An Algorithm for Constructing and Solving Imperfect Recall Abstractions of Large Extensive-Form Games

Jiří Čermák, Branislav Bošanský, Viliam Lisý
Department of Computer Science, Czech Technical University in Prague
{cermak, bosansky, lisy}@agents.fel.cvut.cz

Abstract

We solve large two-player zero-sum extensive-form games with perfect recall. We propose a new algorithm based on fictitious play that significantly reduces memory requirements for storing average strategies. The key feature is exploiting imperfect recall abstractions while preserving the convergence rate and guarantees of fictitious play applied directly to the perfect recall game. The algorithm creates a coarse imperfect recall abstraction of the perfect recall game and automatically refines its information set structure only where the imperfect recall might cause problems. Experimental evaluation shows that our novel algorithm is able to solve a simplified poker game with $7 \cdot 10^5$ information sets using an abstracted game with only 1.8% of information sets of the original game. Additional experiments on poker and randomly generated games suggest that the relative size of the abstraction decreases as the size of the solved games increases.

1 Introduction

Dynamic games with a finite number of moves can be modeled as extensive-form games (EFGs) that are general enough to represent scenarios with stochastic events and imperfect information. EFGs can model recreational games, such as poker, as well as real-world situations in physical security [Lisý *et al.*, 2016], auctions, or medicine [Chen and Bowling, 2012]. EFGs are represented as game trees where nodes correspond to states of the game and edges to actions of players. Imperfect information is modeled by grouping indistinguishable states into information sets.

There are two approaches to solving EFGs. First, the online (or game-playing) algorithms which given the observations of the game state compute the action to be played. Second, the offline algorithms which compute (approximate) the strategy in the whole game and play according to this strategy. The latter algorithms typically provide a better approximation of equilibrium strategies in large games compared to online algorithms [Bošanský *et al.*, 2016]. One exception is the recently introduced continual resolving algorithm used in DeepStack [Moravčík *et al.*, 2017], which provides less exploitable strategies than existing offline algorithms in heads-up no-limit Texas Hold'em, an imperfect information game with 10^{160} decision points. The main caveat is that DeepStack exploits the specific structure of poker where all actions are observable, and the generalization to other games is not straightforward. We thus focus on offline algorithms.

Most of the existing offline algorithms [von Stengel, 1996; Zinkevich *et al.*, 2008] require players to remember all the information gained during the game – a property denoted as a *perfect recall*. The main disadvantage of perfect recall is that it causes the size of strategies (a randomized selection of an action in each information set) to grow exponentially with the number of moves. Therefore, a popular approach is to use *abstractions* [Gilpin *et al.*, 2007] – create an abstracted game by merging information sets to reduce the size of the strategy representation, solve the abstracted game, and translate the strategies back to the original game. The majority of existing algorithms create perfect recall abstractions, where the requirement of perfect memory severely limits possible reductions in the size of strategies of the abstracted game, as it still grows exponentially with increasing number of moves in the abstracted game (e.g., see [Gilpin and Sandholm, 2007; Kroer and Sandholm, 2014; Brown and Sandholm, 2015]). Additionally, finding optimal perfect recall abstractions is computationally hard [Kroer and Sandholm, 2014].

A limited amount of work relaxes the perfect recall restriction in abstractions. Very specific imperfect recall abstractions that allow using perfect recall solution techniques are *(skew) well-formed games* [Lanctot *et al.*, 2012; Kroer and Sandholm, 2016] and *normal-form games with sequential strategies* [Bošanský *et al.*, 2015; Lisý *et al.*, 2016]. Skew well-formed games only merge information sets, which satisfy strict restrictions on the structure of the game tree above and below them, such that for all possible strategies of the opponent, a strategy which is optimal in one of the information sets must have bounded error in the other merged sets. Players cannot observe actions of the opponent at all in normal-form games with sequential strategies. These restrictions prevent us from creating sufficiently small and useful abstracted games and thus fully exploit the possibilities of imperfect recall. Existing methods for using imperfect recall abstractions without severe limitations cannot provide any guarantees of the quality of computed strategies [Waugh *et al.*, 2009], or assume that the abstraction is given and require computationally complex algorithms to solve it [Čermák *et al.*, 2017].

We address these issues in this paper and demonstrate the possible space savings achievable by automated imperfect recall abstractions. We create a novel domain-independent algorithm FPIRA, which starts with creating a coarse imperfect recall abstraction of the given perfect recall game. FPIRA then uses the fictitious play framework to simultaneously solve and refine the imperfect recall abstraction, to guarantee the convergence to Nash equilibrium of the original perfect recall game. FPIRA is conceptually similar to Double Oracle algorithm (DO) [Bošanský *et al.*, 2014] since it creates a smaller version of the original game and repeatedly refines it until the optimal solution of the original game is found. FPIRA, however, uses a game with imperfect recall during the computation, while DO uses a smaller perfect recall game. Hence, FPIRA exploits completely different type of sparseness than DO. The experimental evaluation shows that FPIRA can solve modification of Leduc hold'em with $7 \cdot 10^5$ information sets using an abstracted game with 1.8% of information sets of the original game. Furthermore, experiments on other simplified pokers and random games suggest that the relative size of the needed abstractions significantly decreases as the size of the solved games increases.

2 Extensive-Form Games

We first describe necessary technical introduction to extensive-form games (EFGs). A two player EFG G is a tuple $\{\mathcal{P}, \mathcal{H}, \mathcal{Z}, P, u, \mathcal{I}, A\}$. $\mathcal{P} = \{1, 2\}$ denotes the set of players. We use i to refer to a player and $-i$ to refer to the opponent of i. Set $\mathcal{H}$ contains all the states of the game. $P : \mathcal{H} \to \mathcal{P} \cup \{c\}$ is the function associating a player from $\mathcal{P}$ or nature c with every $h \in \mathcal{H}$. Nature c represents the stochastic environment of the game. $\mathcal{Z} \subseteq \mathcal{H}$ is a set of terminal states. $u_i(z)$ is a utility function assigning to each leaf the value of preference for player i; $u_i : \mathcal{Z} \to \mathbb{R}$. For zero-sum games it holds that $u_i(z) = -u_{-i}(z), \forall z \in \mathcal{Z}$. The imperfect information is defined using the information sets. $\mathcal{I}_i$ is a partitioning of all $\{h \in \mathcal{H} : P(h) = i\}$ into these information sets. All states h contained in one information set $I_i \in \mathcal{I}_i$ are indistinguishable to player i. The set of available actions $A(h)$ is the same $\forall h \in I_i$. We overload the notation and use $A(I_i)$ as actions available in I_i. A *sequence* σ_i is a list of actions of player i ordered by their occurrence on the path from the root of the game tree to some node. By $seq_i(h)$ we denote the sequence of player i leading to the state h. We overload the notation and use $seq_i(I)$ as a set of sequences of player i leading to the information set I. A game has *perfect recall* iff $\forall i \in \mathcal{P}\ \forall I_i \in \mathcal{I}_i$, for all the states $h, h' \in I_i$ holds that $seq_i(h) = seq_i(h')$. If there exists at least one information set where this does not hold (denoted as imperfect recall information set) the game has *imperfect recall*.

Definition 1. *By the* coarsest perfect recall refinement *of an imperfect recall game G we define a perfect recall game G' where we split the imperfect recall information sets to largest subsets satisfying the perfect recall assumption.*

2.1 Strategies in Imperfect Recall Games

There are several representations of strategies in EFGs. A *pure strategy* s_i for player i is a mapping assigning $\forall I_i \in \mathcal{I}_i$ a member of $A(I_i)$. $\mathcal{S}_i$ is a set of all pure strategies for player i. A *mixed strategy* m_i is a probability distribution over $\mathcal{S}_i$, set of all mixed strategies of i is denoted as $\mathcal{M}_i$. *Behavioral strategy* b_i assigns a probability distribution over $A(I_i)$ for each I_i. $\mathcal{B}_i$ is a set of all behavioral strategies for i, $\mathcal{B}_i^p \subseteq \mathcal{B}_i$ denotes the set of deterministic behavioral strategies for i. A *strategy profile* is a set of strategies, one strategy for each player. We overload the notation and use u_i as the expected utility of i when the players play according to pure (mixed, behavioral) strategies.

Behavioral strategies and mixed strategies have the same expressive power in perfect recall games, but their expressive power can differ in imperfect recall games [Kuhn, 1953]. Moreover, the size of these representations differs significantly. Mixed strategies of player i use probability distribution over $\mathcal{S}_i$, where $|\mathcal{S}_i| \in \mathcal{O}(e^{|\mathcal{Z}|})$. Behavioral strategies create probability distribution over the set of actions (its size is proportional to the number of information sets, which can be exponentially smaller than $|\mathcal{Z}|$). Hence we need to use behavioral strategies if we want to exploit the space savings caused by the reduced number of information sets due to some information abstraction.

A *best response* of player i against b_{-i} is a strategy $b_i^{BR} \in BR(b_{-i})$, where $u_i(b_i^{BR}, b_{-i}) \geq u_i(b'_i, b_{-i})$ for all $b'_i \in \mathcal{B}_i$ ($BR(b_{-i})$ denotes a set of all best responses to b_{-i}).

Definition 2. *We say that b_i and b'_i are* realization equivalent *if for any b_{-i}, $\forall z \in \mathcal{Z}\ \pi^b(z) = \pi^{b'}(z)$, where $b = (b_i, b_{-i})$ and $b' = (b'_i, b_{-i})$ and $\pi^b(z)$ stands for the probability that z is reached when playing accordint to b.*

The concept of realization equivalence can be applied also to different strategy representations. Finally, we define the Nash equilibrium in behavioral strategies.

Definition 3. *We say that strategy profile $b = \{b_i, b_{-i}\}$ is a Nash equilibrium (NE) in behavioral strategies iff $\forall i \in \mathcal{P}\ \forall b_i^p \in \mathcal{B}_i^p : u_i(b_i, b_{-i}) \geq u_i(b_i^p, b_{-i})$.*

3 Fictitious Play

Fictitious play (FP) is an iterative algorithm defined on normal-form games [Brown, 1949]. It keeps track of average strategies of both players $\bar{m}_i^T, \bar{m}_{-i}^T$. Players take turn updating their average strategy. In iteration T, player i computes $s_i^T \in BR(\bar{m}_{-i}^{T-1})$. He then updates his average strategy $\bar{m}_i^T = \frac{T_i - 1}{T_i} \bar{m}_i^{T-1} + \frac{1}{T_i} s_i^T$ (T_i is the number of updates performed by i plus 1). In two-player zero-sum games $\bar{m}_i^T, \bar{m}_{-i}^T$ converge to a NE [Robinson, 1951]. There is a long-standing conjecture [Karlin, 2003; Daskalakis and Pan, 2014] that the convergence rate of FP is $O(T^{-\frac{1}{2}})$, the same order as the convergence rate of Counterfactual Regret Minimization (CFR) [Zinkevich *et al.*, 2008] (though the empirical convergence of CFR tends to be better).

When applying FP to behavioral strategies in perfect recall zero-sum EFG G', one must update the average behavioral strategy $\bar{b}_i^t$ such that it is realization equivalent to $\bar{m}_i^t$ obtained when solving the normal form game corresponding to G' for all t and all $i \in \mathcal{P}$ to keep the convergence guarantees. To update the behavioral strategy in such a way we use the following Lemma [Heinrich *et al.*, 2015].

Lemma 1. *Let b_i, b'_i be two behavioral strategies and m_i, m'_i two mixed strategies realization equivalent to b_i, b'_i, and $\lambda_1, \lambda_2 \in (0, 1)$, $\lambda_1 + \lambda_2 = 1$. Then $\forall I \in \mathcal{I}_i$*

$$b''_i(I) = b_i(I) + \frac{\lambda_2 \pi_i^{b'_i}(I)}{\lambda_1 \pi_i^{b_i}(I) + \lambda_2 \pi_i^{b'_i}(I)}(b'_i(I) - b_i(I)),$$

where $\pi_i^{b_i}(I)$ is the probability that I is visited when playing b_i, defines a behavioral strategy b''_i realization equivalent to the mixed strategy $m''_i = \lambda_1 m_i + \lambda_2 m'_i$.

4 The FPIRA Algorithm

Let us now describe the main algorithm (denoted as FPIRA, Fictitious Play for Imperfect Recall Abstractions) presented in this paper and prove its convergence in two-player zero-sum EFGs. We give a high-level idea behind FPIRA, and we provide a pseudocode with the description of all steps.

Given a perfect recall game G', FPIRA creates a coarse imperfect recall abstraction G of G'. The algorithm then follows the FP procedure. It keeps track of average strategies of both players in the information set structure of G and updates the strategies in every iteration based on the best responses to the average strategies computed directly in G'. To ensure the convergence to Nash equilibrium of G', FPIRA refines the information set structure of G when needed to make sure that the strategy update does not lead to more exploitable average strategies in the following iterations compared to the strategy update made directly in G'.

Algorithm 1: FPIRA algorithm

input : G', T
output : $\bar{b}_i^T, \bar{b}_{-i}^T, G^T$

1 $G^1 \leftarrow$ `BuildAbstraction`(G')
2 $\bar{b}_1^0 \leftarrow$ `PureStrat`$(G^1), \bar{b}_2^0 \leftarrow$ `PureStrat`(G^1)
3 **for** $t \in \{1, ..., T\}$ **do**
4 $\quad i \leftarrow$ `ActingPlayer`(t)
5 $\quad b_i^t \leftarrow BR(G', \bar{b}_{-i}^{t-1})$
6 $\quad G^t \leftarrow$ `RefineForBR`(G^t, b_i^t)
7 $\quad \hat{b}_i^t \leftarrow$ `UpdateStrategy`$(G^t, \bar{b}_i^{t-1}, b_i^t)$
8 $\quad \widetilde{b}_i^t \leftarrow$ `UpdateStrategy`$(G', \bar{b}_i^{t-1}, b_i^t)$
9 $\quad$ **if** `ComputeDelta`$(G', \hat{b}_i^t, \widetilde{b}_i^t) > 0$ **then**
10 $\quad\quad G^{t+1} \leftarrow$ `Refine`$(G^t), \bar{b}_i^t \leftarrow \widetilde{b}_i^t$
11 $\quad$ **else**
12 $\quad\quad G^{t+1} \leftarrow G^t, \bar{b}_i^t \leftarrow \hat{b}_i^t$

In Algorithm 1 we present the pseudocode of FPIRA. FPIRA is given the original perfect recall game $G' = \{\mathcal{P}, \mathcal{H}, \mathcal{Z}, P, u, \mathcal{I}', A'\}$ and a number of iterations to perform T. FPIRA first creates a coarse imperfect recall abstraction $G^1 = \{\mathcal{P}, \mathcal{H}, \mathcal{Z}, P, u, \mathcal{I}^1, A^1\}$ of G' (line 1) as described in Section 4.1. Next, it initializes the strategies of both players to an arbitrary pure strategy in G^1 (line 2). FPIRA then performs T iterations. In every iteration it updates the average strategy of one of the players and if needed the information set structure of the abstraction (the game used in iteration t is denoted as $G^t = \{\mathcal{P}, \mathcal{H}, \mathcal{Z}, P, u, \mathcal{I}^t, A^t\}$). In every iteration player i first computes the best response b_i^t to $\bar{b}_{-i}^{t-1}$ in G' (line 5). Since b_i^t is computed in G', FPIRA first needs to make sure that the structure of information sets in G^t allows b_i^t to be played. If not, G^t is updated as described in Section 4.2, Case 1 (line 6). Next, FPIRA computes $\hat{b}_i^t$ as the strategy resulting from the update in abstracted G^t (line 7) and $\widetilde{b}_i^t$ as the strategy resulting from the update in original G' (line 8). FPIRA then checks whether the update in G^t changes the expected values of the pure strategies of the opponent compared to the update in G' using $\hat{b}_i^t$ and $\widetilde{b}_i^t$ (line 9, Section 4.2 Case 2). If yes, FPIRA refines the information set structure of G^t, creating G^{t+1} such that no error in expected values of pure strategies of the opponent is created (Section 4.2 Case 2), sets $\bar{b}_i^t = \widetilde{b}_i^t$ (line 10), and continues using G^{t+1}. If there is no need to update the structure of G^t, FPIRA sets $G^{t+1} = G^t, \bar{b}_i^t = \hat{b}_i^t$ and continues with the next iteration.

4.1 Creating the Initial Abstraction

FPIRA creates G^1 (line 1 in Algorithm 1) as a coarse imperfect recall abstraction of G' by merging possible information sets, such that the coarsest perfect recall refinement of G^1 is G'. To achieve this, FPIRA needs to make sure that when merging a set of information sets $\mathcal{J} \subseteq \mathcal{I}_i$ there are no two distinct $I, I' \in \mathcal{J}$ which, when merged, create a perfect recall information set. This is required since, as discussed in the next section, the algorithm splits only imperfect recall information sets. If FPIRA joined information sets not resulting in the imperfect recall information set, it would end up solving a different game.

More formally, we use the following algorithm to build G^1. For all i create $\mathcal{K}_i$ as a set of disjoint subsets of information sets $\mathcal{I}'_i$ of G', such that $\bigcup_{\mathcal{K}' \in \mathcal{K}_i} \mathcal{K}' = \mathcal{I}'_i$, and

$$\forall \mathcal{K}' \in \mathcal{K}_i, \forall I, I' \in \mathcal{K}' \colon |A'(I)| = |A'(I')| \wedge |seq_i(I)| = |seq_i(I')|.$$

In other words, all the information sets in $\mathcal{K}'$ must have the same number of actions available and the same length of the sequence of i leading to them. Every $\mathcal{K}' \in \mathcal{K}_i$ gives us candidates for merging. However, as discussed above, we need to make sure that we never merge any pair of information sets, which would result in a perfect recall information set after the merge. Hence, we further split every $\mathcal{K}' \in \mathcal{K}_i$ to the smallest possible set $\mathcal{J} = \{\mathcal{J}^1, ..., \mathcal{J}^k\}$, such that

$$\forall \mathcal{J}^j \in \mathcal{J}, \forall I, I'' \in \mathcal{J}^j : seq_i(I) \neq seq_i(I').$$

I and I' are information sets of G', hence both $seq_i(I)$ and $seq_i(I')$ are singletons. $\mathcal{J}$ needs not be unique, in our implementation we choose randomly between possible $\mathcal{J}$. Finally, every information set in the information set structure $\mathcal{I}^1$ of G^1 corresponds to $\mathcal{J}^j$ from the union of all $\mathcal{J}$ for all players.

4.2 Updating G^t

There are two reasons for splitting some $I \in \mathcal{I}_i^t$ in iteration t (we assume player i computes the best response in t): (1) the best response computed in G' prescribes more than one action in I or (2) I causes expected values of some pure strategy of $-i$ to be different after the average strategy update of i compared to what would happen when updating in G'.

To formally describe the splitting rules, let us first define mappings $\Phi_t : \mathcal{I}' \to \mathcal{I}^t$, which for $I \in \mathcal{I}'$ returns the information set containing I in G^t and $\Phi_t^{-1} : \mathcal{I}^t \to \wp(\mathcal{I}')$, the inverse of Φ_t. By $\Xi_t : A' \to A^t$ and $\Xi_t^{-1} : A^t \to \wp(A')$ we denote a mapping of actions from G' to G^t and vice versa.

Case 1: FPIRA checks in every iteration t if there exists $I \in \mathcal{I}_i^t$ where the best response b_i^t prescibes more than 1 action. If yes, FPIRA splits I to a set of information sets $\hat{\mathcal{I}}$ and inforation set I'', such that $\forall \hat{I}_a \in \hat{\mathcal{I}}$, $\hat{I}_a$ is a unification of all $I' \in \Phi_t^{-1}(I)$ where $b_i^t(I', a') = 1$ for $\Xi_t(a') = a$ and $I'' = \{h \in I | \forall \hat{I} \in \hat{\mathcal{I}} : h \notin \hat{I}\}$ (line 6 in Algorithm 1).

Case 2: The algotrithm first constructs the average behavioral strategy $\hat{b}_i^t$ in G^t (line 7). This is done according to Lemma 1 from $\bar{b}_i^{t-1}$ with weight $\frac{t_i - 1}{t_i}$ and b_i^t with weight $\frac{1}{t_i}$, where t_i is the number of updates performed by i so far, plus 1 for the initial strategy (b_i^t is used with mappings Φ_t and Ξ_t). Next, FPIRA constructs $\widetilde{b}_i^t$ (line 8) in the same way in the information set structure of G' ($\bar{b}_i^{t-1}$ is used with mappings Φ_t^{-1} and Ξ_t^{-1}). FPIRA then computes

$$\Delta_i^t = \max_{b_{-i} \in \mathcal{B}_{-i}^P} |u_{-i}(\widetilde{b}_i^t, b_{-i}) - u_{-i}(\hat{b}_i^t, b_{-i})|,$$

as described below (line 9). If $\Delta_i^t = 0$, none of the pure strategies of $-i$ changed its expected value compared to the update in G'. In this case, FPIRA sets $G^{t+1} = G^t$, $\bar{b}_i^t = \hat{b}_i^t$ (line 12). If $\Delta_i^t > 0$, the expected value of some pure strategy of $-i$ changed when updating the strategy in G^t, compared to the expected value it would get against the strategy updated in G'. FPIRA then creates G^{t+1} in the following way. Every imperfect recall information set $I \in \mathcal{I}_i^t$ which is visited when playing b_i^t is split to a set of information sets $\hat{\mathcal{I}} \subseteq \Phi_t^{-1}(I)$ and an information set I'', such that $\hat{\mathcal{I}}$ contains all the $I' \in \Phi_t^{-1}(I)$ which can be visited when playing b_i^t, I'' contains the rest of $h \in I$. The average strategy in all $I' \in \hat{\mathcal{I}} \cup \{I''\}$ before the strategy update is set to the strategy previously played in I. More formally, $\forall I' \in \hat{\mathcal{I}} \cup \{I''\}$ the strategy is set to

$$\bar{b}_i^{t-1}(I', a) = \bar{b}_i^{t-1}(\Phi_t(I'), \Xi_t(a)), \forall a \in A'(I').$$

The strategy resulting from update in G' is a valid strategy in G^{t+1} after such update, hence $\bar{b}_i^t = \widetilde{b}_i^t$. Notice that G' is still the coarsest perfect recall refinement of G^{t+1}, additionally by setting $\bar{b}_i^t = \widetilde{b}_i^t$, we made sure that $\Delta_i^t = 0$ since the update is now equal to the update that would occur in G'. This, as we will show in Section 4.3, is sufficient to guarantee the convergence of $\bar{b}_i^t, \bar{b}_{-i}^t$ to Nash equilibrium of G'.

Computing Δ_i^t. Given $\hat{b}_i^t$ and $\widetilde{b}_i^t$, Δ_i^t can be computed as

$$\Delta_i^t = \max_{b_{-i} \in \mathcal{B}_{-i}^P} \sum_{z \in \mathcal{Z}} |\pi_{-i}^{b_{-i}}(z) \left[\pi_i^{\widetilde{b}_i^t}(z) - \pi_i^{\hat{b}_i^t}(z)\right] u_{-i}(z)|.$$

Δ_i^t can be computed in $O(|\mathcal{Z}|)$ as a standard best response tree traversal.

Example 1. *Let us demonstrate several iterations of* FPIRA *algorithm. Consider the perfect recall game from Figure 1 (a) as G' and the imperfect recall game from Figure 1 (b) as G^1. The function Ξ_1 is $\Xi_1(t) = \Xi_1(v) = c, \Xi_1(u) = \Xi_1(w) = d$, identity otherwise. Note that when we apply strategies from G' to G^t and vice versa, we assume that it is done with respect to Ξ_t and Ξ_t^{-1}. Lets assume that* FPIRA *first initializes the strategies to $\bar{b}_1^0(b) = \bar{b}_1^0(d) = 1, \bar{b}_2^0(e) = 1$.*

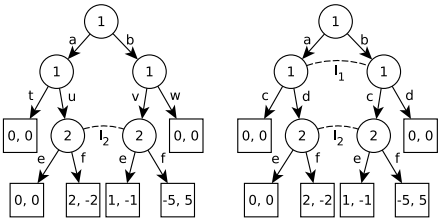


Figure 1: (a) G' for demonstration of FPIRA iterations (b) G^1 for demonstration of FPIRA.

Iteration 1: *The player 1 starts in iteration 1.* FPIRA *computes $b_1^1 \in BR(\bar{b}_2^0)$ in G', resulting in $b_1^1(b) = b_1^1(v) = 1$. Next,* FPIRA *checks whether b_1^1 is playable in G^1. Since there is no information set in G^1 for which b_1^1 assigns more than one action, we do not need to update G^1 in any way. We follow by computing $\hat{b}_1^1$ and $\widetilde{b}_1^1$ according to Lemma 1 with $\lambda_1 = \lambda_2 = 0.5$. In this case $\hat{b}_1^1(b) = \widetilde{b}_1^1(b) = 1, \hat{b}_1^1(c) = \widetilde{b}_1^1(v) = 0.5$. Since $\hat{b}_1^1$ and $\widetilde{b}_1^1$ are equal, w.r.t. Ξ_1, we know that $\Delta_i = 0$. Hence we let $G^2 = G^1$, $\bar{b}_1^1 = \hat{b}_1^1$ and $\Xi_2 = \Xi_1$.*

Iteration 2: *Player 2 continues in iteration 2. Notice that player 2 does not have imperfect recall, hence there is no need to discuss this iteration in such detail.* FPIRA *computes the best response to $\bar{b}_1^1$, resulting in $b_2^2(f) = 1$. The algorithm then computes $\hat{b}_2^2$ and $\widetilde{b}_2^2$, resulting in $\hat{b}_2^2(e) = \widetilde{b}_2^2(e) = 0.5$. Hence, we let $G^3 = G^2$, $\bar{b}_2^2 = \hat{b}_2^2$ and $\Xi_3 = \Xi_2$.*

Iteration 3: *The best response in this iteration is $b_1^3(a) = b_1^3(u) = 1$, which is again playable in G^3, hence we do not need to update G^3 at this point.* FPIRA *computes $\hat{b}_1^3$ resulting in $\hat{b}_1^3(a) = \frac{1}{3}, \hat{b}_1^3(d) = \frac{2}{3}$, $\widetilde{b}_1^3$ is, on the other hand, $\widetilde{b}_1^3(a) = \frac{1}{3}$, $\widetilde{b}_1^3(t) = 1, \widetilde{b}_1^3(w) = 0.5$ (both according to Lemma 1 with $\lambda_1 = \frac{2}{3}, \lambda_2 = \frac{1}{3}$). In this case, $\Delta_1^3 = 1$ since by playing f player 2 gets $\frac{2}{3}$ against $\hat{b}_1^3$ compared to $\frac{5}{3}$ against $\widetilde{b}_1^3$. Hence, the algorithm splits all imperfect recall information sets reachable when playing b_1^3, in this case I_1, as described in Section 4.2, Case 2, resulting in G'. Therefore, $G^4 = G'$, $\bar{b}_1^3 = \widetilde{b}_1^3$ and Ξ_4 is set to identity.*

4.3 Theoretical Properties

We show that the convergence quarantees of FP in two-player zero-sum perfect recall game G' [Heinrich *et al.*, 2015] directly apply to FPIRA solving G'.

Theorem 1. *The exploitability of $\bar{b}_i^t$ computed by* FPIRA *applied to perfect recall two-player zero-sum EFG G' is exactly equal to the exploitability of $\bar{b'}_i^t$, computed by FP applied to G' in all iterations t and for all i.*

Proof. Assume that the initial strategies $\bar{b}_1^0, \bar{b}_2^0$ in FPIRA and initial strategies $\bar{b'}_1^0, \bar{b'}_2^0$ in the FP are realization equivalent, additionally assume that the same tie breaking rules are used

when more than one best response is available. We prove the Theorem by induction. If

$$\forall b_{-i} \in \mathcal{B}^p_{-i} : u_{-i}(\bar{b}^t_i, b_{-i}) = u_{-i}(\bar{b'}^t_i, b_{-i}), \quad (1)$$

$$\forall b_i \in \mathcal{B}^p_i : u_i(b_i, \bar{b}^t_{-i}) = u_i(b_i, \bar{b'}^t_{-i}), \quad (2)$$

where $\mathcal{B}^p$ is the set of pure behavioral strategies in G', then

$$b_{-i} \in \mathcal{B}^p_{-i} : u_{-i}(\bar{b}^{t+1}_i, b_{-i}) = u_{-i}(\bar{b'}^{t+1}_i, b_{-i}).$$

First, the initial step trivially holds from the initialization of strategies. Now let us show that the induction step holds. Let b^t_i be the best response chosen in iteration t in FPIRA and ${b'}^t_i$ be the best response chosen in t in FP. From (2) and the use of the same tie breaking rule we know that $b^t_i = {b'}^t_i$. From Lemma 1 we know that

$$\forall b_{-i} \in \mathcal{B}^p_{-i} : u_{-i}(\bar{b'}^{t+1}_i, b_{-i}) = \frac{t_i}{t_i+1} u_{-i}(\bar{b'}^t_i, b_{-i}) + \frac{1}{t_i+1} u_{-i}({b'}^t_i, b_{-i}),$$

However, same holds also for $\bar{b}^{t+1}_i$ since FPIRA creates G^{t+1} from G^t so that $\Delta^t_i = 0$. Hence

$$\forall b_{-i} \in \mathcal{B}^p_{-i} : u_{-i}(\bar{b}^{t+1}_i, b_{-i}) = \frac{t_i}{t_i+1} u_{-i}(\bar{b}^t_i, b_{-i}) + \frac{1}{t_i+1} u_{-i}(b^t_i, b_{-i}).$$

From (1) and from the equality $b^t_i = {b'}^t_i$ follows that

$$\forall b_{-i} \in \mathcal{B}^p_{-i} u_{-i}(\bar{b}^{t+1}_i, b_{-i}) = u_{-i}(\bar{b'}^{t+1}_i, b_{-i}),$$

and therefore also

$$\max_{b_{-i} \in \mathcal{B}^p_{-i}} u_{-i}(\bar{b}^{t+1}_i, b_{-i}) = \max_{b_{-i} \in \mathcal{B}^p_{-i}} u_{-i}(\bar{b'}^{t+1}_i, b_{-i}). \quad \square$$

4.4 Memory and Time Efficiency

FPIRA needs to store the average behavioral strategy for every action in every information set of the solved game, hence storing the average strategy in G^t instead of G' results in significant memory savings directly proportional to the decrease of information set count. Additionally, when the algorithm computes $\widetilde{b}^t_i$, it can temporarily refine the information set structure of G^t only in the parts of the tree that can be visited when playing the pure best response b^t_i according to $\mathcal{I}'_i$ to avoid representing and storing G'. Moreover, one typically does not have to store and traverse the whole game tree when computing a best response. When storing the b^t_i, we do not store behavior in the parts of the game unreachable due to actions of i. For this reason, there are typically large parts of the game tree omitted, since i plays only 1 action in his information sets. Hence, the best response computation does not prevent us from solving large domains with excessive memory requirements (we provide results showing that the best responses are small in Section 5). Finally, efficient domain-specific implementations of best response (e.g., on poker [Johanson *et al.*, 2011]) can be employed to further reduce the memory and time requirements. The iteration of FPIRA takes approximately twice the time needed to perform one iteration of FP in G', as it now consists of the standard best response computation in G', the modified best response computation to obtain Δ^t_i and two updates of average behavioral strategies (which are faster than the update in G' since the average strategy is smaller).

5 Experiments

We introduce the domains used for experimental evaluation of FPIRA. We follow by the discussion of the convergence and the size of abstractions needed to solve these domains.

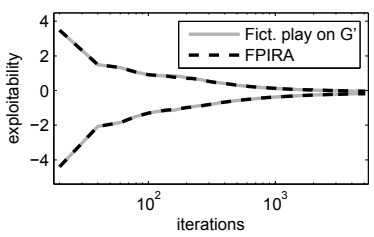
Leduc Hold'em. Leduc Hold'em is a two-player poker, which is used as a common benchmark in imperfect-information game solving because it is small enough to be solved but still strategically complex. There is a deck of cards with a given number of card types and a given number of cards per type (in standard Leduc hold'em there are 3 types of cards, 2 cards for each type). 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 follows. Every player can bet from a limited set of allowed values or check. After a bet, the player can raise, again choosing the value from a limited set, call or forfeit the game by folding. The number of consecutive raises is limited. A public shared card is dealt after one of the players calls or after both players check. Another round of betting takes place with identical rules. The player with the highest pair wins. If none of the players has a pair, the player with the highest card wins.

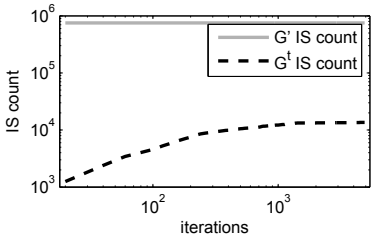
Random Games. Since there is no standardized collection of benchmark EFGs, we use randomly generated games to obtain statistically significant results. We randomly generate a perfect recall game with varying depth and branching factor 5. To control the information set structure, we use observations assigned to actions – for player i, nodes h with the same observations generated by all actions in history belong to the same information set. Every action generates a temporary utility which is randomly generated in the interval $(-2, 2)$. The utility for player 1 in every leaf is then computed as the sum of the temporary utilities of actions leading from the root of the game to the leaf. In this way, we create more realistic games, with the notion of good and bad moves.

5.1 Results

In all the presented results, FPIRA was terminated when the difference of the expected values of best responses against the average strategies was below 10^{-2}.

Leduc Hold'em. In Figure 2 (a) we show the exploitability of the average strategies computed by the FP and FPIRA (y-axis) as a function of iterations (log x-axis) on Leduc Hold'em with 4 card types, 3 for each type, 2 possible bet and raise values and 4 consecutive raises allowed. The observed identical convergence rate of FPIRA and the FP is a direct consequence of Theorem 1. Both algorithms needed $\sim 5{\cdot}10^3$ iterations to converge to the gap 10^{-2} in the expected values of best responses against the average strategies. In Figure 2 (b) we show the information set count of G^t and G' (log y-axis) as a function of iterations (log x-axis) in the same setting. As expected, the highest increase in the information set count of G^t is at the beginning of the algorithm since the information set structure is extremely coarse and the strategies vary significantly between iterations. In the later stages of the convergence, we observe almost no changes in the information set structure. Additionally, in Figure 2 (c) we present relative information set counts for initial and final abstraction, relative size of the largest best response which needed to be





	G^1 size	G^∞ size	BR size	G' size
b. 2, r. 2, c.r. 1	4.4%	31.1%	10.7%	8216
b. 3, r. 3, c.r. 1	1.8%	22.7%	4.0%	30200
b. 4, r. 4, c.r. 1	0.9%	15.9%	2.1%	81016
b. 5, r. 5, c.r. 1	0.6%	8.5%	0.9%	179096
b. 2, r. 2, c.r. 2	1.4%	16.6%	3.0%	40600
b. 2, r. 2, c.r. 3	0.4%	5.6%	0.7%	179096
b. 2, r. 2, c.r. 4	0.1%	1.8%	0.2%	751000

Figure 2: (a) The exploitability of the average strategies on Leduc Hold'em with 4 card types, 3 for each, 2 bet and raise values and 4 consecutive raises allowed. (b) The information set count of G^t and G' in the same setting. (c) Relative information set counts of G^1 and final abstraction G^∞, relative size of the largest best response which needed to be stored, and total information set count of G' for Leduc holdem with deck with 4 types of cards, 3 cards for each, for increasing number of bets and raises (rows 1 to 4) and increasing consecutive raise count (row 1, 5 - 7)

stored and total information set count of the original perfect recall game in poker domains, first when increasing number of bets and raises and when increasing number of allowed consecutive raises. The size of both abstractions decreases in both parameters, hence the sizes of the average strategies stored in FPIRA decrease proportionally. Besides storing the average strategy in the current abstraction, FPIRA also stores the best response computed in G' before updating the average strategy. To show the maximal memory needed to store these best responses we provide the number of information sets of G' for which there is an action prescribed in the largest best response during the run of FPIRA. These results show that the sizes of the best responses do not threaten the memory efficiency of FPIRA. This leads to significant memory savings in large domains with only approximately twice as much time needed per iteration compared to the FP applied directly to the perfect recall game. Additionally, the results suggest that further scaling of the solved domains will significantly reduce the relative size of the abstractions. This observation is further supported by the fact that the relative support size of Nash equilibria in poker-like domains decreases as the game size increases [Schmid *et al.*, 2014].

Random games. To show the performance of FPIRA in games with varying structure, we performed additional experiments on 50 instances of random games for depth 7 and 30 instances for depth 8. FPIRA was able to solve the instances using abstractions which had on average only $4.9\% \pm 0.7\%$ of the information sets of the original game for depth 7 and $1.2\% \pm 0.3\%$ for depth 8. The perfect recall instances had on average $2.6 \cdot 10^4 \pm 4 \cdot 10^3$ and $1.1 \cdot 10^5 \pm 3 \cdot 10^4$ information sets for depth 7 and 8 respectively. Notice that the sizes of abstractions needed to solve the random games are significantly smaller than in the case of poker with similar information set counts. This is caused by the moves of nature at the start of poker, which cause large parts of the game tree to be visited when playing according to one pure strategy. Hence when computing Δ_i^t, there is a significantly higher number of pure strategies of $-i$ that we need to check, and therefore more information set splits, compared to games with no nature.

6 Conclusion

We present the first algorithm that automatically creates imperfect recall abstractions and uses them to solve the original extensive-form game with perfect recall. While the imperfect recall abstractions of perfect recall games can significantly reduce the space necessary for storing strategies, their use has been rather limited. Previous works use either very restricted subclasses of imperfect recall abstractions [Lanctot *et al.*, 2012; Kroer and Sandholm, 2016; Bošanský *et al.*, 2015], heuristic approaches [Waugh *et al.*, 2009], or only focus on solving the given imperfect recall game [Čermák *et al.*, 2017].

Our main contribution is the memory efficient algorithm FPIRA that provides an automatically created imperfect recall abstraction that is sufficient for a Nash equilibrium strategy in the original perfect recall game. As a consequence, a strategy from the abstracted game can be used directly in the original game without the need to use translation techniques, nor is the quality of such strategy affected by choice of the abstraction. The FPIRA algorithm is based on fictitious play (FP) and provides the same guarantees of the convergence as if FP were applied directly to the original perfect recall game. The experimental evaluation shows that we are able to solve modification of Leduc hold'em with $7 \cdot 10^5$ information sets using an abstracted game with 1.8% of information sets of the original game. Furthermore, the experiments on different versions of Leduc hold'em and random games suggest that the relative size of the needed abstractions significantly decreases as the size of the solved games increases.

A natural step for future work is to generalize FPIRA to other learning algorithms – e.g., Counterfactual Regret Minimization (CFR) [Zinkevich *et al.*, 2008] that is used for solving large extensive-form games with perfect recall. However, this generalization is not straightforward since the updates, and the identification of whether an information set should be split is significantly more challenging in that case.

Acknowledgments

This research was supported by the Czech Science Foundation (grant no. 15-23235S), and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/235/OHK3/3T/13. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures".

References

[Bošanský *et al.*, 2015] Branislav Bošanský, Albert Xin Jiang, Milind Tambe, and Christopher Kiekintveld. Combining compact representation and incremental generation in large games with sequential strategies. In *AAAI*, pages 812–818, 2015.

[Bošanský *et al.*, 2016] Branislav Bošanský, Viliam Lisý, Marc Lanctot, Jiří Čermák, and Mark H.M. Winands. Algorithms for Computing Strategies in Two-Player Simultaneous Move Games. *Artificial Intelligence*, 237:1–40, 2016.

[Bošanský *et al.*, 2014] Branislav Bošanský, Christopher Kiekintveld, Viliam Lisý, and Michal Pěchouček. An Exact Double-Oracle Algorithm for Zero-Sum Extensive-Form Games with Imperfect Information. *Journal of Artificial Intelligence Research*, 51:829–866, 2014.

[Brown and Sandholm, 2015] Noam Brown and Tuomas W Sandholm. Simultaneous abstraction and equilibrium finding in games. In *AAAI*, 2015.

[Brown, 1949] George W Brown. Some notes on computation of games solutions. Technical report, DTIC Document, 1949.

[Čermák *et al.*, 2017] Jiři Čermák, Branislav Bošanskỳ, and Michal Pěchouček. Combining incremental strategy generation and branch and bound search for computing maxmin strategies in imperfect recall games. In *AAMAS*, pages 902–910, 2017.

[Chen and Bowling, 2012] Katherine Chen and Michael Bowling. Tractable objectives for robust policy optimization. In *NIPS*, pages 2078–2086, 2012.

[Daskalakis and Pan, 2014] Constantinos Daskalakis and Qinxuan Pan. A counter-example to karlin's strong conjecture for fictitious play. In *Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2014.

[Gilpin and Sandholm, 2007] Andrew Gilpin and Tuomas Sandholm. Lossless Abstraction of Imperfect Information Games. *Journal of the ACM*, 54(5), 2007.

[Gilpin *et al.*, 2007] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *AAAI*, pages 50–57, 2007.

[Heinrich *et al.*, 2015] Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *ICML*, pages 805–813, 2015.

[Johanson *et al.*, 2011] Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. In *IJCAI*, volume 11, pages 258–265, 2011.

[Karlin, 2003] Samuel Karlin. *Mathematical methods and theory in games, programming, and economics*, volume 2. Courier Corporation, 2003.

[Kroer and Sandholm, 2014] Christian Kroer and Tuomas Sandholm. Extensive-Form Game Abstraction with Bounds. In *EC*, pages 621–638. ACM, 2014.

[Kroer and Sandholm, 2016] Christian Kroer and Tuomas Sandholm. Imperfect-Recall Abstractions with Bounds in Games. In *EC*, pages 459–476. ACM, 2016.

[Kuhn, 1953] Harold W. Kuhn. Extensive Games and the Problem of Information. *Contributions to the Theory of Games*, II:193–216, 1953.

[Lanctot *et al.*, 2012] Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-Regret Learning in Extensive-Form Games with Imperfect Recall. In *ICML*, pages 1–21, 2012.

[Lisý *et al.*, 2016] Viliam Lisý, Trevor Davis, and Michael Bowling. Counterfactual Regret Minimization in Sequential Security Games. In *AAAI*, 2016.

[Moravčík *et al.*, 2017] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 2017.

[Robinson, 1951] Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[Schmid *et al.*, 2014] Martin Schmid, Matej Moravcik, and Milan Hladik. Bounding the support size in extensive form games with imperfect information. In *AAAI*, pages 784–790, 2014.

[von Stengel, 1996] Bernhard von Stengel. Efficient Computation of Behavior Strategies. *Games and Economic Behavior*, 14:220–246, 1996.

[Waugh *et al.*, 2009] Kevin Waugh, Martin Zinkevich, Michael Johanson, Morgan Kan, David Schnizlein, and Michael H Bowling. A Practical Use of Imperfect Recall. In *Symposium on Abstraction, Reformulation and Approximation (SARA)*, 2009.

[Zinkevich *et al.*, 2008] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret Minimization in Games with Incomplete Information. In *NIPS*, pages 1729–1736, 2008.