm that resembles self-play. Two strategies, one for each player, are represented in memory and initialized arbitrarily. In each iteration, the strategies are evaluated with respect to each other and updated so as to minimize a weighted form of regret at each decision: the difference in utility between the actions currently being selected and the best action in retrospect. Over a series of iterations, the average strategy for the players approaches a Nash equilibrium. As our algorithm builds upon CFR, we will restate some theory and formalism from that work.

Define R_i^T, player i's **average overall regret** over T

steps, as $R_i^T = \frac{1}{T} \max_{\sigma_i^* \in \Sigma_i} \sum_{t=1}^{T} (u_i(\sigma_i^*, \sigma_{-i}^t) - u_i(\sigma^t))$. In other words, average overall regret is how much more utility a player could have attained on average had they played some other static strategy instead of the sequence of strategies they actually played.

Theorem 1 *(Folk theorem: Zinkevich et al. 2008, Theorem 2) In a two-player zero-sum game at time T, if $R_i^T < \epsilon_i$ for both players, then $\bar{\sigma}^T$ is an $(\epsilon_1 + \epsilon_2)$-Nash equilibrium.*

Theorem 2 *(Zinkevich et al. 2008, Theorem 4) If player i is updating their strategy with CFR, then $R_i^T \leq \Delta|\mathcal{I}_i|\sqrt{|A_i|}/\sqrt{T}$ where $|A_i| = \max_{I \in \mathcal{I}} |A(I)|$*

Since Theorem 2 bounds R_i^T, it follows from Theorem 1 that both players playing according to CFR will yield an average strategy $\bar{\sigma}^T$ that is an $(\epsilon_1 + \epsilon_2)$-Nash equilibrium where $\epsilon_i = \Delta|\mathcal{I}_i|\sqrt{|A_i|}/\sqrt{T}$.

CFR-BR

In Waugh and colleagues' work on abstraction pathologies, they found one case in which abstraction pathologies do not occur (Waugh et al. 2009a, Theorem 3). When solving a game where one agent uses abstraction and the other does not, Waugh *et al.* noted that a strict refinement to the abstraction will result in a monotonic decrease in the abstracted player's exploitability. In addition, we note that the abstracted player's strategy in this equilibrium is by definition the least exploitable strategy that can be represented in the space; otherwise, it would not be an equilibrium. Thus, applying an iterative algorithm such as CFR to this asymmetrically abstracted game will avoid both the pathologies and the overfitting problem, as convergence towards the equilibrium directly minimizes exploitability. However, Waugh *et al.* (2009a, Page 4) note that "...solving a game where even one player operates in the null abstraction is typically infeasible. This is certainly true in the large poker games that have been examined recently in the literature."

We will now present an algorithm that achieves exactly this goal – solving a game where the opponent is unabstracted – and we will demonstrate the technique in the large domain of two-player limit Texas hold'em poker, just such a poker game which has been examined recently in the literature. Our technique, called **CFR-BR**, does this without having to explicitly store the unabstracted opponent's entire strategy, and thus avoids the large memory requirement for doing so. Our explanation of CFR-BR involves two steps, and is illustrated in Figure 2. For our explication, we will assume without loss of generality that the abstracted player is player 1, while the unabstracted player is player 2.

Training against a Best Response. We begin by presenting an alternative method for creating the unabstracted opponent's strategy. The proof of CFR's convergence relies on the folk theorem presented as Theorem 1. Using CFR to update a player's strategy is just one way to create a regret minimizing agent needed to apply the theorem. A best response is also a regret minimizing agent, as it will achieve at most zero regret on every iteration by always choosing the highest valued actions. We will call an agent with this

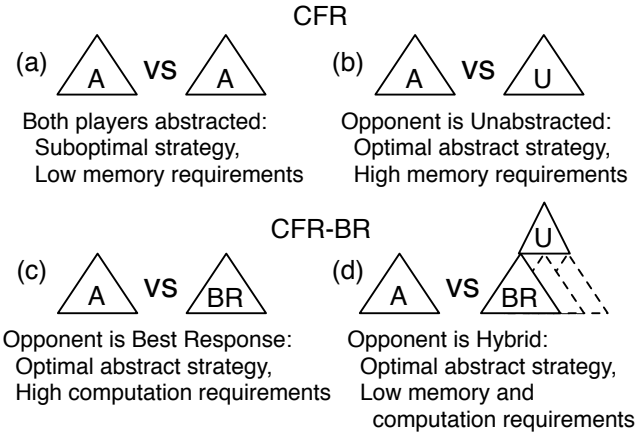


Figure 2: Moving from CFR to CFR-BR

strategy update rule a **BR-agent**, and its strategy on any iteration will be a best response to its opponent's strategy on that same iteration.[1]

In the CFR-BR algorithm, we will start with an agent that updates its strategy using CFR (a **CFR-agent**) and use a BR-agent as its opponent. The CFR-agent may use abstraction. Over a series of iterations, we will update these strategies with respect to each other. Since both of these agents are regret minimizing agents, we can prove that they converge to an equilibrium at a rate similar to the original symmetric CFR approach.

Theorem 3 *After T iterations of CFR-BR, $\bar{\sigma}_1^T$ is player 1's part of an ϵ-Nash equilibrium, with $\epsilon = \frac{\Delta|\mathcal{I}_1|\sqrt{|A_1|}}{\sqrt{T}}$.*

Proof. Since player 1 is playing according to CFR, by Zinkevich et al. (2008), $R_1^T \leq \epsilon$. By the folk theorem, to finish the proof it is enough to show that player 2 has no positive regret.

$$T \cdot R_2^T = \max_{\sigma_2} \left(\sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2) - \sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2^t) \right) \quad (1)$$

$$= \max_{\sigma_2} \sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2) - \sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2^t) \quad (2)$$

$$= \max_{\sigma_2} \sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2) - \sum_{t=1}^{T} \max_{\sigma_2'} u_2(\sigma_1^t, \sigma_2') \leq 0 \quad (3)$$

□

Using an unabstracted BR-agent as opposed to an unabstracted CFR-agent for the opponent has two benefits. First, its strategy will be pure, and can thus be represented more compactly than a behavioral strategy that assigns probabilities to actions. Second, we will now prove that when a CFR-agent plays against a BR-agent, the CFR-agent's sequence of strategies converges to a Nash equilibrium. Typ-

[1]Note that we could not employ two BR-agents in self-play, as they would each have to be a best-response to each other, and so a single iteration would itself require solving the game.

ically, it is only the *average* strategy that converges. However, since the current strategy converges with high probability, tracking the average strategy is unnecessary and only half as much memory is required for the CFR-agent. Note that the proof requires the algorithm to be stopped stochastically in order to achieve its high-probability guarantee. In practice, our stopping time is dictated by convenience and availability of computational resources, and so is expected to be sufficiently random.

Theorem 4 *If CFR-BR is stopped at an iteration T^* chosen uniformly at random from $[1, T]$, then for any $p \in (0, 1]$, with probability $(1-p)$, $\sigma_1^{T^*}$ is player 1's part of an $\frac{\epsilon}{p}$-Nash equilibrium with ϵ defined as in Theorem 3.*

Proof. As in Theorem 3, after T iterations, $R_1^T \leq \epsilon$. This gives a bound on the average observed value based on the game value v_1.

$$R_1^T = \frac{1}{T}\max_{\sigma_1}\sum_{t=1}^{T} u_1(\sigma_1, \sigma_2^t) - \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, \sigma_2^t) \leq \epsilon \quad (4)$$

$$\therefore \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, \sigma_2^t) \geq \frac{1}{T}\max_{\sigma_1}\sum_{t=1}^{T} u_1(\sigma_1, \sigma_2^t) - \epsilon \quad (5)$$

$$\geq \max_{\sigma_1} u_1(\sigma_1, \bar{\sigma}_2^T) - \epsilon \quad (6)$$

$$\geq v_1 - \epsilon \quad (7)$$

For all t, σ_2^t is a best response to σ_1^t, so $u_1(\sigma_1^t, \sigma_2^t) \leq v_1$. With the bounds above, this implies $u_1(\sigma_1^t, \sigma_2^t) < v_1 - \frac{\epsilon}{p}$ on no more than $\lfloor p * T \rfloor$ of the T iterations. If T^* is selected uniformly at random from $[1, T]$, there is at least a $(1-p)$ probability that $u_1(\sigma_1^{T^*}, \sigma_2^{T^*}) \geq v_1 - \frac{\epsilon}{p}$. Because $\sigma_2^{T^*}$ is a best response to $\sigma_1^{T^*}$, this means $\sigma_1^{T^*}$ is player 1's part of an $\frac{\epsilon}{p}$-Nash equilibrium. □

CFR-BR with sampling. CFR-BR still has two remaining challenges that make its use in large games intractable. First, while a best response can be stored compactly, it is still far too large to store in human-scale settings. Second, best response strategies are nontrivial to compute. Recently Johanson and colleagues demonstrated an accelerated best response technique in the poker domain that required just 76 CPU-days, and could be run in parallel in one day (Johanson et al. 2011). While previously such a computation was thought intractable, its use with CFR-BR would involve repeatedly doing this computation over a large number of iterations for convergence to a desired threshold.

However, there is an alternative. Monte-Carlo CFR (MCCFR) is a family of sampling variants of CFR in which some of the actions in a game, such as the chance events, can be sampled instead of enumerated (Lanctot et al. 2009). This results in faster but less precise strategy updates for the agents, in which only subgames of the game tree are explored and updated on any one iteration. One such variant, known as Public Chance Sampled CFR (PCS), uses the fast game tree traversal from the accelerated best response technique to produce a CFR variant that efficiently traverses the game tree, updating larger portions on each iteration than were previously possible (Johanson et al. 2012). The new variant samples only public chance events while updating all possible information sets that vary in each agent's private information.

We can use a variant of PCS with CFR-BR to avoid the time and memory problems described above. On each iteration of CFR-BR, we will sample one public chance event early in the game and only update the complete subgame reachable given that outcome. This subgame includes all possible subsequent chance events after the sampled one. This divides the game tree into two parts: the **trunk** from the root to the sampled public chance event, and the **subgames** that descend from it. Unlike strategies based on regret accumulated over many iterations, portions of a best response strategy can be computed in each subgame as required and discarded afterwards. This avoids the memory problem described above, as at any one time, we only need to know the BR-agent's strategy in the trunk and the one sampled subgame for the current iteration. However, the computation problem remains, as creating the BR-agent's trunk strategy would still require us to traverse all of the possible chance events, in order to find the value of actions prior to the sampled public chance event.

To avoid this final computation problem, we replace the BR-agent with yet another regret-minimizing agent which we call a **Hybrid-agent**. This agent will maintain a strategy and regret values for the trunk of the game, and update it using Public Chance Sampled CFR. In the subgames, it will compute and follow a best response strategy to the opponent's current strategy. Together, this means that on any one iteration, we only need to compute and store one subgame of a best response, and thus require far less time and memory than a BR-agent does. We will now prove that the Hybrid-agent is a regret minimizing agent.

Definition 1 *$\tilde{\mathcal{I}}_2 \subset \mathcal{I}_2$ is a* trunk for player 2 if and only if for all $I, I' \in \mathcal{I}_2$ such that there exists $h \sqsubseteq h'$ with $h \in I$ and $h' \in I'$, if $I' \in \tilde{\mathcal{I}}_2$ then $I \in \tilde{\mathcal{I}}_2$. In other words, once player 2 leaves the trunk, she never returns to the trunk.

Theorem 5 *After T iterations of hybrid CFR-BR using a trunk $\tilde{\mathcal{I}}_2$, with probability $(1-p)$, $R_2^T \leq \epsilon = \left(1 + \frac{\sqrt{2}}{\sqrt{p}}\right)\frac{\Delta|\tilde{\mathcal{I}}_2|\sqrt{|A_2|}}{\sqrt{T}}$.*

Proof. Define a partial best-response with respect to the trunk $\tilde{\mathcal{I}}_2$ as follows

$$\sigma_{2:\mathcal{I}_2\setminus\tilde{\mathcal{I}}_2\to \mathrm{BR}(\sigma_1)} = \underset{\sigma_2' \text{ s.t. } \sigma_2'(I)=\sigma_2(I)\forall I\in\tilde{\mathcal{I}}_2}{\operatorname{argmax}} u_2(\sigma_1, \sigma_2') \quad (8)$$

We can bound the regret using this partial-best response.

$$R_2^T = \frac{1}{T}\max_{\sigma_2}\sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2) - \frac{1}{T}\sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2^t) \quad (9)$$

$$\leq \frac{1}{T}\max_{\sigma_2}\sum_{t=1}^{T} u_2\left(\sigma_1^t, \sigma_{2:\mathcal{I}_2\setminus\tilde{\mathcal{I}}_2\to \mathrm{BR}(\sigma_1^t)}\right) - \frac{1}{T}\sum_{t=1}^{T} u_2\left(\sigma_1^t, \sigma^t_{2:\mathcal{I}_2\setminus\tilde{\mathcal{I}}_2\to \mathrm{BR}(\sigma_1^t)}\right) \quad (10)$$

Because σ_2^t no longer has any effect outside $\tilde{\mathcal{I}}_2$, this is equivalent to doing sampled CFR on a modified game where player 2 only acts at information sets in the trunk. This means we can bound the regret by ϵ with probability $(1-p)$ by application of the MCCFR bound from Lanctot et al. (2009, Theorem 5). □

Since the Hybrid-agent is regret minimizing, it is simple to show that a CFR-agent playing against it will converge to an equilibrium using our sampling variant of CFR-BR.

Theorem 6 *For any $p \in (0,1]$, after T iterations of hybrid CFR-BR using a trunk $\tilde{\mathcal{I}}_2$, with probability $(1-p)$, $(\bar{\sigma}_1^T, \bar{\sigma}_2^T)$ is an $(\epsilon_1+\epsilon_2)$-Nash equilibrium profile with $\epsilon_1 = \left(1+\frac{2}{\sqrt{p}}\right)\frac{\Delta|\mathcal{I}_1|\sqrt{|A_1|}}{\sqrt{T}}$ and $\epsilon_2 = \left(1+\frac{2}{\sqrt{p}}\right)\frac{\Delta|\tilde{\mathcal{I}}_2|\sqrt{|A_2|}}{\sqrt{T}}$.*

Proof. Because player 1 is playing according to sampled CFR, we can bound $R_1^T \leq \epsilon_1$ with probability $(1-p/2)$ by application of the MCCFR bound (2009, Theorem 5). Theorem 5 shows that $R_2^T \leq \epsilon_2$ with probability $(1-p/2)$. Using the union-bound, we have that both conditions hold with at least probability $(1-p)$. If both conditions hold, Theorem 1 gives us that $(\bar{\sigma}_1^T, \bar{\sigma}_2^T)$ is an $(\epsilon_1+\epsilon_2)$-Nash equilibrium. □

Unfortunately, since the Hybrid-agent does not use a best response strategy in the trunk, only the CFR-agent's average strategy (and not the current strategy) is guaranteed to converge to a Nash equilibrium. Since the trunk is such a miniscule fraction of the tree, the current strategy might still converge (quickly) in practice. We will specifically investigate this empirically in the next section. In the remainder of the paper, we will use the name CFR-BR to refer to the variant that uses the Hybrid-agent, as this is the variant that can be practically applied to human scale problems.

Empirical Analysis

Our empirical analysis begins by exploring the correctness of our approach in a toy poker game. We then apply our technique to two-player (heads-up) limit Texas hold'em. Finally, we explore how we can use CFR-BR to answer previously unanswered questions about abstraction quality, abstraction size, and the quality of strategies in competition.

Toy Game. We begin our empirical analysis of CFR-BR in the small poker game of 2-round 4-bet hold'em ([2-4] hold'em), recently introduced by Johanson et al. (2012). While we call this a "toy game", this game has 94 million canonical information sets and 2 billion game states. It is similar to the first two rounds of two-player limit Texas hold'em. A normal sized deck is used, each player is given two private cards at the start of the game, and three public cards are revealed at the start of the second round. In each round, the players may fold, call and bet as normal, with a maximum of four bets per round. At the end of the second round, the remaining player with the best five-card poker hand wins. This game is useful for our analysis because it is small enough to be solved by CFR and CFR-BR without requiring any abstraction. In addition, we can also solve this game when one or both players do use abstraction, so that we can evaluate the impact of the overfitting effect described earlier. The following [2-4] experiments were performed on a 12-core 2.66 GHz computer, using a threaded implementation of CFR and CFR-BR.

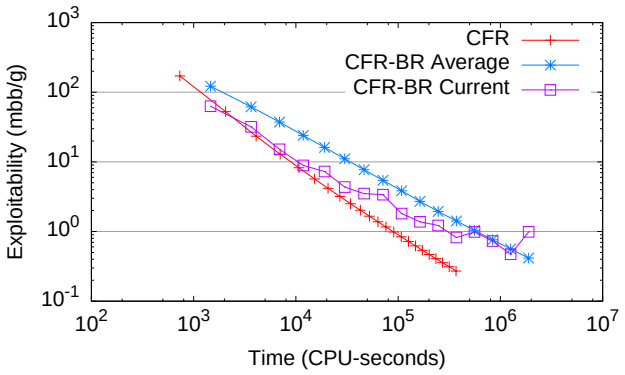


Figure 3: Convergence to equilibrium in unabstracted [2-4] hold'em, 94 million information sets.

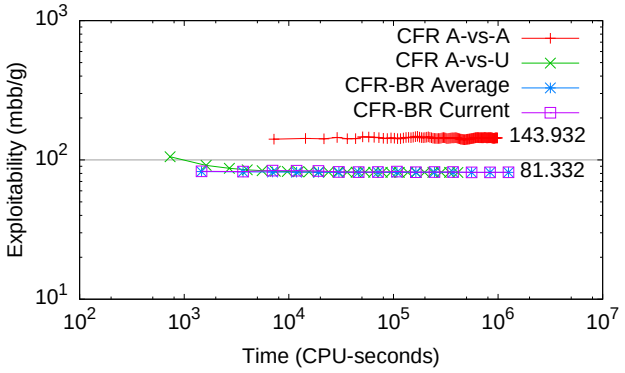


Figure 4: Convergence in [2-4] hold'em using a perfect recall 5-bucket abstraction, 1,790 information sets.

Figure 3 shows the convergence rate of Public Chance Sampled CFR and CFR-BR in unabstracted [2-4] hold'em on a log-log plot. In this two-round game, CFR-BR uses a "1-round trunk", and each iteration involves sampling one set of flop cards. Each series of datapoints represents the set of strategies produced by CFR or CFR-BR as it runs over time, and the y-axis indicates the exploitability of the strategy. In the computer poker community, exploitability is measured in **milli-big-blinds per game (mbb/g)**, where a milli-big-blind is one one-thousandth of a big blind, the ante made by one player at the start of the game. All exploitability numbers for all experiments are computed exactly using the technique in Johanson et al. (2011). From the graph, we see that CFR smoothly converges towards an optimal strategy. The CFR-BR average strategy also smoothly converges towards equilibrium, although at a slower rate than CFR. Finally, the CFR-BR current strategy also improves over time, often faster than the average strategy, although it is noisier.

In Figure 4, we investigate the effects of applying a simple perfect recall abstraction technique to [2-4] hold'em. When CFR solves a game where both players are abstracted (CFR A-vs-A), we see that the strategies are exploitable for 144

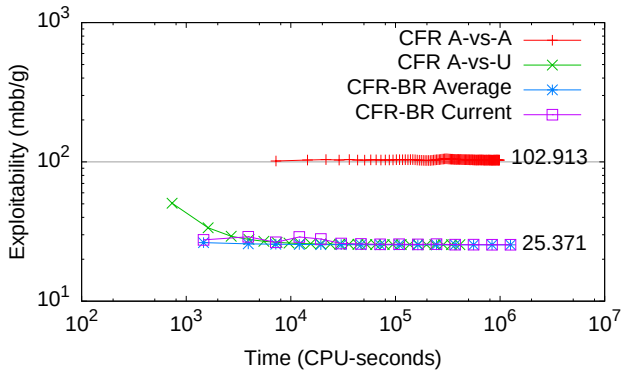


Figure 5: Convergence in [2-4] hold'em using an imperfect recall 570-bucket abstraction, 41k information sets.

mbb/g in the unabstracted game. When CFR is used to create an abstracted player through games against an unabstracted opponent (CFR A-vs-U), the abstracted strategies converge to an exploitability of 81 mbb/g. This demonstrates that the abstraction is capable of representing better approximations than are found by CFR as it is typically used. With CFR-BR, both the average strategy and the current strategy converge to this same improved value.

In Figure 5, we perform a similar experiment where an imperfect recall abstraction is applied to [2-4] hold'em. Imperfect recall abstractions have theoretical problems (*e.g.*, the possible non-existence of Nash equilibria), but have been shown empirically to result in strong strategies when used with CFR (Waugh et al. 2009b; Johanson et al. 2011). When both players are abstracted, CFR converges to an exploitability of 103 mbb/g. When only one player is abstracted, or when CFR-BR is used, the abstracted player's strategy converges to an exploitability of 25 mbb/g.

These results in [2-4] hold'em show that CFR-BR converges to the same quality of solution as using CFR with one unabstracted player, while avoiding the high memory cost of representing the unabstracted player's entire strategy. We also note that while the CFR-BR current strategy is not guaranteed to converge since the unabstracted Hybrid-agent uses CFR in the trunk, in practice the current strategy converges nearly as well as the average strategy. Having demonstrated these properties in a small game, we can now move to the large game of Texas hold'em in which it is intractable to use CFR with an unabstracted opponent.

Texas Hold'em. We can now apply the CFR-BR technique to the large game of two-player limit Texas hold'em, one of the events in the Annual Computer Poker Competition (Zinkevich and Littman 2006). First, we will investigate how the choice of the size of the trunk impacts the memory requirements and convergence rate. In the [2-4] hold'em results presented above, we used a "1-round trunk", where each iteration sampled the public cards revealed at the start of the second round. While the split between the trunk and the subgames could happen at any depth in the tree, in practice it is convenient to start subgames at the start of a round. In a four-round game such as Texas hold'em, there are three

	Texas hold'em RAM required		
CFR-BR Trunk	Trunk	Subgame	Total (48 cores)
1-Round	14.52 KB	1.18 GB	56.64 GB
2-Round	936.33 MB	2.74 MB	1.07 GB
3-Round	359.54 GB	6.54 KB	359.54 GB
CFR (4-round)	140.26 TB	n/a	140.26 TB

Table 1: Memory requirements for the CFR-BR Hybrid-agent in heads-up limit Texas hold'em

such convenient choices for the size of the trunk: 1-round, 2-round, or 3-round. With a 1-round trunk, each iteration involves sampling one set of public cards for the flop, and then unrolling all possible turn and river cards to create a best response strategy for this 3-round subgame. We then update the CFR-agent throughout this large subgame, and use the resulting values to perform CFR updates for both players in the trunk. Alternatively, with a 2-round trunk we will sample one set of flop and turn public cards and unroll all possible river cards. The trunk is thus larger and requires more time to update, but each subgame is smaller and updates are faster. Similarly, a 3-round trunk will sample one set of flop, turn and river cards, and each small subgame involves only the betting on the final round. A 4-round trunk would be equivalent to running CFR with an unabstracted opponent, as the entire game would be in the trunk.

Our choice of the size of the trunk thus allows us to trade off between the time required for the trunk and subgame updates, and the memory required to store an unabstracted CFR trunk strategy and the unabstracted best response subgame strategy. In practice, multiple threads can be used that each perform updates on different subgames simultaneously. Thus, the program as a whole requires enough memory to store one copy of the CFR player's strategy and one copy of the Hybrid-agent's trunk strategy, and each thread requires enough memory to store one pure best response subgame strategy. In Table 1, we present the memory required for a CFR-BR Hybrid-agent using these trunk sizes, after merging isomorphic information sets that differ only by a rotation of the cards' suits. As a 3-round trunk would require 360 gigabytes of RAM just for the Hybrid-agent, our Texas hold'em experiments will only use 1-round and 2-round trunks. Since CFR with an unabstracted opponent requires an infeasible 140 terabytes of RAM, our results will only compare CFR-BR to CFR with both players abstracted. For our experiments on Texas hold'em, a 48-core 2.2 GHz computer was used with a threaded implementation of Public Chance Sampled CFR and CFR-BR.

Figure 6 shows a log-log convergence graph of CFR compared to 1-round and 2-round CFR-BR's current and average strategies in a 10-bucket perfect recall abstraction. This abstraction was used to demonstrate the overfitting effect in the recent work on accelerated best response computation (Johanson et al. 2011, Figure 6), and was the abstraction used by Hyperborean in the 2007 Annual Computer Poker Competition's heads-up limit instant runoff event. Due to the overfitting effect, CFR reaches an observed low point of 277 mbb/g after 2,713 seconds (130k seconds of CPU-time), but then gradually increases to an exploitability of 305 mbb/g.

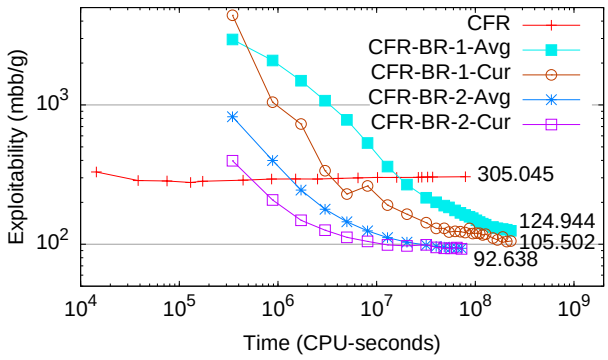


Figure 6: Convergence in Texas hold'em using a perfect recall 10-bucket abstraction, 57 million information sets.

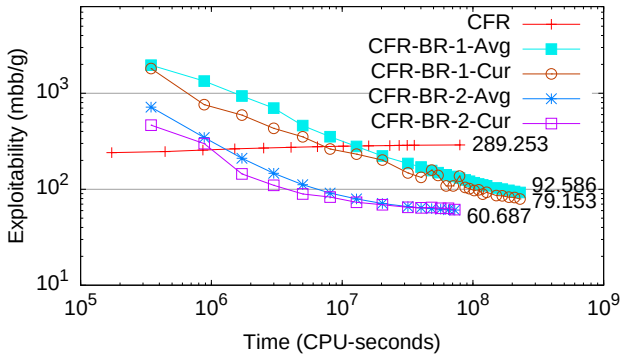
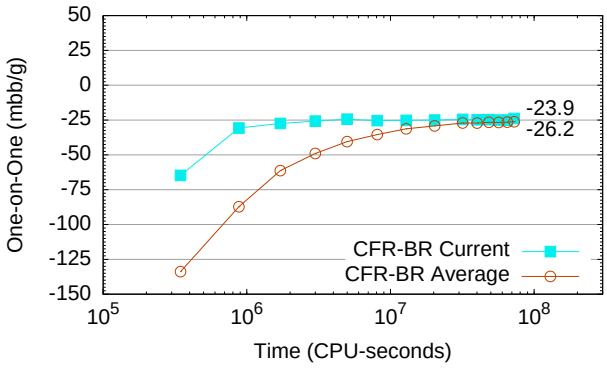


Figure 7: Convergence in Texas hold'em using an imperfect recall 9000-bucket abstraction, 57 million information sets.

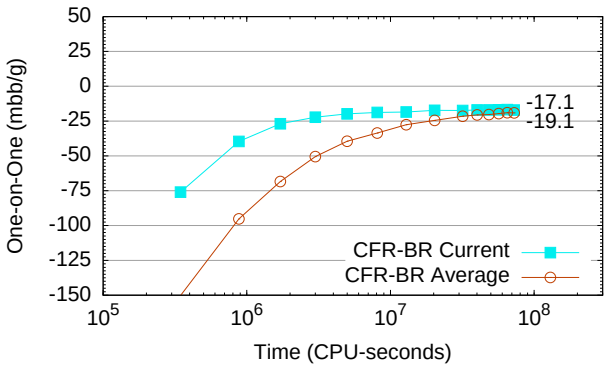
The 2-round trunk CFR-BR current and average strategies reach 92.638 mbb/g and 93.539 mbb/g respectively, and very little progress is being made through further computation.

Figure 7 demonstrates CFR and CFR-BR in a 9000-bucket imperfect recall abstraction. This abstract game is almost exactly the same size as the perfect recall abstraction presented in Figure 6, and was also used previously to demonstrate the overfitting effect (Johanson et al. 2011, Figure 6). In this setting, CFR reaches an observed low of 241 mbb/g within the first 3600 seconds (172k seconds of CPU-time), and then gradually increases to 289 mbb/g. The 2-round trunk CFR-BR current and average strategies reach 61.339 mbb/g and 60.687 mbb/g respectively, after which point the curves appear to have very nearly converged.

These two figures demonstrate that CFR-BR can find dramatically less exploitable strategies than is possible with CFR. The previous least exploitable known strategy for this game was Hyperborean2011.IRO, which was exploitable for 104.410 mbb/g while using an abstraction with 5.8 billion information sets, one hundred times larger than the abstractions used in Figures 6 and 7. While the 1-round and 2-round trunk strategies will converge to the same level of exploitability, we find that the 2-round trunk strategy converges significantly faster while, as shown in Table 1, using far less memory.



(a) 10-bucket perfect recall abstraction.



(b) 9000-bucket imperfect recall abstraction.

Figure 8: One-on-One performance in Texas hold'em between CFR-BR strategies and the final CFR strategy with the same abstraction. Results are accurate to ± 1.2 mbb/g.

In Competition. The significant drop in exploitability provided by CFR-BR is accompanied by a cost to the performance of the strategies against suboptimal opponents, such as those likely to be faced in the Annual Computer Poker Competition. When CFR is applied to an abstract game, it finds a Nash equilibrium within the abstraction and these strategies will do no worse than tie against any other strategy in the abstraction, including those generated by CFR-BR. In fact, since the CFR-BR strategies minimize their loss against an unabstracted opponent, the CFR-BR strategies will likely deviate from the abstract equilibrium in ways that incur losses against an equilibrium found via CFR. Figures 8a and 8b present the in-game performance of the 2-round trunk current and average strategies from Figures 6 and 7 against the final CFR strategy from those abstractions. While the CFR-BR strategies are far less exploitable, they lose to the CFR strategies that share their abstraction.

To further investigate this effect, we can also compare the performance of CFR and CFR-BR average strategies against a CFR strategy from a much larger abstraction. In Figure 9, we use these same CFR and CFR-BR strategies to play games against Hyperborean2011.IRO, which uses an abstraction 100 times larger. Even though this opponent uses a much finer grained abstraction, the CFR strategies

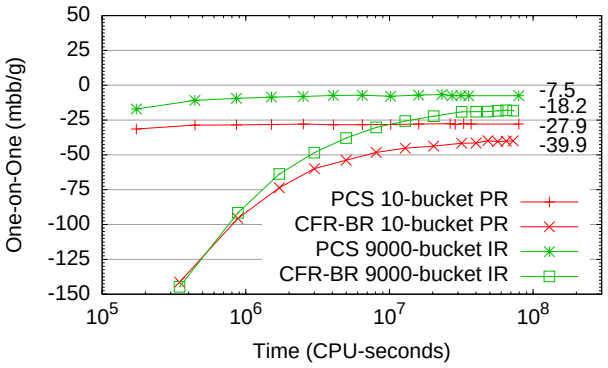


Figure 9: One-on-One performance in Texas hold'em between CFR-BR strategies in varying abstractions and the final CFR strategy using the Hyperborean2011.IRO abstraction. Results are accurate to ± 1.2 mbb/g.

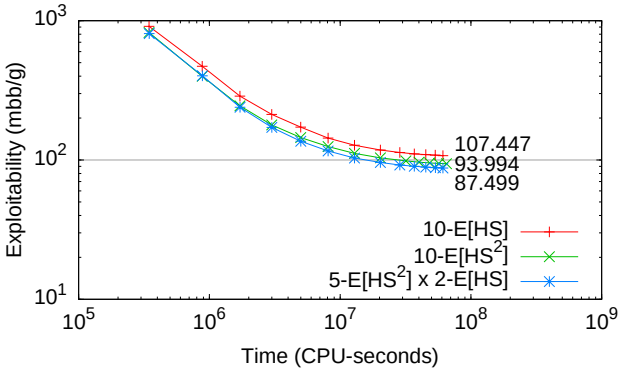


Figure 10: Convergence in Texas hold'em in three perfect recall 10-bucket abstractions, 57 million information sets.

still lose less to this opponent than the CFR-BR strategies. These results underscore an observation made in the analysis of the 2010 Annual Computer Poker Competition competitors: while minimizing exploitability is a well defined goal, lower exploitability is not sufficient on its own to ensure a victory in competition against other suboptimal opponents.

Comparing Abstractions. CFR-BR allows us to find optimal strategies within an abstraction. We can use this tool, then, to evaluate abstractions themselves. In the past, abstractions were typically compared by using CFR to produce strategies, and the one-on-one performance of these strategies was used to select the "strongest" abstraction. When real game best response calculations became feasible, the exploitability of the CFR strategies could instead be used to compare abstractions (Johanson et al. 2011). However, Waugh *et al.* have shown that different abstract game equilibria can have a wide range of exploitability (Waugh et al. 2009a, Table 3), making this approach unreliable. Since CFR-BR finds a least exploitable strategy within an abstraction, it can replace CFR in this task by directly measuring the ability of an abstraction to represent a good approximation to a Nash equilibrium.

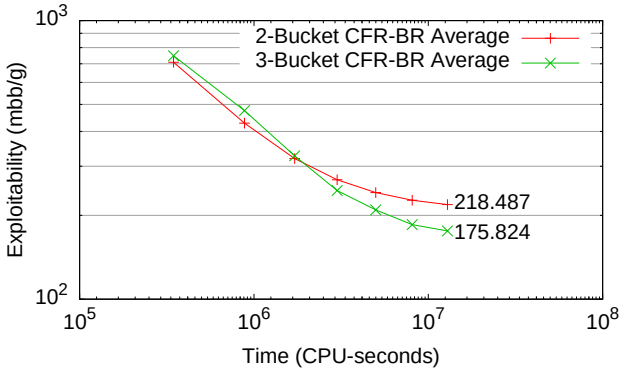


Figure 11: Convergence in Texas hold'em in perfect recall 2-bucket and 3-bucket abstractions, 96056 and 476934 information sets.

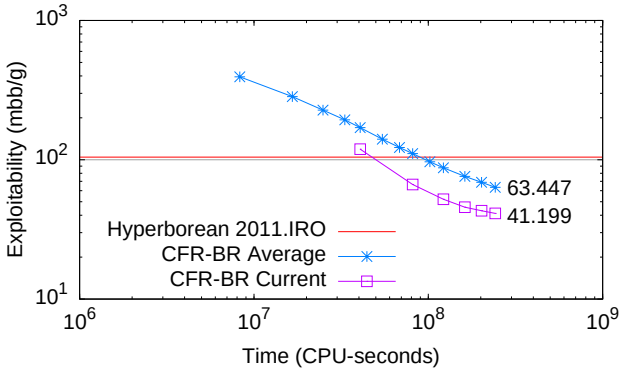


Figure 12: Convergence in Texas hold'em in the Hyperborean2011.IRO abstraction, 5.8 billion information sets.

Figure 10 demonstrates this abstraction comparison by applying CFR-BR to three different 10-bucket perfect recall abstractions. Each abstraction divides the set of hands into equal weight buckets according to different domain features: expected hand strength, expected hand strength squared, or a combination of both, as described in (Johanson 2007, Page 24). While these abstractions are exactly the same size, we found a range of 20 mbb/g – nearly 20% – by changing the features used to create the abstraction.

Abstraction Size. While abstractions can vary in the features used, they also naturally vary in size. In the 2011 Annual Computer Poker Competition entries had a hard disk limit of 30 GB, and some of the entries use large abstractions that fill this space. However, we first focus on the opposite extreme, abstractions whose strategies are so small they can fit on a single 1.44 MB floppy disk. Figure 11 shows the exploitability of CFR-BR strategies in extremely small 2-bucket and 3-bucket perfect recall abstractions. Despite their very coarse abstractions, the resulting strategies are exploitable for just 218.487 mbb/g and 175.824 mbb/g respectively, and are less exploitable than most of the 2010 Annual Computer Poker Competition strategies evaluated by Johanson et al. (2011).

In Figure 12 we apply CFR-BR to the large, fine-grained abstraction used by Hyperborean2011.IRO in the 2011 Annual Computer Poker Competition. This abstraction has 5.8 billion information sets and uses no abstraction beyond merging isomorphic states in the first two rounds. The turn and river rounds have 1.5 million and 840 thousand imperfect recall buckets respectively. The resulting strategy is 20GB using only a single byte per probability. The Hyperborean2011.IRO strategy was created with CFR and was exploitable for 104.410 mbb/g, and prior to this work was the least exploitable strategy known for the game. However, by applying CFR-BR to this abstraction, the current strategy at the final datapoint is exploitable for just 41.199 mbb/g and is the new least exploitable strategy known for heads-up limit Texas hold'em poker.

Conclusion

Although there are efficient game solving algorithms for two-player, zero-sum games, many games are far too large to be tractably solved. State space abstraction techniques can be used in such cases to produce an abstract game small enough to be tractably solved; however, recent work has demonstrated that an equilibrium in an abstract game can often be far more exploitable in the unabstracted game compared to the least exploitable strategies that can be represented in the abstraction. In this work we presented CFR-BR, a new game solving algorithm that converges to one of these least exploitable abstract strategies, while avoiding the high memory cost that made such a solution previously intractable. We demonstrated the effectiveness of our approach in the domain of two-player limit Texas hold'em, where it was used to generate far closer approximations to the unknown, optimal Nash equilibrium strategy within an abstraction than was possible using previous state-of-the-art techniques.

Acknowledgements

The authors would like to thank Marc Lanctot and the members of the Computer Poker Research Group at the University of Alberta for helpful conversations pertaining to this research. This research was supported by NSERC, Alberta Innovates Technology Futures, and the use of computing resources provided by WestGrid, Réseau Québécois de Calcul de Haute Performance, and Compute/Calcul Canada.

References

Billings, D.; Burch, N.; Davidson, A.; Holte, R.; Schaeffer, J.; Schauenberg, T.; and Szafron, D. 2003. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*.

Gatti, N. 2008. Extending the alternating-offers protocol in the presence of competition: Models and theoretical analysis. *Annals of Mathematics in Artificial Intelligence* 55(3-4):189–236.

Gilpin, A.; Sandholm, T.; and Sørensen, T. B. 2007. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI)*. AAAI Press.

Hoda, S.; Gilpin, A.; Peña, J.; and Sandholm, T. 2010. Smoothing techniques for computing nash equilibria of sequential games. *Mathematics of Operations Research* 35(2):494–512.

Johanson, M.; Waugh, K.; Bowling, M.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 258–265. AAAI Press.

Johanson, M.; Bard, N.; Lanctot, M.; Gibson, R.; and Bowling, M. 2012. Efficient nash equilibrium approximation through monte carlo counterfactual regret minimization. In *Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems. To appear.

Johanson, M. 2007. Robust strategies and counter-strategies: Building a champion level computer poker player. Master's thesis, University of Alberta.

Lanctot, M.; Waugh, K.; Zinkevich, M.; and Bowling, M. 2009. Monte Carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems 22 (NIPS)*.

Lazaric, A.; de Cote, J. E. M.; and Gatti, N. 2007. Reinforcement learning in extensive form games with incomplete information: the bargaining case study. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*.

Mostafa, H.; Lesser, V.; and Miklau, G. 2008. Self-interested database managers playing the view maintenance game. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Osborne, M., and Rubinstein, A. 1994. *A Course in Game Theory*. The MIT Press.

Procaccia, A. D., and Rosenschein, J. S. 2005. Extensive-form argumentation games. In *The Third European Workshop on Multi-Agent Systems (EUMAS)*.

Risk, N. A., and Szafron, D. 2010. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010)*. International Foundation for Autonomous Agents and Multiagent Systems.

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

Waugh, K.; Schnizlein, D.; Bowling, M.; and Szafron, D. 2009a. Abstraction pathology in extensive games. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Waugh, K.; Zinkevich, M.; Johanson, M.; Kan, M.; Schnizlein, D.; and Bowling, M. 2009b. A practical use of imperfect recall. In *Proceedings of the Eighth Symposium on Abstraction, Reformulation and Approximation (SARA)*.

Zinkevich, M., and Littman, M. 2006. The AAAI computer poker competition. *Journal of the International Computer Games Association* 29. News item.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2008. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20 (NIPS)*.