# An extensive-form fictitious play algorithm with local best decision updates

Tim P. Schulze[1*]

[1]Department of Mathematics, University of Tennessee, 1403 Circle Dr.,
Knoxville, 37916, TN, USA.

Corresponding author(s). E-mail(s): tschulze@utk.edu;

**Abstract**

We introduce a simple extensive-form algorithm for finding equilibria of two-player, zero-sum games. The algorithm is realization equivalent to a generalized form of Fictitious Play. We compare its performance to that of a similar extensive-form fictitious play algorithm and a counter-factual regret minimization algorithm. All three algorithms share the same advantages over normal-form fictitious play in terms of reducing storage requirements and computational complexity. The new algorithm is intuitive and straightforward to implement, making use of a locally optimized best decision update instead of the best response update of traditional fictitious play.

**Keywords:** game theory, fictitious play, differential inclusion, poker

## 1 Introduction

In recent years there has been a great deal of progress in computational methods for solving large games. Interest in the subject stems from both practical applications where AIs, such as self-driving vehicles, interact with each other and humans, and from a handful recreational games, such as chess, poker and Go, that are seen as challenging surrogates for real-world applications while simultaneously appealing to a large population of devoted enthusiasts. In particular, work on the popular variant of poker known as Texas Hold'em has seen many years of progress culminate in a number of high-profile success stories. Poker and other card games are especially challenging, as they are games with imperfect information and a large number of game states. The development of the Counter-Factual Regret Minimization (CFR) algorithm [1] marked a significant advance in solving large extensive-form games, eventually leading to the numerical solution of the two-player, limit version of Texas Hold'em [2]. This was followed by other successful AIs that defeated top professional poker players in heads-up no-limit [3] and multi-player no-limit [4] Texas Hold'em.

It has been recognized from the earliest days of game theory that using *behavior strategies* is often preferable to *mixed strategies* for analyzing large games. For example, while the term had not yet been introduced, von Neumann and Morgenstern use behavior strategies to analyze the poker poker model we review in Section 4.1 [5]. Despite this, many computational methods use mixtures, $\sigma^i(s)$, of pure strategies, $s$, that specify a specific action to be taken by player $i$ at every game state that player may encounter. The mixtures represent the probability with which that particular pure strategy will be played, so that $\sum_{s\in S^i} \sigma^i(s) = 1$, with $S^i$ being the set of all pure strategies for player $i$. Fictitious Play (FP), for example, is one of the oldest computational methods for solving games [6, 7]. In its original formulation, it is a method for finding a Nash Equilibrium (NE) in two-player, zero-sum, normal form games. In this method, the average of the prior play is iteratively updated to

$$\sigma^i_{n+1} = \left(1 - \frac{1}{n+1}\right)\sigma^i_n + \frac{1}{n+1}\beta^i(\sigma^{-i}_n), \quad \forall i \in \{1,2\}, \tag{1}$$

where $n$ indicates the time-step, $-i$ indicates player $i$'s opponent (not $i$), $\beta^i(\sigma^{-i}_n)$ is a best response to the opponent's play on the previous time-step:

$$\beta^i(\sigma^{-i}_n) \in \arg\max_{\sigma^i} u^i(\sigma^i, \sigma^{-i}_n), \tag{2}$$

and $u^i(\sigma^i, \sigma^{-i})$ is expected utility from player $i$'s perspective, accounting for the mixed strategies of both players and the role of chance. In other words, each player is simultaneously updating their strategy to be a best response to their opponent's average strategy over the history of their play. One can also implement this by alternating the updates, so that the best response accounts for the opponent's freshly updated average play. For the most part, we will focus on algorithms with simultaneous updates, but alternating updates will be briefly addressed later in the paper.

While the normal form of a game is often preferable for analyzing games in general, it balloons the computational cost and amount of storage required for games. In the end, the extensive form of a game provides a more compact way of representing strategies despite its more cumbersome notation. To define a game in extensive form, we begin with a *game tree*, which consists of a set of vertices $\{x\}$ (also referred to as nodes), including the *root*, denoted $x_0$, of the tree, and a set of directed edges $\{e\}$ that correspond to the moves players can make in the game. To qualify as a game tree, we require that each vertex $x$ can be reached by following a unique path from the root. In this paper, we will be exclusively interested in two-player games, with players $i \in \{1, 2\}$. To account for chance moves, we expand this set to include a *chance player*, denoted as $\{0\}$. The terminal vertices are referred to as leaves $\ell \in L$, and the non-leaf vertices are partitioned among the players according to who will act at a given vertex. We are interested in games with imperfect information, where the players may not know which vertex they are at. This possible uncertainty partitions the subset of vertices associated with the non-chance players into subsets $I \in \mathcal{I}$ referred to as *information sets*. Similarly, the edges emerging from the nodes in information sets are partitioned into equivalence classes referred to as *actions* $a \in A(I)$. We will find it convenient to include the chance nodes in $\mathcal{I}$ as subsets containing a single vertex, and we will let $\{\mathcal{I}^i\}$ be a partition of $\mathcal{I}$ over $i \in \{0, 1, 2\}$. Finally, the leaves represent the outcome of the game, which we express in terms of utility functions for the non-chance players $\mathcal{U}^i(\ell)$. We will consider only *zero-sum* games where $\mathcal{U}^i(\ell) = \mathcal{U}^{-i}(\ell)$.

While mixed strategies can also be defined in the extensive form as probability mixtures of pure strategies, where a specific action is specified at every information set, behavior strategies provide a more compact way of representing a strategy by assigning a distribution, $b(I, a) \geq 0$, over the actions $a \in A(I)$ available at each information set $I \in \mathcal{I}$:

$$\sum_{a \in A(I)} b(I, a) = 1,$$

including those controlled by the chance player, who plays a fixed strategy. When players have perfect recall—they do not forget any information they knew in the past—there exists a strategy of either type equivalent to a given strategy of the other type [8, 9].

The use of behavior strategies was one of the features that allowed the CFR algorithm and its derivatives, like CFR+ [10], to achieve the successes mentioned above. A key advantage of the behavior strategy description lies in being able to easily compute the expected utility, $U(I, a; b)$, of a specific action, $a \in A(I)$, within the set of actions available at a given information set $I$, where the utility is measured from the perspective of the player who controls $I$, and $b$ represents the collection of behavior strategies for all players at all information sets. In what follows, we refer to $U(I, a; b)$ as the *action-utility*, and suppress $b$ when the strategies are clear from context, writing $U(I, a) \equiv U(I, a; b)$. Similarly, we will use $U(I, b(I)) \equiv U(I, b(I); b)$ to indicate the utility of playing a specific mixture of actions $b(I) \in \mathbb{R}^{|A(I)|}$ at $I$ that may or may not be consistent with that dictated by $b$.

More recently, a version of FP (Extensive-Form Fictitious Play or XFP) that takes advantage of the efficiency of behavior strategies and is realization equivalent to FP has also been been developed [11]. In this paper we introduce an alternative extensive form algorithm that is realization equivalent to a perturbed FP. We refer to this as Best Decision Fictitious Play (BDFP). In close analogy to FP (1), BDFP consists of a sequence of behavior strategies,

$$b_{n+1} = \left(1 - \frac{1}{n+1}\right) b_n + \frac{1}{n+1} d(b_n), \tag{3}$$

where $d(b_n)$ is the collection of what we refer to as *best decisions* that are *locally* optimized with respect to both the opponent's current strategy and *a player's own current strategy following actions* $a \in A(I)$:

$$d(I; b_n) \in \arg\max_{b(I)} U(I, b(I); b_n). \tag{4}$$

Note that, unlike the best response (2) defined above, the best decisions are defined using the action-utility at a particular information set rather than the expected utility of entire strategies. We will see that BDFP enjoys the same advantages as CFR and XFP in terms of how computational cost and storage scale with the size of the game, but with a simpler and more intuitive implementation. In practice (4) can be computed by simply selecting the best action, and this mimics the way humans think. In particular, expected value computations for what we have called action utilities are routinely discussed in the recreational poker literature, but to the extent humans can really make these calculations they focus on their immediate decision using their own current strategy and their beliefs about how their opponents play.

In the next section, we briefly review the CFR and XFP algorithms, and further introduce BDFP. In section 3, we show that BDFP is equivalent to a generalized FP, and therefore inherits its convergence properties. In section 4, we discuss a benchmark game that generalizes a classic model of poker put forward by von Neumann and Morgenstern (vN&M). In section 5, we use this game, along with another poker model, to compare the performance of the three algorithms. We summarize and conclude in the final section.

## 2 Algorithms

We start with a description of elements common to all three algorithms considered in this paper, and follow this with a discussion of each algorithm separately.

As previously mentioned, in games with imperfect information, a player may not know which node he/she is at, and must analyze their decisions based on the probability that their opponent's prior play has brought them to a particular node within the information set. In comparing the expected value of actions, a player need not consider the probability that their own prior actions will bring them to that information set. Thus the play of a player in any such calculation is assumed to have been consistent with the need to make the decision. For this reason, $U(I, a)$ is often referred to as *counter-factual utility* [1]. This "play-to-reach" assumption is common to all of the algorithms we consider in this paper.

The action-utility is thus the conditional expectation of utility, $\mathbb{E}[\mathcal{U}^i(\ell)|I, a]$, given we are at a specified information set, the player controlling that information set takes a specified action, and $\mathcal{U}^i(\ell)$ a basic utility function defined on the set of leaves in the game tree. This can be computed using the conditional probability $P^{-i}(x|I)$ of $i$'s opponent's, including the chance player, playing so as to reach a node $x \in I$ controlled by $i$, and the conditional probability $P(\ell|x, a)$ of all players playing so as to reach leaf $\ell$ starting from $I$ with action $a$:

$$\begin{aligned}
\mathbb{E}[\mathcal{U}^i(\ell)|I, a] \equiv U(I, a) &= \sum_{\ell \in L} P(\ell|I, a)\mathcal{U}^i(\ell), \quad I \in \mathcal{I}^i, && (5) \\
&= \sum_{x \in I} P^{-i}(x|I) \sum_{\ell \in L_{x,a}} P(\ell|x, a)\mathcal{U}^i(\ell) && (6)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{x \in I} \frac{P^{-i}(x)}{P^{-i}(I)} \sum_{\ell \in L_{x,a}} P(\ell|x,a)\mathcal{U}^i(\ell), \quad P^{-i}(I) > 0, \\
&= \frac{1}{P^{-i}(I)} \sum_{x \in I} \sum_{\ell \in L_{x,a}} P^{-i}(\ell) P^i(\ell|x,a)\mathcal{U}^i(\ell), \qquad (7)
\end{aligned}
$$

where the inner sum is over leaves, $L_{x,a}$, that can be reached using action $a$ at node $x$.

The various "reach" probabilities can be computed from the behavior strategies $b(I, a)$ and the unique sequence of actions starting at the root of the game tree and terminating at a node $x$: $a_1^x, a_2^x, \ldots, a_{J_x}^x$, where there are $J_x$ actions along the path leading to $x$. Letting $\hat{I}(a)$ indicate the information set at which action $a$ is taken, we have

$$
\begin{aligned}
P^{-i}(x) &= \prod_{j=1, \hat{I}(a_j^x) \notin \mathcal{I}^i}^{J_x} b(\hat{I}(a_j^x), a_j^x), \qquad (8) \\
P^{-i}(I) &= \sum_{x \in I} P^{-i}(x), \qquad (9) \\
P(\ell|x,a) &= \prod_{j=J_x+2}^{J_\ell} b(\hat{I}(a_j^\ell), a_j^\ell), \qquad (10)
\end{aligned}
$$

where the behavior coefficient for $a$, action $J_x + 1$, is omitted in the last product, as the probability is conditioned on that choice.

In the rest of this section, we describe the three algorithms considered in this paper.

## 2.1 Counter-factual Regret Minimization

The basic version of CFR [1] is now sometimes referred to as "vanilla" CFR, and is a popular entry point for those getting started with reinforcement learning (RL). While the authors go beyond this version, adapting it to specific features of Texas Hold'em, and there have been subsequent developments, most notably CFR+ [10], we will be considering only this basic version.

CFR is based on the notion of *regret* for having played the game according to the current strategy $b(I)$ rather than taking a specific action $a$ at information set $I$:

$$
U(I, a) - U(I, b(I)),
$$

where

$$
U(I, b(I)) = \sum_{a \in A(I)} b(I, a) U(I, a). \qquad (11)
$$

More specifically, CFR maintains the average regret, weighted by the opponent's reach probabilities $P^{-i}(I; b_n)$:

$$
R_n(I, a) = \frac{1}{n} \sum_{k=1}^{n} P^{-i}(I; b_k) \left( U(I, a; b_k) - U(I, b_k(I); b_k) \right), \quad I \in \mathcal{I}^i. \qquad (12)
$$

Notice that the opponent's reach probability appears as the normalization factor in the computation of the action-utilities (7), canceling the weighting factor and eliminating the need to compute these quantities unless one actually wishes to compute the utility. At the same time, this removes the possibility of division by zero should $P^{-i}(I) = 0$.

The strategy at the next iteration is proportional to the amount of positive regret

$$b_{n+1}(I, a) = \begin{cases} \frac{\max(R_n(I,a),0)}{\sum\limits_{\tilde{a}\in A(I)} \max(R_n(I,\tilde{a}),0)}, & \text{if } \sum\limits_{\tilde{a}\in A(I)} \max(R_n(I,\tilde{a}), 0) > 0, \\ \frac{1}{|A(I)|}, & \text{otherwise.} \end{cases} \tag{13}$$

Finally, it is the average of the sequence of strategies $b_n(I, a)$, weighted by the reach probability of the player who controls $I$, that converges to a NE:

$$\bar{b}_n(I, a) = \frac{\sum_{k=1}^n P^i(I; b_k) b_k(I, a)}{\sum_{k=1}^n P^i(I; b_k)}, \quad I \in \mathcal{I}^i \tag{14}$$

$$P^i(x) = \prod_{j=1, \hat{I}(a_j^x)\in\mathcal{I}^i}^{J_x} b(\hat{I}(a_j^x), a_j^x), \tag{15}$$

$$P^i(I) = \sum_{x\in I} P^i(x). \tag{16}$$

## 2.2 Extensive-Form Fictitious Play

The key advantage of CFR over (normal form) FP is the ability to efficiently store and compute with behavior strategies rather than mixed strategies. XFP also has this feature Heinrich et. al. [11] construct [11] XFP as a sequence of behavior strategies that is realization equivalent to the sequence of mixed strategies

$$\sigma_{n+1}^i = (1 - \alpha_{n+1})\sigma_n^i + \alpha_{n+1}\beta^i(\sigma_n^{-i}), \tag{17}$$

where $\beta^i$ is a best response to the opponent's current strategy $\sigma_n^{-i}$. This is a generalized FP with weights $\alpha_n$ that decay to zero with a diverging sum $\sum \alpha_n = \infty$, and reduces to the classic FP algorithm when $\alpha_n = \frac{1}{n}$.

The proof that BDFP converges, presented in Section 3 below, shares some common features with the proof that XFP converges, so we review this result here. In particular, the Heinrich et. al. result relies on a result of Leslie and Collins [12], who define the following class of generalized fictitious play algorithms, and then proceed to show that any such algorithm converges to a NE of a zero-sum game. In what follows, $\sigma$ without the superscript indicates a vector of mixed strategies, $(\sigma^1, \sigma^2)$, one for each player, and $\Sigma$ is the set of all such strategy vectors.

**Definition 1.** *A* generalized weakened fictitious play process *is any process* $\{\sigma_n\}_{n\geq 0}$, *with* $\sigma_n \in \Sigma$, *such that*

$$\sigma_{n+1} \in \{(1 - \alpha_{n+1})\sigma_n + \alpha_{n+1}(\beta_{\epsilon_n}(\sigma_n) + M_{n+1})\}_{\beta_{\epsilon_n}}, \tag{18}$$

*where $\beta_{\epsilon_n} = (\beta^1_{\epsilon_n}, \beta^2_{\epsilon_n})$ is in the set of $\epsilon_n$-best response vectors, $\alpha_n \to 0$, $\epsilon_n \to 0$ as $n \to \infty$,*

$$\sum_{n\geq 1} \alpha_n = \infty,$$

*and $\{M_n\}_{n\geq 1}$ is a sequence of perturbations such that, for any $T > 0$,*

$$\lim_{n\to\infty} \sup_k \{||\sum_{j=n}^{k-1} \alpha_{j+1} M_{j+1}|| : \sum_{j=1}^{k-1} \alpha_{j+1} \leq T\} = 0.$$

Note that in (17), $\epsilon = 0$ and $\beta^i$ is a best response. Neither XFP nor BDFP make use of $\epsilon$-best responses, and we will later use the $\epsilon$ subscript to indicate a strategy in a perturbed game where actions must be taken with finite probability.

The Leslie & Collins theorem relies on a result of Benaïm, Hofbauer & Sorin, (2006) [**?** ] that we will need in the following section. We present the theorem in the form given by Leslie & Collins.

**Theorem 1** (Benaïm, et. al.)**.** *Assume $F : \mathbb{R}^m \to \mathbb{R}^m$ is a closed set-valued map such that $F(\sigma)$ is a non-empty compact convex subset of $\mathbb{R}^m$ with*

$$\sup_{z\in F(\sigma)} ||z|| \leq c(1 + ||\sigma||) \quad \forall \sigma.$$

*Let $\{\sigma_n\}_{n\geq 0}$ be the process satisfying*

$$\sigma_{n+1} - \sigma_n - \alpha_{n+1} M_{n+1} \in \alpha_{n+1} F(\sigma_n),$$

*with $\alpha_n \to 0$ as $n \to \infty$,*

$$\sum_{n\geq 1} \alpha_n = \infty,$$

*and $\{M_n\}_{n\geq 1}$ be a sequence of perturbations such that, for any $T > 0$,*

$$\lim_{n\to\infty} \sup_k \{||\sum_{j=n}^{k-1} \alpha_{j+1} M_{j+1}|| : \sum_{j=1}^{k-1} \alpha_{j+1} \leq T\} = 0.$$

*The set of limit points of $\{\sigma_n\}$ is a connected internally chain-recurrent set of the differential inclusion*

$$\frac{d}{dt}\sigma(t) \in F(\sigma(t)). \tag{19}$$

To apply Theorem 1 in the context of game theory, Benaim, Hofbauer & Sorin, (2006) extend the domain of the best response function $\beta(\sigma)$ to all of $R^m$ by associating points outside the simplex of mixed strategies $\Sigma^m$ with the unique closest point within the simplex. With this understanding, Leslie and Collins first show that any GFP (18) satisfies the requirements of this theorem with

$$\frac{d}{dt}\sigma(t) \in F(\sigma(t)) = \{\beta(\sigma(t)) - \sigma(t)\}_\beta, \tag{20}$$

and then show that the set of limit points is the set of NE.

**Theorem 2** (Leslie and Collins)**.** *Any generalized weakened fictitious play process will converge to the set of NE in two-player zero-sum games, potential games, and generic 2 × 2 games.*

Finally, Heinrich et. al. show that the mapping from mixed strategies (17) to behavior strategies requires

$$b_{n+1}(I,a) = b_n(I,a) + \frac{\alpha_{n+1} P^i(I; B^i_{n+1})(B^i(I,a;b^{-i}_n) - b_n(I,a))}{(1-\alpha_{n+1})P^i(I;b^i_n) + \alpha_{n+1} P^i(I;B^i)}, \quad I \in \mathcal{I}^i, \tag{21}$$

where $B^i$ is a best response behavior strategy to $b^{-i}_n$:

$$B^i_{n+1} \in \arg\max_{b^i} u^i(b^i, b^{-i}_n), \tag{22}$$

and $P^i$ is player $i$'s reach probability for either the current strategy or the current best response.

The calculation of the best response $B^i$ is similar to the calculation of the best decisions. In either case the action-utility must be computed at every information set, and those utilities are used to make a best decision at each node. In the case of best responses, we work backward through the game tree from the leaves toward the root, using the previously computed best decisions to calculate the utility at the next level. Thus, at each level we are choosing a best option that will be followed by best options at every subsequent information set, so that we end up with a best response.

## 2.3 Best Decision Fictitious Play

In practice, best decisions (4) can be computed by simply choosing any optimal action:

$$a \in \arg\max_{\tilde{a} \in A(I)} U(I, \tilde{a}; b_n). \tag{23}$$

Like XFP, BDFP can be implemented with a more general weight $\alpha_n$, but we chose to implement this with $\alpha_n = \frac{1}{n}$, the weight used in classical FP. This is equivalent to a simple average of best decisions that can be computed by counting the number of times each action is best at a given information set:

$$c_{n+1}(I,a) = c_n(I,a) + 1. \tag{24}$$

One can then replace the update (3) with a non recursive formula

$$b_n(I,a) = \frac{c_n(I,a)}{n}. \tag{25}$$

Table 1 compares the key steps required for BDFP to those required for CFR and XFP. All three algorithms start by computing the action utilities at every information set, with XFP simultaneously calculating a best response for each player to the other

**Table 1** Schematic outline for each of the three algorithms under consideration.

| CFR | XFP | BDFP |
|---|---|---|
| For $i \in \{1,2\}$ | For $i \in \{1,2\}$ | For $i \in \{1,2\}$ |
| For $I \in \mathcal{I}^i$ & $a \in A(I)$ | For $I \in \mathcal{I}^i$ & $a \in A(I)$ | For $I \in \mathcal{I}^i$ & $a \in A(I)$ |
| Update $U(I,a)$ using (7) | Update $B$ using (22) | Update $U(I,a)$ using (7) |
| For $i \in \{1,2\}$ | For $i \in \{1,2\}$ | For $i \in \{1,2\}$ |
| For $I \in \mathcal{I}^i$ | For $I \in \mathcal{I}^i$ | For $I \in \mathcal{I}^i$ |
| Update $U(I,b)$ using (11) | Update $P^i(I;b)$ using (16) | Choose $a$ using (23) |
| For $a \in A(I)$ | Update $P^i(I;B)$ using (16) | Update $c_n$ using (24) |
| Update regrets using (12) | For $a \in A(I)$ | For $a \in A(I)$ |
| Update $b_n$ using (13) | Update $b_n$ using (21) | Update $b_n$ using (25) |
| Update $\bar{b}_n$ using (14) | | |

player's *entire* current strategy. This is the most costly part of all three algorithms, requiring on the order of $\bar{N}(|\mathcal{I}^1| + |\mathcal{I}^2|)$ operations, where $\bar{N}$ is the average number vertices per information set. The remaining steps scale with the number of information sets $|\mathcal{I}^1| + |\mathcal{I}^2|$. For CFR, this includes calculating the expected values for the current strategy profiles $b_n(I)$ using (11), calculating the regrets (12), updating $b_n$ using (13), and updating $\bar{b}_n$ using (14). For XFP we must calculate two sets of reach probabilities (16) and update $b_n$ using (21). For BDFP, we select a best option (23), update the appropriate counter at each information set (24), and update $b_n$ using (25).

Finally, we note that the regret calculation (12) used in CFR is similar to the best decision calculation (4) in that it is updated using the behavior strategies $b_n$ at the previous time-step, and depends on both the opponent's strategy and a player's own strategy at information sets that are encountered after the one that is being updated. As a result, the BDFP and CFR updates can be done by visiting the information sets in any order, a feature that may be useful in learning algorithms that sample nodes randomly. This is in contrast to the XFP update, which makes use of the freshly updated behavior coefficients as it works its way from the leaves to the root of the game tree.

## 3 Convergence of Best Decision Fictitious Play

To prove that BDFP converges, we first expand Definition 1 to include an alternative "better" response. We then show that BDFP is equivalent to one of these expanded GFPs. Next, we use Theorem 1 to adapt Theorem 2 to establish convergence. Finally we adapt a theorem due to Hofbauer and Sorin [13] to prove that the attractive set is the set of NE.

We will need the mapping from behavior strategies to mixed strategies:

$$\sigma^i(s;b) = \prod_{I \in \mathcal{I}^i} b(I, a(s,I)), \tag{26}$$

where where $a(s, I)$ is the action $a \in A(I)$ consistent with the pure strategy $s \in S^i$. The better response required to show that (3) converges takes the form

$$\begin{aligned}\delta^i(s; b) &= \frac{1}{|\mathcal{I}^i|} \sum_{\bar{I} \in \mathcal{I}^i} d(\bar{I}, a(s, \bar{I}); b) \prod_{I \neq \bar{I}, I \in \mathcal{I}^i} b(I, a(s, I)) \\ &= \frac{1}{|\mathcal{I}^i|} \sum_{I \in \mathcal{I}^i} \frac{\sigma^i(s; b) d(I, a(s, I); b)}{b(I, a(s, I))}, \qquad (27)\end{aligned}$$

where $d$ is the best decision (4) introduced earlier. Note that $\delta^i$ is a mixed strategy, whereas $d$ is a behavior strategy, and that these depend on the strategies of both opponents. As we did with $\sigma$, we will use $\delta$ without a superscript to indicate the vector of better responses, one for each player.

**Theorem 3.** *BDFP is realization equivalent to a generalized weakened FP with best responses $\beta_\epsilon(\sigma)$ replaced by weakened better decisions $\delta$.*

*Proof.* Inserting (4) into the mapping from behavior strategies to mixed strategies gives

$$\begin{aligned}\sigma^i_{n+1}(s; b_{n+1}) &= \prod_{I \in \mathcal{I}^i} b_{n+1}(I, a(s, I)) \\ &= \prod_{I \in \mathcal{I}^i} \left[ \left(1 - \frac{1}{n+1}\right) b_n(I, a(s, I)) + \frac{1}{n+1} d(I, a(s, I); b_n) \right].\end{aligned}$$

Next, we isolate terms of $O(\frac{1}{n})$ and larger from the product

$$\begin{aligned}\sigma^i_{n+1}(s; b_{n+1}) &= \left(1 - \frac{|\mathcal{I}^i|}{n+1}\right) \prod_{I \in \mathcal{I}^i} b_n(I, a(s, I)) \\ &\quad + \frac{1}{|\mathcal{I}^i|} \sum_{\bar{I} \in \mathcal{I}^i} \frac{|\mathcal{I}^i|}{n+1} d(\bar{I}, a(s, \bar{I}); b_n) \prod_{I \neq \bar{I}, I \in \mathcal{I}^i} b_n(I, a(s, I)) \\ &\quad + \frac{1}{n+1} M^i_{n+1}(s; b_n), \qquad (28)\end{aligned}$$

where we have grouped the finite number of higher order terms into the perturbation $M^i_{n+1}$. Letting

$$\alpha_n = \frac{1}{n},$$

and rearranging (28) gives

$$\sigma^i_{n+1} \in \{\left(1 - \alpha_{n+1} |\mathcal{I}^i|\right) \sigma^i_n + \alpha_{n+1} \left(|\mathcal{I}^i| \delta^i + M^i_{n+1}\right)\}_{\delta^i}.$$

The weights $\alpha_n$ are the same as those in standard FP and satisfy the requirements in Definition 1, while the perturbations $M^i_n = O\left(\frac{1}{n}\right)$ decay sufficiently fast to ensure the requirement

$$\lim_{n\to\infty} \sup_k \{||\sum_{j=n}^{k-1} \alpha_{j+1} M_{j+1}|| : \sum_{j=1}^{k-1} \alpha_{j+1} \leq T\} = 0.$$

□

In the Leslie & Collins (2006) result, the relevant differential inclusion is (**??**). In view of Theorem 3, we must consider instead the weakened better response defined above. Since Theorem 1 is cast in terms of mixed strategies, we re-express the better responses $\delta^i(s; b)$ as functions of $\sigma$ rather than $b$. One can do this by using the mapping from from mixed strategies to behavior strategies:

$$b(I, a; \sigma) = \frac{\sum_{s\in S_I^i(a)} \sigma^i(s)}{\sum_{s\in S_I^i} \sigma^i(s)} = \frac{\sum_{s\in S_I^i(a)} \sigma^i(s)}{P^i(I)}, \quad I \in \mathcal{I}^i, \forall a \in A(I), P^i(I) > 0, \quad (29)$$

where $S_I^i$ is the set of all pure strategies for player $i$ that reach information set $I$ (controlled by $i$), and $S_I^i(a)$ is the subset of these where $i$ plays action $a$. If player $i$ reaches $I$ with zero probability, we can assign arbitrary values to $b(I, a)$, as it will make no difference.

**Theorem 4.** *The set of limit points of a generalized weakened fictitious play process with best responses $\beta_\epsilon(\sigma)$ replaced by weakened better decisions $\delta$ is a connected internally chain-recurrent set of the differential inclusion*

$$\frac{d}{dt}\sigma(t) \in \{(|\mathcal{I}^1|, |\mathcal{I}^2|) \odot (\delta(b(\sigma(t))) - \sigma(t))\}_\delta \equiv F(\sigma(t)), \quad (30)$$

*where $\odot$ indicates the element-wise product of two vectors.*

*Proof.* When the domain of the best decision function is extended to all of $R^2$ in the same way discussed earlier for the best response function, $F(\sigma)$ satisfies the requirement of Theorem 1:

$$\sup_{z\in F(\sigma)} ||z|| \leq c(1 + ||\sigma||) \quad \forall\sigma.$$

The requirements on the perturbation $M_n$ in Definition 1 are the same as in Theorem 1, and were already shown to be satisfied in the proof of Theorem 3. □

Finally, we must show that the attractors of the differential inclusion (30) are the set of NE. We will follow the proof of Hofbauer & Sorin (2006), who show this for (20). They do this by considering the total exploitability,

$$\begin{aligned} v(t) &= V(\sigma^1(t), \sigma^2(t)) = \max_{\sigma^1} u(\sigma^1, \sigma^2) - \min_{\sigma^2} u(\sigma^1, \sigma^2) \\ &= u(\beta^1(\sigma^2), \sigma^2) - u(\sigma^1, \beta^2(\sigma^1)) \geq 0, \quad (31) \end{aligned}$$

where we have expressed the utilities $u(\sigma^1, \sigma^2)$ from player 1's perspective. They show that $v(t)$ evolving under the best response differential inclusion (20) satisfies

$$\frac{d}{dt}v(t) \leq -v(t),$$

implying

$$v(t) \leq e^{-t}v(0),$$

so that $v(t)$ decays to zero. They also show how to adapt this to the discrete dynamics of a FP process. An unexploitable strategy pair is a NE by definition.

Arguments similar to that of Hofbauer & Sorin apply to (30) if we replace the best responses with the weakened better response $\delta^i$. To this end, let $\tilde{\delta}^i(t) = \delta^i(b(\sigma(t)))$ and then define a similarly weakened measure of exploitability:

$$\tilde{v}(t) = \widetilde{V}(\sigma(t)) = u(\tilde{\delta}^1(t), \sigma^2) - u(\sigma^1, \tilde{\delta}^2(t)) \geq 0,$$

where the inequality follows from the linearity of utility in mixed strategies and each player having improved, or left unchanged, the utility of each term in (27) from their own perspective.

**Theorem 5.** *Any generalized weakened fictitious play process with best responses $\beta_\epsilon(\sigma)$ replaced by weakened better decisions $\delta(b)$ will converge to the set of NE in two-player zero-sum games.*

*Proof.* The function $\tilde{\delta}^i(t)$ is piecewise constant for almost every $t$, hence its derivative vanishes whenever it exists, giving us the following result (compare Lemma 4 in Hofbauer & Sorin (2006)):

$$\begin{aligned}
\frac{d}{dt}\tilde{v}(t) &= \nabla_{\sigma^2} u(\tilde{\delta}^1(t), \sigma^2(t)) \cdot \frac{d}{dt}\sigma^2(t) - \nabla_{\sigma^1} u(\sigma^1(t), \tilde{\delta}^2(t)) \cdot \frac{d}{dt}\sigma^1(t) && (32)\\
&= |\mathcal{I}^1| \left(u(\tilde{\delta}^1(t), \sigma^2(t)) - u(\sigma^1(t), \sigma^2(t))\right) + \\
&\quad |\mathcal{I}^2| \left(u(\tilde{\delta}^1(t), \sigma^2(t)) - u(\tilde{\delta}^1(t), \tilde{\delta}^2(t))\right) - \\
&\quad |\mathcal{I}^1| \left(u(\sigma^1(t), \tilde{\delta}^2(t)) - u(\tilde{\delta}^1(t), \tilde{\delta}^2(t))\right) - \\
&\quad |\mathcal{I}^2| \left(u(\sigma^1(t), \tilde{\delta}^2(t)) - u(\sigma^1(t), \sigma^2(t))\right) && (33)\\
&\leq -\max(|\mathcal{I}^1|, |\mathcal{I}^2|) \left(u(\tilde{\delta}^1(t), \sigma^2(t)) - u(\sigma^1(t), \tilde{\delta}^2(t))\right) \\
&= -\max(|\mathcal{I}^1|, |\mathcal{I}^2|)\tilde{v}(t), && (34)
\end{aligned}$$

where (33) follows from (30) and the linearity of the utility function in mixed strategies.

The above inequality implies

$$\tilde{v}(t) \leq e^{-\max(|\mathcal{I}^1|, |\mathcal{I}^2|)t}\tilde{v}(0),$$

with $\tilde{v}(t)$ decaying to zero. When this alternative measure of exploitability $\tilde{v}(t)$ reaches zero, there is no information set at which either player can exploit the other. Working backward through the game tree, we can then see that the overall strategy is also a best response. In other words, if we compute a best response in the manner described in Section 2.2, we can do so without changing any of the best decisions, a result that does not hold when $\tilde{v} > 0$. This means we converge to strategies where both players are playing a best response, hence we are at a NE.

□

# 4 Some Exactly Solvable Poker Models

In this section, we present an exact solution for one of the two poker models we will use for comparing BDFP to CFR and XFP in the next section. While we will also present numerical solutions for a game (Leduc poker) for which we do not have an exact solution, having the solution is helpful in a couple of ways. First, we can use the value for the game and the equilibrium strategy to verify that we have made no coding mistakes, including mistakes where the exploitability may be decaying to zero, but is doing so spuriously, or we are converging to the solution of the wrong game. Second, these games can often exhibit a continuum of solutions, in which case the algorithms endlessly drift among an infinite number of possible NE. The analysis in this section will explain why this happens, and motivate our later consideration of numerical solutions in perturbed strategy spaces.

Perhaps the best known exactly solvable poker model within the game theory and RL community is Kuhn Poker [14]. Kuhn poker is played with just a three-card deck, and the solution to this game can easily be worked out by hand. Nevertheless, we will see that it is closely related to the both of the games we end up using for our numerical experiments in the next section.

The analysis of larger games is often absent from books on game theory, which tend to focus on extremely simple games, e.g. rock-paper-scissors or the Prisoner's Dilemma. A notable exception to this occurs in *The theory of games and economics behavior* [5], which features an in depth analysis of two models for the game of poker. The text refers to these as the symmetric and asymmetric games. A footnote at the opening of this discussion reveals that these models were largely responsible for von Neumann's original exploration of game theory in the 1920's.

Of the two models considered by vN&M, the asymmetric one is more similar to the actual game of poker. Further, it is more challenging from a computational perspective, as it possesses an infinite number of NE. While this is a useful starting point for benchmarking, the game tree is not deep enough for the play-to-reach feature to matter, as neither player has any control over which of their own informative sets are reached. The game of poker, however, suggests many ways of generalizing this game into a broad family of potential benchmark games that are still relatively easy to implement.

In this section, we first review the results from vN&M for their asymmetric game, and then introduce a generalized version that features a somewhat deeper game tree.

## 4.1 Von Neumann and Morgenstern's asymmetric game

The solutions to large games can be bewildering from a human perspective. A further advantage of simplified models is that one can gain some intuitive understanding of how more complicated games work. Starting with vN&M's asymmetric game will help us understand what is happening in the extended game we consider in the next subsection.

In these two-player zero-sum games players are 'dealt' hands (private information) that take the form of random numbers, and wager on who has the higher number. Unlike Kuhn poker, the von Neumann model uses sampling with replacement, so that the cards are dealt from separate decks. The discrete version of the game, where the players are dealt random integers, $1 \leq i \leq N$ from an $N$-card deck is mentioned in vN&M, but the text principally focuses on the continuous version of the game, with hands $x \in [0, 1]$. We will refer to these as Neumann($N$) and Neumann($\infty$), respectively. The continuous version is more readily solved exactly, and the solutions for the asymmetric game closely track that of the discrete game for large $N$. This will be useful for understanding our numerical solutions.

In the asymmetric game[1], each of the players *ante* an amount $A$, forming a *pot* $P = 2A$, and then take turns deciding whether or not to place an additional bet $B$. The players are betting on their private information—the value of a random number dealt to them after placing their antes, but before making the additional bets. The first player to act may either *check* (i.e. bet zero) or place a *bet* of fixed size $B > 0$. In the vN&M model, the second player only acts if this bet is placed, and then has the option to *fold* or *call*. If the second player folds, the first player receives the antes. If the fist player checks or the second player calls, the pot is distributed according to the highest hand. In their text, vN&M introduce what would later be called a behavior-strategy description of these games, where $b^i(x, a)$ describes the fraction of time player $i$ takes action $a$ at each information set. They find this game to have a continuum of optimal solutions (equivalent to NE, which had not yet been invented). The first player plays the same strategy in all of these, having two thresholds between which they never bet, and outside of which they always bet:

$$\begin{aligned} x_1 &= \frac{AB}{4A^2 + 5AB + B^2}, \\ x_2 &= \frac{2A^2 + 4AB + B^2}{4A^2 + 5AB + B^2}. \end{aligned}$$

The lower region corresponds to a *bluff.* In the modern recreational poker literature, betting one's weakest and strongest hands is referred to as betting a *polarized* range. Despite its simplicity, the model captures this significant insight into poker strategy. The second player has an infinite number of choices that achieve NE. If we let $b^2(y, \text{call})$

[1] We have adopted a more modern poker parlance, but the game is equivalent to the version described by von Neumann and Morgenstern with the "low bid" and "high bid" options.

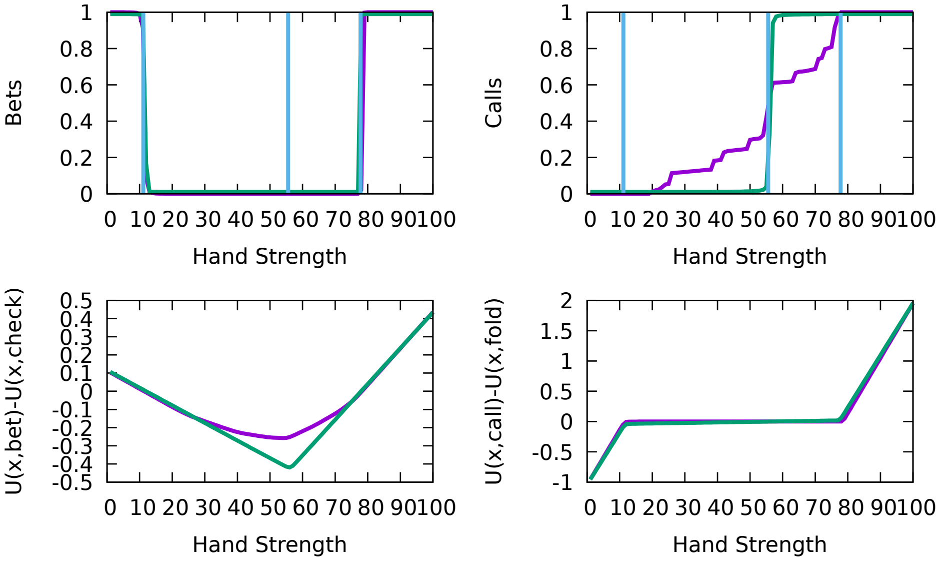

**Fig. 1** The discrete version of vN&M's asymmetric game with $P = 1, B = 1$ and $N = 100$ hands. The purple curves correspond to no perturbation, $\epsilon = 0$; the green curves are the perturbed game with $\epsilon = 0.01$. Top row: the fractions with which player 1 bets (left panel) and player 2 calls (right panel), along with the thresholds for the continuous game (blue vertical lines). Bottom row: the difference in the expected value of player 1's options (left panel) and player 2's options (right panel).

be the fraction of time the second player calls a bet when facing one, vN&M show that

$$\frac{1}{x_2 - z_0} \int_{z_0}^{x_2} b^2(y, \text{call}) dy \begin{cases} = \frac{A}{A+B} & \text{if } z_0 = x_1 \\ \geq \frac{A}{A+B} & \text{if } x_1 < z_0 < x_2 \end{cases}$$

are both necessary and sufficient conditions for equilibrium. Among these choices, there is a single, weakly-dominant strategy where the second player always folds/calls below/above a threshold

$$y_1 = \frac{3AB + 2B^2}{4A^2 + 5AB + B^2}.$$

Numerical solutions using any of the algorithms described above reveal an analogous result, where player 2's strategy endlessly drifts among weakly dominated strategies. For this reason, we consider both the perturbed game, where each option at a given information set is played with a minimum probability, $b(I, a) \geq \epsilon$, and the non-perturbed game. This is straightforward with BDFP, as we simply constrain the best decision:

$$d_\epsilon(I, a) = \epsilon + (1 - \epsilon)\delta_{a\tilde{a}}, \text{ where } \tilde{a} \in \arg\max_{a \in I} U(I, a; b_n). \tag{35}$$

A similar calculation can be made when computing best responses with XFP, while CFR requires a somewhat more complicated adjustment [15]. In Figure 1, we present results for the game with $N = 100$ uniformly distributed hands. These results are consistent with those shown in vN&M.

The graphs in the top row are the fractions with which player 1 bets (left panel) and player 2 calls (right panel). For player 1, the unperturbed result (shown in purple) and the perturbed result with $\epsilon = 0.01$ (shown in green) are nearly indistinguishable, indicative of there being a unique strategy for player 1 at equilibrium. For comparison, the thresholds given above for the continuous version of the game are shown as blue vertical lines. For player 2, the perturbed result is unique and approximates the solution with pure-strategy thresholds given above, while the unperturbed result endlessly drifts among the set of NE that employ a weakly-dominated strategy for player 2.

In the bottom row of Figure 1, we graph the difference in the expected value of player 1's options (left panel) and player 2's options (right panel), with a positive difference corresponding to the betting and calling options, respectively. From the left panel, we see that player 2's weakly dominated strategy is outperformed by the dominant one if player 1 is forced to bet with probability $\epsilon$ in the region where the unperturbed strategy is to check. Examining the lower-right panel, we see that there is no significant difference in player 2's perturbed and unperturbed payoff, as player 1 plays a nearly equal strategy in each case. We also see that the ambivalence in player 2's strategy is due to being indifferent between calling and folding throughout the region where mixed equilibria exist.

## 4.2 Expanded Neumann Poker

We will consider a more computationally demanding version of the asymmetric game that allows for a *bet* and a single *raise*, including the possibility of a *check-raise*. This game has been briefly addressed in a book aimed at recreational poker players, but the discussion omits the details given below [16].

For the continuous version of the game, a NE using pure-action choices containing 12 thresholds exists, but the strategy for player 1 is weakly dominated by an infinite number of other strategies, including pure-action strategies that feature two additional thresholds. This differs from what happens in the vN&M game, where there is a unique pure-action NE with the smallest number of thresholds and weakly dominant strategies. For the expanded game just described, the linear system of equations that determines the full set of thresholds is singular, leading to a degeneracy of pure action equilibria. The numerical results presented in the next section reveal that the discrete version of this game also features an infinite number of mixed strategy NE, with this occurring for both players.

When the pot $P = 2A = 1$, bet $B = 1$ and raise $R = 1$, the eight thresholds for player 1 are $\{x_1 = 64/1083, x_2 = 369/722, x_3 = 10/19, x_4 = x_5 - 32/1083, x_5, x_6 = 307/361, x_7 = x_8 - 22/361, x_8\}$, where $x_5$ and $x_8$ can be chosen arbitrarily so long as all of the thresholds remain in ascending order. These correspond to nine intervals of hands where player 1 takes a specific sequence of actions: {bet-fold < check-fold < check-raise < check-call < bet-fold < check-call < bet-call < check-raise < bet-call}. The six thresholds for the second player divide into two sets of three: $\{y_1^1 = 8/57, y_2^1 = 41/57, y_3^1 = 15/19\}$, corresponding to four intervals where player 2 responds to a check with the actions {bet-fold < check < bet-fold < bet-call} and $\{y_1^2 = 1/2, y_2^2 =$

$10/19, y_3^2 = 17/19\}$ where player 2 responds to a bet with the actions $\{\text{fold} < \text{raise} < \text{fold} < \text{call}\}$.

# 5 Numerical experiments

In this section we compare numerical solutions using all three of the algorithms discussed earlier. Our first set of experiments use an expanded form of Leduc poker [17], a model game that incorporates some additional aspects of real poker variants. We then explore numerical solutions of the expanded version of Neumann poker discussed in Section 4.2. These games can be considered with an arbitrary number of card ranks $N_r$ and allowed bets $N_b$ per betting round. We will indicate these as Leduc($N_r$,$N_b$) and Neumann($N_r$,$N_b$). Table 3 summarizes the essential differences between these and Kuhn poker. Below, we present results for Leduc(10,2) and Neumann(100,2), with the former having a deeper and the latter a wider game tree. These variants were chosen to have a similar computational complexity.

**Table 2** Characteristic features of the poker models.

| | replacement | duplicates | betting rounds |
|---|---|---|---|
| Neumann | Y | 0 | 1 |
| Kuhn | N | 0 | 1 |
| Leduc | N | 1 | 2 |

## 5.1 Leduc Poker

Standard Leduc, or Leduc(3,2), uses a deck of six cards—two copies for each of three ranks. Like Kuhn poker, Leduc poker is played without replacement. Players initially ante and are allowed a bet and up to one raise as in Neumann($N_r$,2), discussed in the previous section. If neither player folds, a card that is shared by both players is dealt, so that each player has a two-card hand. When comparing hands, a pair beats any non-paired hand. Unlike Kuhn and Neumann poker, there is a second betting round after the community card is dealt. Leduc(10,2) is played the same way, but with a deck consisting of two copies for each of ten ranks.

To make comparisons between the three algorithms discussed earlier, we examine plots of the utility of the current strategy pair $u^1(b_n^1, b_n^2)$ and the total exploitability $v(t) \geq 0$, defined in equation (31). In principle, the former should converge to the value of the game, but in practice there is some numerical error due to finite precision arithmetic. The total exploitability serves as a measure of this error.

In the top panel of Figure 2 we plot the expected value of the current strategy pair returned by the three algorithms at intervals of $1,000$ iterations. A sample for each of the three algorithms is shown for random initial data: BDFP in purple, CFR in blue and XFP in green. The algorithms all converge to a value around $-0.0165$ in favor of player 2. As mentioned earlier, the cost per iterations is similar for the three methods, with BDFP only slightly faster.

While getting the correct value of the game is desirable, a better measure of error is the total exploitability. In the middle panel, we plot this quantity at every $1,000$ iterations for the same samples presented in the top panel. The BDFP exploitability fluctuates somewhat. While the scale of the fluctuations tends to decrease as the computation proceeds, one can achieve significantly better results sooner by monitoring the exploitability. For example, in these calculations BDFP would have met a $v(t) \leq 10^{-3}$ stopping criteria at just $111,000$ iterations. If desired, one can avoid these fluctuations by including an additional averaging step, like that in CFR. We plot the simple running average in Figure 2, where we see it is nearly coincident with the XFP data in the middle panel. There is a small disadvantage to this approach in that it requires slightly more memory. Also, despite the monotone decrease in exploitability, it took 21,000 more iterations to meet the same stopping criteria. Finally, we note that the averaging is not necessary for convergence. As further evidence of this, the lower panel includes the unaveraged CFR data, which does not converge. On this scale the BDFP fluctuations are barely noticeable, as the exploitability of all three methods is already near zero after 1,000 iterations.

## 5.2 Neumann(100,2)

As with the vN&M's asymmetric game, the degeneracy of the Neumann(100,2) solutions means that the numerical solution will drift among the possible equilibria. This makes direct comparison of the strategy profiles generated by the methods difficult. Thus in our first figure in this section we present only a single realization generated by BDFP. The CFR and XFP algorithms give qualitatively similar results, as do other initial conditions. In the rest of the section we will be able to make direct comparisons.

In Figure 3 we plot the strategies returned by the BDFP algorithm as a function of the hand strength, $1 \leq i \leq 100$. In the top panel, the purple curve is the probability of checking, followed by folding to a bet; the green curve is the probability of checking, followed by calling a bet, and the light blue curve is the probability of checking, followed by raising, these three quantities adding to one. The remaining two strategy sequences, bet-call (dark blue) and bet-fold (gold) also add to one. The middle panel is player 2's response to an initial check from player 1: either another check (green), a bet followed by a fold if raised (purple), or a bet followed by a call if raised (light blue), these three quantities adding to one. The bottom panel is player 2's response to an initial bet from player 1: either fold (gold), call (red), or a raise (dark blue), with these three quantities again adding to one. The main thing to notice here is that there is a mixture of strategies being used for most hands, indicative of the type of degeneracy we saw in the asymmetric game, only this time it occurs for both players.

As with the asymmetric game, we find that we can again remove the weakly dominated strategies by approximating the equilibria subject to a small perturbation. The results for doing this with all three algorithms are shown in Figure 4, along with the unperturbed solution from Figure 2 and the thresholds for the continuous version of the game. The two arbitrary thresholds ($x_5$ and $x_8$) were roughly fit to these graphs, but the fit of all other thresholds provides a useful way of detecting coding errors. The mixing near the thresholds is due to boundary effects inherent to the discrete game and diminishes as the number of possible hands increases. Some of these regions are

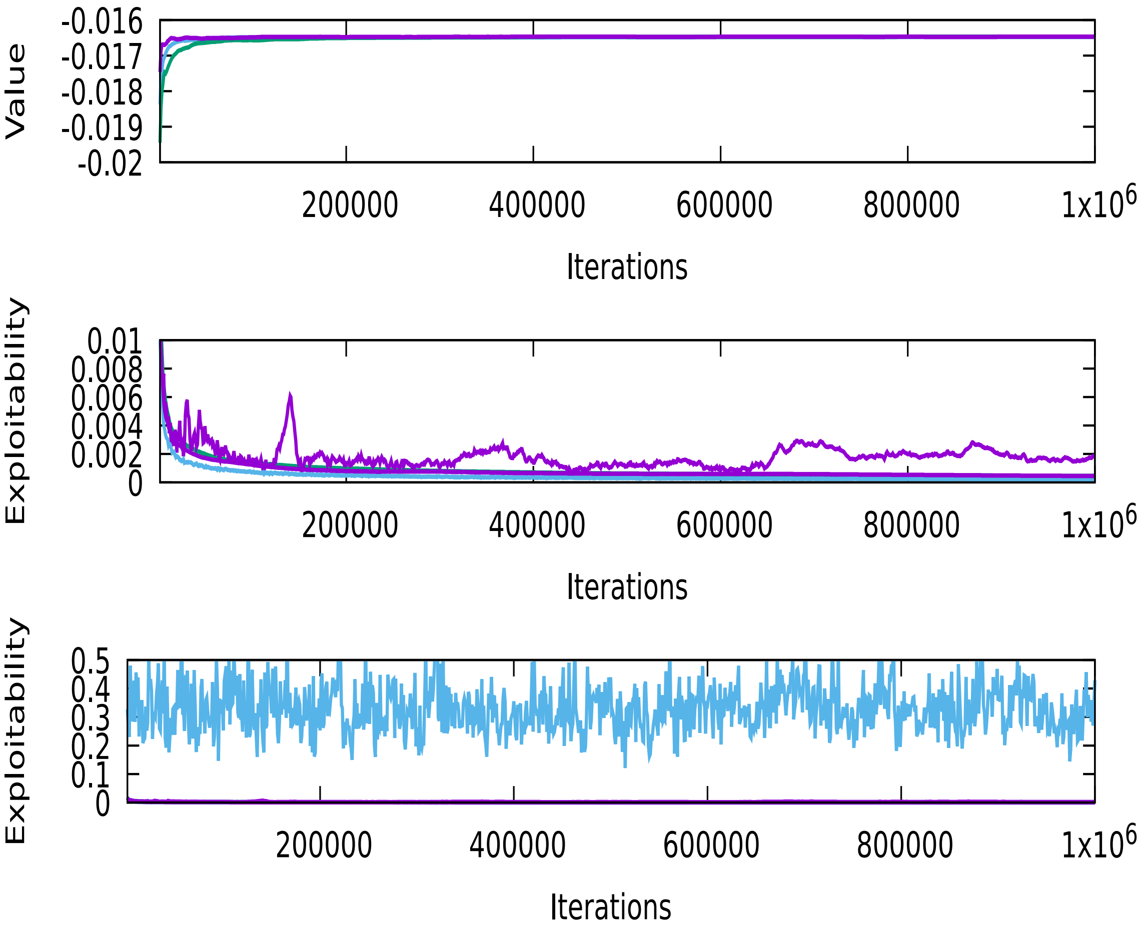

**Fig. 2** Numerical solution of Leduc(10,2) with $P = 1, B_1 = 1, B_2 = 2$. Top panel: the expected value of the current strategy pair (from player 1's perspective) returned at intervals of $1,000$ iterations. A sample for each algorithm is shown for random initial data: BDFP in purple, CFR in blue and XFP in green. Middle panel: the corresponding plots for total exploitability, along with the running average of the BDFP data in purple. Bottom panel: the same exploitability plots compared to the unaveraged CFR data in blue. On this scale, the exploitability curves for all three methods are barely distinguishable.

very narrow, so the discretization affects the result more strongly. The results for the three algorithms are nearly identical, as expected.

In the top panel of Figure 4 we plot the expected value of the current strategy pair (from player 1's perspective) returned by the three algorithms at intervals of $10,000$ iterations, along with the the exact value of $-\frac{44}{1083} \approx 0.406$ for the continuous version of the game, shown in gold. A sample for each of the three algorithms is shown for random initial data. We omitted the running average of the BDFP data, as the fluctuations were less significant in this case. Results vary with initial conditions, but

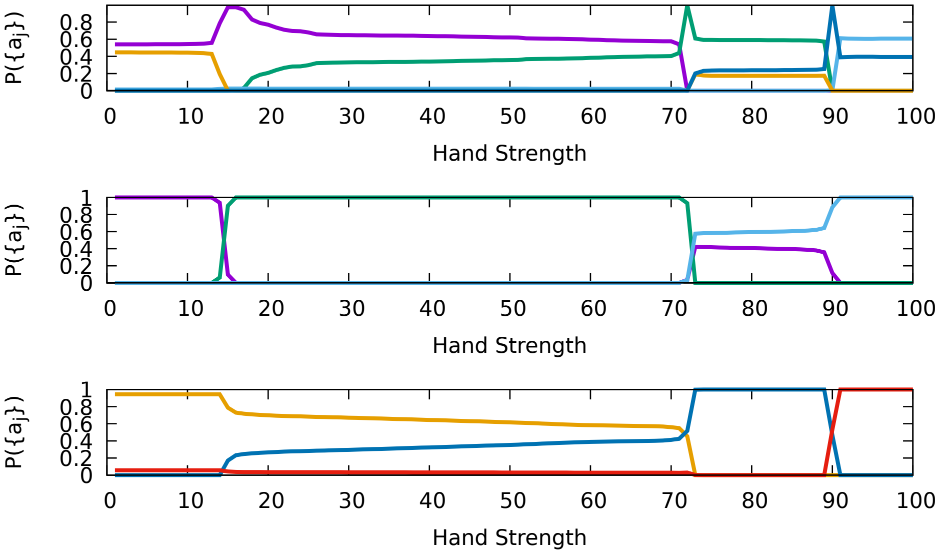

**Fig. 3** Numerical solution of the benchmark game with $P = 1, B = 1, R = 1$ and $N = 100$ hands. Top panel: the probability with which player 1 takes the action sequences check-fold (purple), check-call (green), check-raise (light blue), bet-call (dark blue), and bet-fold (gold). Middle panel: player 2's response to an initial check from player 1: either check (green), bet-fold (purple), or bet-call (light blue). Bottom panel: player 2's response to an initial bet from player 1: either fold (gold), call (dark blue), or raise (red).

are qualitatively similar. The numerical solutions will get closer to the solution for the continuous version of the game as $N \to \infty$, but round off error will prevent them from achieving this solution exactly. In the lower panel, we plot the total exploitability at every 10,000 iterations for the same samples presented in the top panel. The exploitability and the rate at which it decays is similar for all three methods.

In Figure 5, we compare the performance of all three algorithms for the generalized NE with $\epsilon = 0.01$. In the top panel, the curves are once again the utility of the current strategy pair from player 1's perspective returned by BDFP (purple), CFR (light blue) and XFP (green) at intervals of 10,000 iterations. The bottom panel is the exploitability using the same color scheme. Note that convergence for the perturbed game is much faster than for the unperturbed one, a result of the degenerate NE making convergence much more difficult to achieve.

## 6 Summary

In this work we have introduced a new algorithm, BDFP, that is realization equivalent to a generalized form of Fictitious Play, thus inheriting the convergence properties of that class of algorithms. We then compared the computational performance of BDFP to that of two additional algorithms, CFR and XFP, using both a well-known poker benchmark and an expanded version of a simple poker model first introduced by von Neumann and Morgenstern. We also presented an exact solution for the continuous

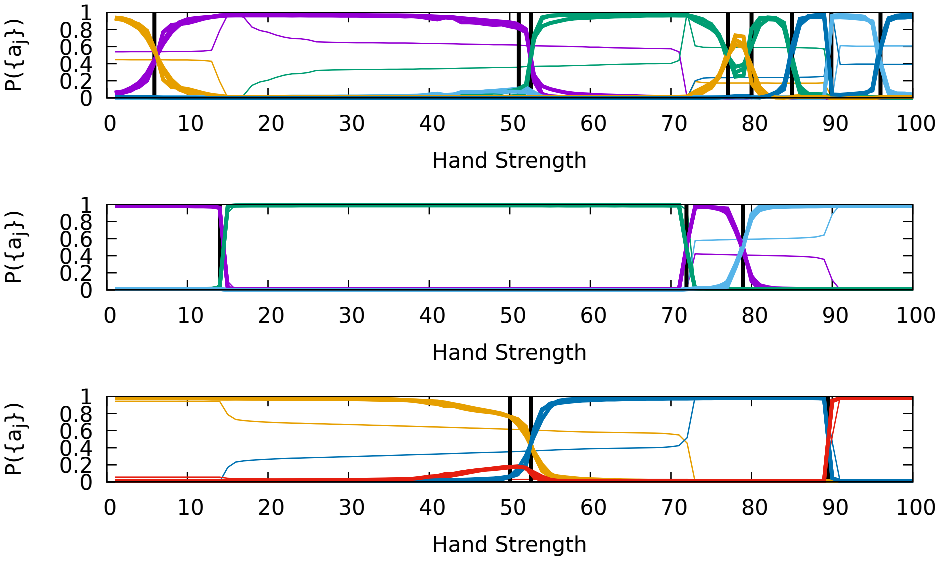

**Fig. 4** Probability with which Player 1 (top) and Player 2 (middle and bottom) play the action sequences described in the text as a function of hand strength for the numerical solution of perturbed ($\epsilon = 0.01$) Neumann(100,2) using all three algorithms. The solutions are nearly identical, so that these curves overlap. The color scheme is the same as that described in the last figure. We also plot the solution for the unperturbed ($\epsilon = 0.0$) game using BDFP (all blue, thin line) for $P = 1, B = 1, R = 1$, and $N = 100$, along with the thresholds (black vertical lines) for the continuous version of the game.

version of this game that is useful for testing the algorithms. Like vN&M's original game, this game features an infinite number of NE. As a result, a variation of the game with a perturbed strategy space and a unique equilibrium was also considered. This generalized NE is computed more quickly and is easier to interpret.

The computational cost per iteration is comparable for all three algorithms. BDFP's simple update formula relies on a best decision calculation that is intuitive and, at least in some approximate sense, routinely used to make decisions in recreational games. This makes it an ideal choice for anyone looking for a quick and easy game solving tool.

Finally, all three algorithms converged much more quickly when using updates that alternate between the two players, using the opponent's most recently updated strategy rather than the strategy from the previous iteration. This is a well-known feature of algorithms of this type, and is, for example, largely responsible for CFR+'s faster convergence [10, 18].

# References

[1] Zinkevich, M., Johanson, M., Bowling, M., Piccione, C.: Regret minimization in games wth incomplete information. Advances in Neural Information Processing Systems **20**, 906–912 (2008)

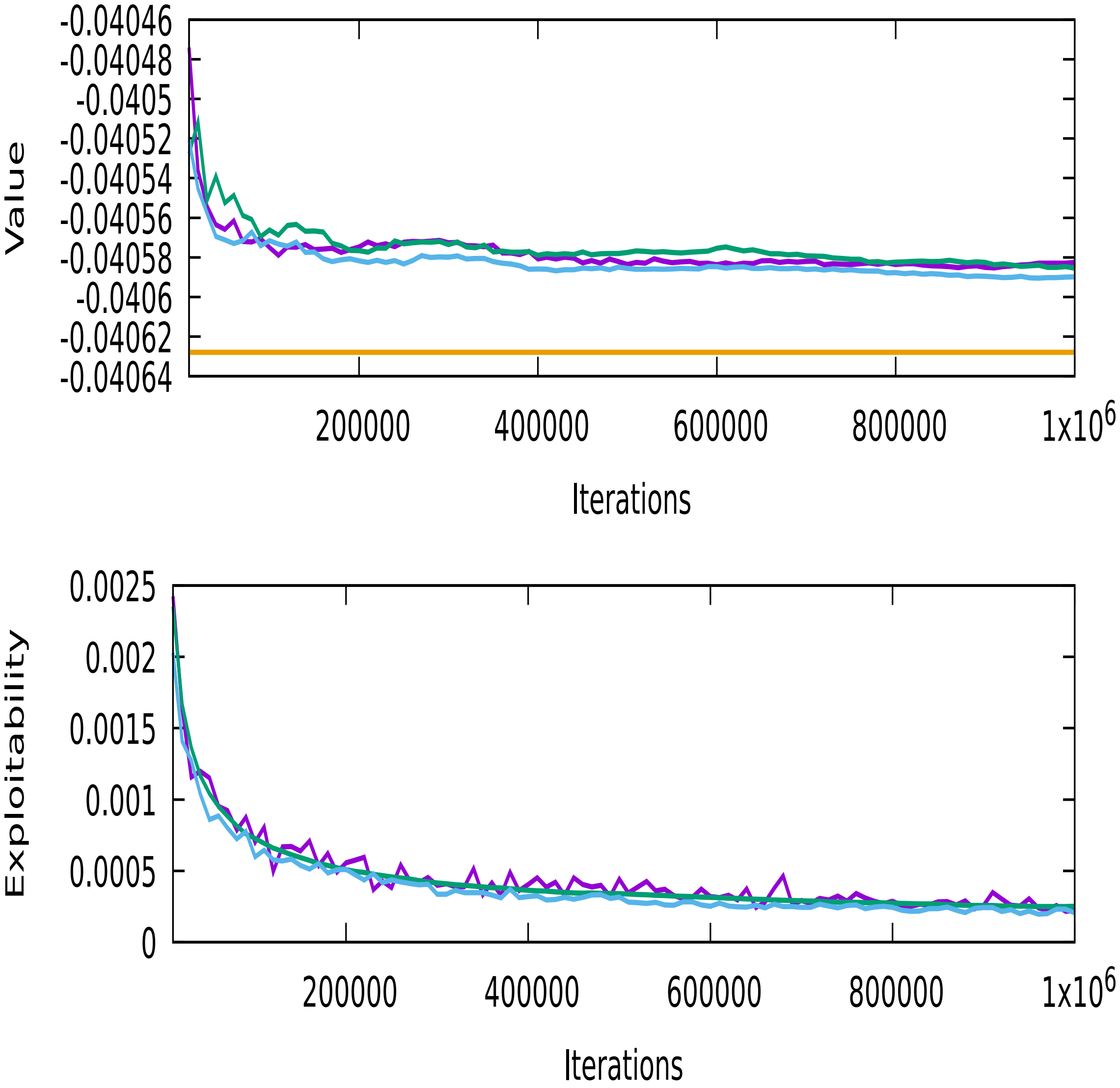

**Fig. 5** Numerical solution of Neumann(100,2) with $P = 1$, $B = 1$ and $R = 1$. Top panel: the expected value of the current strategy pair (from player 1's perspective) returned by the three algorithms at intervals of $10,000$ iterations, along with the exact value for the continuous version of the game, $-\frac{44}{1083} \approx -0.0406$, shown in gold. A sample for each of the of the algorithms is shown for random initial data: BDFP in purple, CFR in light blue and XFP in green. Bottom panel: the corresponding plots for total exploitability.

[2] Bowling, M., Burch, N., Johanson, M., Tammelin, O.: Heads-up limit hold'em poker is solved. Science **347**, 145–149 (2015)

[3] Brown, N., Sandholm, T.: Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. Science **359**, 418–424 (2018)

[4] Brown, N., Sandholm, T.: Superhuman ai for multiplayer poker. Science **365**, 885–890 (2019)

[5] Neuman, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, United States (1944)

[6] Brown, G.W.: Some Notes on Computation of Games Solutions. RAND Corporation, Santa Monica (1949)

[7] Robinson, J.: An iterative method of solving a game. Annals of Math. **54**, 296–301 (1951)

[8] Kuhn, H.W.: Extensive games and the problem of information. In: Contributions to the Theory of Games, Vol. II, Annals of Mathematical Studies No. 28 (1953)

[9] Maschler, M., Solan, E., Zamir, S.: Game Theory. Cambridge University Press, Cambridge (2013)

[10] Tammelin, O.: Solving large imperfect information games using cfr+. In: Corr (2014)

[11] Heinrich, J., Lanctot, M., Silver, D.: Fictitious self-play in extensive-form games. In: Proc. of the 32nd International Converence on Machine Learning (2015)

[12] Leslie, D.S., Collins, E.J.: Generalized weakened fictitious play. Games and Economic Behavior **56**, 285–298 (2006)

[13] Hofbauer, J., Sorin, S.: Best response dynamics for continuous zero-sum games,. Discrete and Continuous Dynamical Systems B **6**, 215–224 (2006)

[14] Kuhn, H.W.: Simplified to-person poker. In: Contributions to the Theory of Games, Vol. I, Annals of Mathematical Studies No. 24 (1950)

[15] Farina, G., Kroer, C., Sandholm, T.: Regret minimization in behaviorally-constrained zero-sum games. In: Proc. of the 34th International Conference on Machine Learning (2017)

[16] Chen, B., Ankenman, J.: The Mathematics of Poker. Conjelco, Pittsburgh (2006)

[17] Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., Rayner, C.: Bayes bluff: Opponent modelling in poker. In: Proceedings of the 24th International Joint Conference on Artificial Intellligence (2015)

[18] Burch, N., Moravcik, M., Schmid, M.: Revisiting cfr+ and alternating updates. J. of Artificial Intelligence Research **64**, 429–443 (2019)

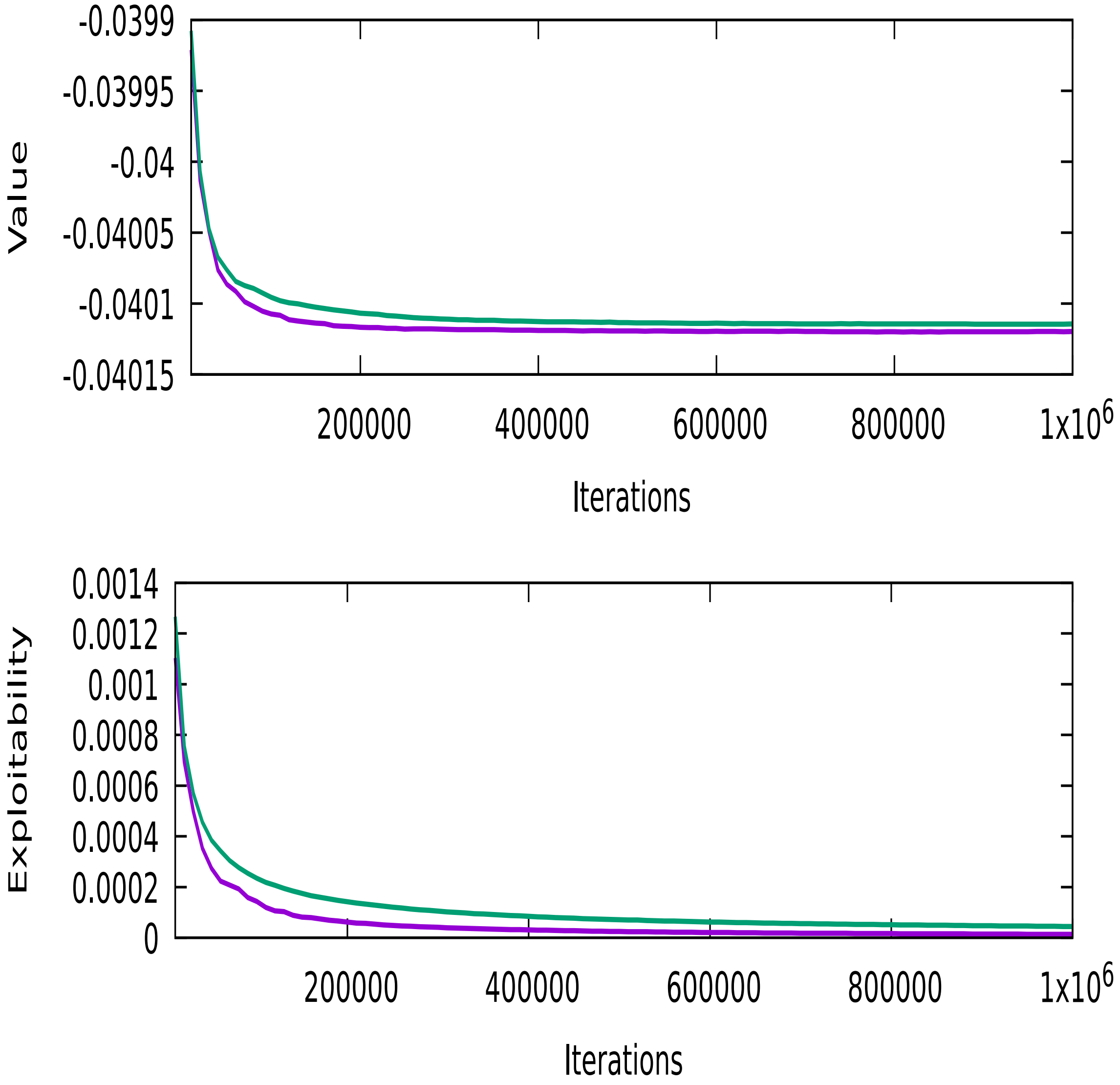

**Fig. 6** Perturbed ($\epsilon = 0.01$) Neumann(100,2) with $P = 1, B = 1$, and $R = 1$. Top panel: the expected value of the current strategy pair (from player 1's perspective) returned at intervals of $10,000$ iterations. A sample of each algorithm is shown for random initial data: BDFP in purple, CFR in light blue and XFP in green. Bottom panel: the corresponding plots for total exploitability.