Approximation Guarantees for Fictitious Play

Vincent Conitzer

***Abstract*— Fictitious play is a simple, well-known, and often-used algorithm for playing (and, especially, learning to play) games. However, in general it does not converge to equilibrium; even when it does, we may not be able to run it to convergence. Still, we may obtain an approximate equilibrium. In this paper, we study the approximation properties that fictitious play obtains when it is run for a limited number of rounds. We show that if both players randomize uniformly over their actions in the first r rounds of fictitious play, then the result is an ϵ-equilibrium, where $\epsilon = (r+1)/(2r)$. (Since we are examining only a constant number of pure strategies, we know that $\epsilon < 1/2$ is impossible, due to a result of Feder *et al.*) We show that this bound is tight in the worst case; however, with an experiment on random games, we illustrate that fictitious play usually obtains a much better approximation. We then consider the possibility that the players fail to choose the same r. We show how to obtain the optimal approximation guarantee when both the opponent's r and the game are adversarially chosen (but there is an upper bound R on the opponent's r), using a linear program formulation. We show that if the action played in the ith round of fictitious play is chosen with probability proportional to: 1 for $i = 1$ and $1/(i-1)$ for all $2 \leq i \leq R+1$, this gives an approximation guarantee of $1 - 1/(2 + \ln R)$. We also obtain a lower bound of $1 - 4/\ln R$. This provides an actionable prescription for how long to run fictitious play.**

I. Introduction

Computing a Nash equilibrium of a given normal-form game is one of the main computational problems in game theory; in multiagent systems, it is a problem that strategic agents often need to solve. Many related problems, such as deciding whether a Nash equilibrium that obtains a certain amount of welfare for the players exists, or whether a Nash equilibrium that places positive probability on a particular strategy exists, are NP-complete [18], [10]. The *Lemke-Howson* algorithm [21], perhaps the best-known algorithm for finding one Nash equilibrium, has been shown to require exponential time on some instances [31]. (Other, search-based algorithms obviously require exponential time on some instances, although they often do well in practice [27], [30].) The complexity of finding one Nash equilibrium remained open until very recently, when in a breakthrough series of papers [11], [7], [14], [8], it was finally shown that finding one Nash equilibrium is PPAD-complete, even in the two-player case. All of this is for the case where the normal form of the game is explicitly given; however, often the normal form is exponentially large, presenting additional difficulties.

A. Approximate Nash equilibrium

In light of this negative result, one approach is to try to find an *approximate* Nash equilibrium. In recent years, significant progress has been made in the computation of approximate Nash equilibria. It has been shown that for any ϵ, there is an ϵ-approximate equilibrium with support size $O((\log n)/\epsilon^2)$ [1], [22], so that we can find approximate equilibria by searching over all of these supports. More recently, Daskalakis *et al.* gave a very simple algorithm for finding a $1/2$-approximate Nash equilibrium [12]; we will discuss this algorithm shortly. Feder *et al.* then showed a lower bound on the size of the supports that must be considered to be guaranteed to find an approximate equilibrium [16]; in particular, supports of constant sizes can give at best a $1/2$-approximate Nash equilibrium. Daskalakis *et al.* then gave a polynomial-time algorithm for finding a .38-approximate equilibrium [13] (which uses arbitrarily large supports). Later results improved the approximation to .36391 [3] and then further to .3393 [36]. An overview of the work on computing approximate equilibria was recently presented at the 2008 Symposium on Algorithmic Game Theory [35].

While it is a desirable property for an algorithm to produce an approximate equilibrium, if the algorithm is to be used for determining how to play in an actual game, the approximation guarantee by itself is not likely to satisfy all of our demands. Rather, we would like the algorithm to have some additional desirable properties. It is not the aim of this paper to produce a list of exactly which properties are needed; however, let us illustrate some of the issues by looking at the following elegantly simple algorithm for computing an approximate equilibrium of a two-player game, due to Daskalakis *et al.* [12].[1]

- Label one of the players as player 1;
- Choose a pure strategy s for player 1;
- Find a best response t to s for player 2;
- Find a best response s' to t for player 1;
- Output the strategy profile in which player 1 plays each of s, s' with probability $1/2$, and player 2 plays t with probability 1.

Surprisingly, this extremely simple algorithm produces a 1/2-approximate equilibrium! An additional nice property of the algorithm is that it is easily applied to games whose normal form has exponential size: all that is needed is the ability to compute best responses. In spite of the algorithm's

V. Conitzer is with the Department of Computer Science, Duke Unversity, Durham, NC 27708, USA conitzer@cs.duke.edu

[1] Of course, we could have chosen one of the other recent algorithms for computing an approximate Nash equilibrium to consider in more detail; however, in this paper, we focus on the Daskalakis *et al.* algorithm because (1) like fictitious play, it is a very simple algorithm; (2) like fictitious play, it does not require the size of the supports to depend on the size of the game; and (3) the approximation guarantee of the other algorithms is not much better.

simplicity, however, it does not seem that it is likely to be used in practice, for a number of reasons.[2]

1) When the two players independently use the algorithm to find strategies for themselves, in general they need to be very well-coordinated for the resulting profile of strategies to constitute an approximate equilibrium. Specifically, they need to agree on who is player 1, which initial strategy s for player 1 is chosen, and which best response t to s is chosen (if there are multiple). If they fail to agree on one of these, then the result is not necessarily an approximate equilibrium. To some extent, this is a not a problem of the specific algorithm or of approximation in general, but one inherent in the definition of Nash equilibrium. In general, a game can have multiple equilibria, and if one player chooses her strategy from one equilibrium, and the other player from another, the result is not necessarily a Nash equilibrium.[3] Hence, coordination is an issue even in the presence of unlimited computation. Nevertheless, the use of an approximation algorithm such as the above greatly exacerbates the problem: it introduces many ways in which the players can fail to coordinate even if the game turns out to have only a single Nash equilibrium.
2) When the players *are* completely coordinated in their use of the algorithm, it is easy for a player to deviate from the profile and get a somewhat better utility for herself. Of course, to some extent, this is inherent in the use of approximate equilibria. Nevertheless, with an approximation algorithm such as the above, it is *extremely* easy to find a beneficial deviation. For example, for player 1, there seems to be no reason whatsoever for placing probability $1/2$ on s; she would generally be better off placing all her probability on s'.[4]
3) The algorithm is not guaranteed to produce an exact equilibrium (or even a very good approximation) in restricted settings where an exact equilibrium is easy to find, for example, (two-player) zero-sum games. Of course, it is trivial to extend the approximation algorithm to obtain this property: first check whether the game is (say) zero-sum; if it is, solve it exactly, otherwise use the approximation algorithm. Nevertheless, an algorithm that splits into cases like this seems less natural, and additionally it would not do well on games that are *very close to* zero-sum.
4) The algorithm seems intuitively somewhat unnatural, for example due to the asymmetry between the roles of the two players in the algorithm. (On the other hand, at least it is a simple algorithm; other algorithms proposed for computing exact or approximate equilibria are much more complicated. So, to the extent that we care about whether the algorithm could be used by a typical human being, perhaps a simple algorithm like this is preferable.)

It is not my aim to completely resolve all of these issues in this paper. Instead, we will study the approximation properties of a very well-known algorithm for playing (and learning to play) games, which arguably fares better on many of the issues discussed above. We show that this algorithm does obtain quite reasonable approximation guarantees.

B. Fictitious play

The algorithm we study is *fictitious play* [6]. Fictitious play is (these days) usually regarded as an algorithm for *learning* in games, that is, settings where a game is played repeatedly and where the play improves over time. Under this interpretation, fictitious play proceeds as follows: in the ith round of play, consider the opponent's historical distribution of play (that is, the fraction of times that each action was played in the first $i-1$ rounds), and play a best response to this distribution. This very simple algorithm has some nice properties: most notably, the players' historical distributions of play are guaranteed to converge to a Nash equilibrium under certain conditions—for example, when the game is zero-sum [28], or has generic payoffs and is 2×2 [24], or is solvable by iterated strict dominance [26], or is a weighted potential game [25]. (However, there are also games in which the distributions do not converge under fictitious play [32].) This algorithm can also be converted into an algorithm for playing a game just once: simply simulate what would happen in the repeated version of the game if both players were to use fictitious play, up to some predetermined round r; then output the historical distribution of play in this simulation as the strategy. This interpretation of fictitious play is not at all novel: in fact, when fictitious play was originally proposed, this is the interpretation that was given to it—hence the name *fictitious* play.

It is interesting to note that Daskalakis *et al.* [12] mention, as a future research direction, the possibility of adding additional iterations to their algorithm, that is, computing another best/better response and including it in the support. This is getting close to the idea of fictitious play.

There are many advantages to fictitious play, even when used as an algorithm for one-time play. For large enough r, it is guaranteed to get very close to exact equilibrium in the classes of games discussed above; it treats the players symmetrically; it is simple and natural; and it can be applied

[2]This is not meant as a criticism of this work: on the contrary, it establishes a key benchmark for an important criterion, and the simplicity of the algorithm is what allows us to easily discuss what other criteria we might need. It is also not meant to deny that studying the approximability of Nash equilibrium is an interesting question in its own right.

[3]In some restricted classes of games, the result is still guaranteed to be a Nash equilibrium: for example, in (two-player) zero-sum games.

[4]In fact, there is a more stringent definition of approximate Nash equilibrium that requires that every pure strategy in a player's support is an approximate best response [11]. The above algorithm will, in general, not produce such a *well-supported* approximate equilibrium. (For well-supported approximate equilibrium, it is known that an approximation of .658 can be obtained [20].) While this more demanding notion of approximate equilibrium addresses the issue to some extent, the problem runs deeper than that. For example, player 2's choice of strategy t in the above algorithm is certainly consistent with the notion of well-supported approximate equilibrium—but is it really reasonable to believe that player 2 would not think one step further and best-respond to the mixture over s and s'?

to games whose normal form has exponential size, as long as best responses can be computed. If the players use fictitious play independently (that is, each of them does her own simulation), there are still some coordination issues that we need to worry about, namely whether they choose the same action when fictitious play does not prescribe a single action. In this paper, we assume away this difficulty, that is, we assume that the players agree on which action to choose in these situations. There is also the issue of what happens when the players choose different final rounds r for their simulations. We will investigate this in more detail later in the paper.

Nevertheless, one of the most important uses of fictitious play is as an algorithm for learning in games. Our results are relevant in this case as well. For example, if we show that randomizing uniformly over the actions in the first r rounds of fictitious play results in an ϵ_r-equilibrium, and $\lim_{r\to\infty} \epsilon_r = \epsilon$, then we know that the historical distributions of play converge to the set of ϵ-equilibria of the game. In this paper, we will show $\epsilon_r = (r+1)/(2r)$, so that $\epsilon = 1/2$. That is, the historical distributions of play converge to the set of $1/2$-equilibria of the game.

C. The importance of fictitious play

Fictitious play is an old and simple algorithm. Many new, more complicated algorithms have been proposed since then, both for solving (or approximately solving) games (*e.g.*, [21], [27], [30], and the approximation algorithms discussed above), and for learning to play games (*e.g.*, [2], [5], [4], [9], [34], [29]). Thus, one may legitimately wonder whether fictitious play is still worthy of further study. I believe that it is, for at least the following reasons:

1) Fictitious play continues to be used in practice. For example, one recent paper [17] describes the use of fictitious play to solve a state-of-the-art poker game. An earlier paper [23] shows that generating an approximate equilibrium of Rhode Island Hold'em poker [33] can be done faster with fictitious play than with the barrier method of CPLEX (as was done in [19]). (There had been work on using fictitious play for poker even before that [15].) So, fictitious play remains practically relevant today, not just for learning but even as an algorithm for solving a game.
2) As for the approximation results for fictitious play that we derive in this paper, I am not aware of similar results for competing algorithms. There are numerous algorithms for computing an exact Nash equilibrium (*e.g.*, [21], [27], [30]), but these do not run in polynomial time [31]. (Some of these algorithms can also be used to find approximate Nash equilibria [30], but again, there is no satisfactory runtime bound.) There are algorithms whose main purpose is to find an approximate Nash equilibrium, such as the Daskalakis *et al.* algorithm mentioned earlier, but those algorithms do not have as many other (known) desirable properties as fictitious play. I am not aware of similar approximation results for other algorithms for learning in games. It is conceivable that good approximation results can be proven for some of the more modern algorithms (or that other good properties can be shown for some of the algorithms that were designed specifically for computing an approximate equilibrium). These are, in my opinion, worthwhile directions for future research, and hopefully this paper provides a good starting point. However, I believe that the results obtained in this paper, in combination with existing results about fictitious play, give fictitious play a rather unique portfolio of known desirable properties—especially because all that it requires is the ability to compute best responses.
3) Fictitious play is extremely simple. This makes it a very flexible algorithm, and one that is often implemented. It is also more believable as a model of human behavior than the more complicated algorithms. Finally, its simplicity makes it easy to analyze (which may be related to the point that similar results are not known for other algorithms).
4) Fictitious play is a natural approach for learning when the opponent is not (or may not be) adapting, playing a fixed strategy instead. This is especially useful in domains where the agent is not sure if she is actually facing an opponent, or rather just natural randomness.
5) There are multiple communities that are interested in solving games and/or learning how to play them, including the artificial intelligence/multiagent systems community, the theoretical computer science community, and the "traditional" game theory community. It appears that these communities are largely heading in different directions, and that they are not always aware of the work going on in the other communities (of course, there are exceptions). Hopefully, the work in this paper appeals to all of these communities: to the artificial intelligence/multiagent systems community because of the practical properties of fictitious play, to the theoretical computer science community because of the formal approximation guarantees, and to the traditional game theory community beause the algorithm studied is a standard one that seems reasonable for people to use. Ideally, this paper will help the communities to stay connected to each other.

II. Definitions

We will consider two-player normal-form games, in which player 1 (the row player) has pure strategy set S, and player 2 (the column player) has pure strategy set T. Each player $i \in \{1, 2\}$ has a utility function $u_i : S \times T \to \mathbb{R}$. None of the results in this paper depend on the size of the game; in fact, the game can have infinite size. All that is necessary is that a player can compute a best response to a given mixed strategy (with finite support) for the opponent.

Let s_i, t_i be the strategies that the players play in the ith round of fictitious play. We assume that when fictitious play is underdetermined (that is, when there are multiple best responses, or in the first round), it is common knowledge which strategy each player will choose. Because we place no

restriction on the size of the game, we can assume without loss of generality that all the s_i and t_i are distinct: for if (say) $s_i = s_j$ for $i < j$, we can create two copies of this strategy and consider one of them s_i, and the other s_j. Let σ_i be the uniform distribution over strategies $s_1, \ldots, s_i$, and let τ_i be the uniform distribution over strategies $t_1, \ldots, t_i$. Hence, for $i \geq 1$, s_{i+1} is a best response to τ_i, and t_{i+1} is a best response to σ_i.

For our purposes, it will generally suffice just to draw the payoff matrix for player 1, as follows:

$u_1(s_1, t_1)$	$u_1(s_1, t_2)$	$u_1(s_1, t_3)$	$\ldots$
$u_1(s_2, t_1)$	$u_1(s_2, t_2)$	$u_1(s_2, t_3)$	$\ldots$
$u_1(s_3, t_1)$	$u_1(s_3, t_2)$	$u_1(s_3, t_3)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

Since we assume (without loss of generality) that the rows and columns are sorted by their order of play in fictitious play (that is, s_i is the ith row and t_i is the ith column), not all matrices are valid: s_{i+1} must be a best response to τ_i, and t_{i+1} to σ_i.

When we consider fictitious play as an algorithm for one-time play, it requires an input parameter r, and it returns σ_r for the row player, and τ_r for the column player. We assume that all utilities lie in $[0, 1]$, which for finite games is without loss of generality because positive affine transformations of the utility functions do not affect the game strategically. As is commonly done [22], [12], [13], we study additive approximation. For a given mixed strategy τ, the mixed strategy σ is an *ϵ-best reponse* if for any $s \in S$, $u_1(\sigma, \tau) \geq u_1(s, \tau) - \epsilon$ (and similarly for player 2). (u_i is extended to the domain of mixed strategies simply by taking the expectation.) A pair of strategies σ, τ is an *ϵ-equilibrium* if they are ϵ-best responses to each other. It is known [16] that it is not possible to get better than a $1/2$-equilibrium unless mixed strategies with supports of size at least $\log n$ are considered, where n is the number of pure strategies per player. Because in this paper, we aim to obtain results that are independent of the size of the game, we effectively will only consider supports of constant size, and hence we cannot hope to get anything better than a $1/2$-equilibrium.

III. Computing approximate equilibria using fictitious play

In this section, we show that the pair of strategies defined by fictitious play (for a given r) is an ϵ_r-equilibrium, where ϵ_r converges (downward) to $1/2$ as r goes to infinity.

Theorem 1: The fictitious-play pair of strategies (σ_r, τ_r) is an ϵ_r-equilibrium, where $\epsilon_r = (r + 1)/(2r)$.

Proof: By symmetry, it suffices to show that σ_r is an ϵ_r-best response to τ_r. Let s^* be a best response to τ_r. The corresponding best-response utility for player 1 is $u_1(s^*, \tau_r) = \sum_{i=1}^{r} (1/r) u_1(s^*, t_i)$. For $2 \leq j \leq r + 1$, because s_j is a best response to τ_{j-1}, and all utilities are nonnegative, we have

$$u_1(s_j, \tau_r) = \sum_{i=1}^{r} (1/r) u_1(s_j, t_i) \geq$$

$$\sum_{i=1}^{j-1} (1/r) u_1(s_j, t_i) \geq \sum_{i=1}^{j-1} (1/r) u_1(s^*, t_i)$$

So, for the case where player 1 plays σ_r, we have

$$u_1(\sigma_r, \tau_r) = \sum_{j=1}^{r} (1/r) u_1(s_j, \tau_r) \geq$$

$$\sum_{j=1}^{r} (1/r) \sum_{i=1}^{j-1} (1/r) u_1(s^*, t_i) =$$

$$(1/r^2) \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} u_1(s^*, t_i) = (1/r^2) \sum_{i=1}^{r} (r - i) u_1(s^*, t_i)$$

On the other hand,

$$u_1(s^*, \tau_r) = \sum_{i=1}^{r} (1/r) u_1(s^*, t_i) = (1/r^2) \sum_{i=1}^{r} r u_1(s^*, t_i)$$

It follows that the suboptimality for player 1 of playing σ_r is

$$u_1(s^*, \tau_r) - u_1(\sigma_r, \tau_r) \leq (1/r^2) \sum_{i=1}^{r} i u_1(s^*, t_i) \leq$$

$$(1/r^2) \sum_{i=1}^{r} i = (1/r^2)(r + 1)(r/2) = (r + 1)/(2r)$$

■

To see that this bound is tight, consider the following game (only player 1's payoff matrix is shown; player 2 could, for example, have the transposed payoff matrix).

0	0	0	0	$\ldots$
1	0	0	0	$\ldots$
1	1	0	0	$\ldots$
1	1	1	0	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

In this game, following the fictitious play strategy σ_r against τ_r gives an expected utility of $\sum_{i=1}^{r} (1/r^2)(i - 1) = (1/r^2) r(r - 1)/2 = (r - 1)/(2r)$; on the other hand, playing any strategy s_j with $j > r$ gives an expected utility of 1. Hence, the suboptimality of the fictitious play strategy is $(r + 1)/(2r)$. We note that in this game, during fictitious play, there are always multiple best responses, and the player always makes, in some sense, the poorest choice among them. However, it is easy to modify the payoffs ever so slightly so that the best responses become unique (for example, by adding a tiny $\delta > 0$ to every $u_1(s_{i+1}, t_i)$, and then rescaling to $[0, 1]$).

IV. Experiment: Finding approximate equilibria in random games using fictitious play

The bound in Theorem 1 is a worst-case result; it seems reasonable to believe that in practice, fictitious play will perform much better. In fact, as we discussed previously, we know that in certain classes of games, fictitious play is guaranteed to converge to an exact equilibrium in the

limit [28], [24], [26], [25]. (Of course, we will not be able to run fictitious play for an infinite number of rounds, so even for these games we may not get an exact equilibrium.) Also, experimentally, fictitious play often converges to an exact equilibrium even if the game is not in one of those (known) classes. On the other hand, there are games, such as a game given by Shapley [32], on which fictitious play does not converge. In this section, we study how good an approximation fictitious play gives experimentally. We draw a random game (specifically, each payoff in the game is drawn independently and uniformly at random from the interval $[0, 1]$); then, we run fictitious play on it for a predetermined number of rounds; finally, we consider the resulting strategies of the players (corresponding to the history of play in fictitious play), and consider how close to equilibrium these strategies are—that is, we find the smallest ϵ for which these strategies are an ϵ-equilibrium. This ϵ is at most the approximation guarantee from Theorem 1, but in general it will be less. We repeat this over multiple games, and take the average. We compare to the algorithm by Daskalakis *et al.*, to support the assertion that fictitious play performs better in practice than that algorithm (which also sometimes outperforms its worst-case guarantee of $1/2$). (We compare to the Daskalakis *et al.* algorithm because it also has the property that the sizes of the mixed strategies' supports do not depend on the size of the game.) Figure 1 gives the results.

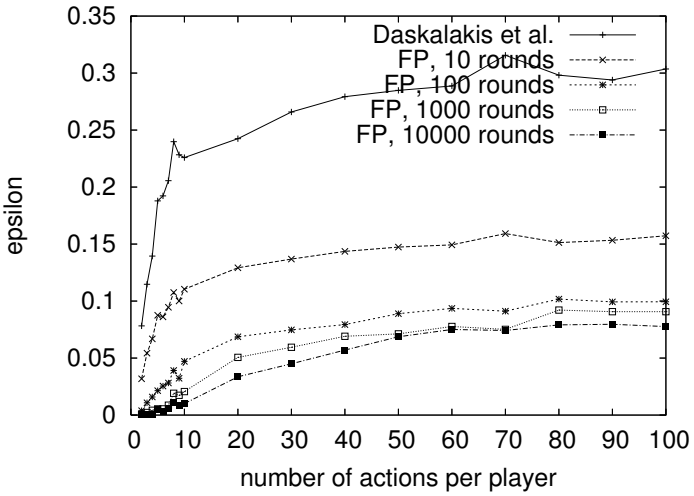

Fig. 1. As a function of the number of actions per player in a random two-player game, the average equilibrium approximation provided by the Daskalakis *et al.* algorithm, as well as by fictitious play with varying numbers of rounds. Each data point is averaged over 100 games.

Fictitious play significantly outperforms its worst-case guarantee of (slightly more than) $1/2$. The Daskalakis *et al.* algorithm also outperforms its $1/2$ bound, though not by as much. Initially, adding more rounds to fictitious play significantly improves the approximation, but eventually the improvement declines sharply.

V. Choosing the number of rounds when the opponent's number of rounds and the game are chosen adversarially

In the above, we implicitly assumed that the two players were coordinated in their choice of r—in a sense, they were thinking equally deeply about the game. Of course, there seems to be some incentive to try to out-think one's opponent. In this section, we consider what happens when the choice of r is not coordinated, that is, the players choose potentially different r_1, r_2. Specifically, we consider, from the perspective of player 1, the worst-case choice of r_2; and in addition, we continue to consider the worst-case game. To guard against the worst case, it makes sense for player 1 to randomize over her choice of r_1 (in effect, her choice of how deeply to think about the game). What is the optimal randomization? In this section, we suppose that some upper bound R on the opponent's choice of r_2 is known. (In the below, r refers to r_2.) The following four lemmas tell us how well individual strategies s_j do against a given r.

Lemma 1: Against τ_r, for $1 \leq j \leq r+1$, s_j is an $\epsilon_{j,r}$-best response, where $\epsilon_{j,r} = (r - j + 1)/r$.

Proof: The claim is trivial for $j = 1$, so let us consider the case where $j \geq 2$. Let s^* be a best response to τ_r. Because s_j is a best response to τ_{j-1}, and all utilities are nonnegative, we have $u_1(s_j, \tau_r) = \sum_{i=1}^{r}(1/r)u_1(s_j, t_i) \geq \sum_{i=1}^{j-1}(1/r)u_1(s_j, t_i) \geq \sum_{i=1}^{j-1}(1/r)u_1(s^*, t_i)$. Because $u_1(s^*, \tau_r) = \sum_{i=1}^{r}(1/r)u_1(s^*, t_i)$, it follows that $u_1(s^*, \tau_r) - u_1(s_j, \tau_r) \leq \sum_{i=j}^{r}(1/r)u_1(s^*, t_i) \leq (r - j + 1)/r$. ∎

Lemma 2: Against τ_r, for $r + 2 \leq j \leq 2r + 1$, s_j is an $\epsilon_{j,r}$-best response, where $\epsilon_{j,r} = (j - r - 1)/r$.

Proof: Let s^* be a best response to τ_r. Because s_j is a best response to τ_{j-1}, we have that $\sum_{i=1}^{j-1}(1/(j - 1))u_1(s_j, t_i) \geq \sum_{i=1}^{j-1}(1/(j-1))u_1(s^*, t_i)$. Because utilities lie in $[0, 1]$, we know that $\sum_{i=r+1}^{j-1}(1/(j - 1))u_1(s_j, t_i) - \sum_{i=r+1}^{j-1}(1/(j-1))u_1(s^*, t_i) \leq (j - r - 1)/(j - 1)$. Hence, $\sum_{i=1}^{r}(1/(j - 1))u_1(s^*, t_i) - \sum_{i=1}^{r}(1/(j - 1))u_1(s_j, t_i) \leq (j - r - 1)/(j - 1)$, or equivalently, $\sum_{i=1}^{r}(1/r)u_1(s^*, t_i) - \sum_{i=1}^{r}(1/r)u_1(s_j, t_i) \leq (j - r - 1)/r$. ∎

Lemma 3: Against τ_r, for $j \geq 2r + 2$, s_j is an ϵ-best response, where $\epsilon = 1$.

Proof: This follows immediately from the fact that utilities lie in $[0, 1]$. ∎

Lemma 4: For any given r, there exists a single game in which all of the bounds in Lemmas 1, 2, and 3 are (simultaneously) tight.

Proof: Consider a game in which player 1's utility function is given as follows:

- For $j \leq r+1$, $u_1(s_j, t_i) = 1$ if $j > i$, and $u_1(s_j, t_i) = 0$ otherwise.
- For $j > r + 1$, $u_1(s_j, t_i) = 1$ if $1 \leq j - i \leq r$, and $u_1(s_j, t_i) = 0$ otherwise.

Player 2's utilities can be arbitrary (except they need to be such that t_{i+1} is in fact a best response to σ_i). For example, the following is the worst-case game for $r = 3$ (player 1's utilities only).

0	0	0	0	0	0	...
1	0	0	0	0	0	...
1	1	0	0	0	0	...
1	1	1	0	0	0	...
0	1	1	1	0	0	...
0	0	1	1	1	0	...
0	0	0	1	1	1	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

(Again, there are multiple best responses in each case, but again they can be made unique using very small perturbations of the payoffs.)

Returning to the case of general r, in this game, we have:

- $u_1(s_{r+1}, \tau_r) = 1$.
- For $j \leq r+1$, $u_1(s_j, \tau_r) = (j-1)/r$, hence the suboptimality of s_j is $1-(j-1)/r = (r-j+1)/r$.
- For $r+2 \leq j \leq 2r+1$, $u_1(s_j, \tau_r) = 1-(j-r-1)/r$, hence the suboptimality of s_j is $(j-r-1)/r$.
- For $j \geq 2r+2$, $u_1(s_j, \tau_r) = 0$, hence the suboptimality of s_j is 1.

Hence, all of the bounds in Lemmas 1, 2, and 3 are (simultaneously) tight. ■

Now, player 1 must choose a probability distribution over her choice of r (let us again refer to player 1's r as r_1, and player 2's as r_2). It will be more convenient to think about the probability p_i with which player 1 plays s_i. We have $p_i = \sum_{j=i}^{\infty} P(r_1 = j)(1/j)$. We note that any nonincreasing sequence $p_1 \geq p_2 \geq p_3 \geq \ldots$ corresponds to a distribution over r_1. Hence, in the remainder, we will no longer refer to r_1, and we will simply refer to r_2 as r.

Suppose that there are p_i that guarantee player 1 a suboptimality of at most ϵ (for any opponent choice of r ($1 \leq r \leq R$) and any game). By Lemma 4, we must have, for any $1 \leq r \leq R$, $\sum_{j=1}^{r+1} p_j(r-j+1)/r + \sum_{j=r+2}^{2r+1} p_j(j-r-1)/r + \sum_{j=2r+2}^{\infty} p_j \leq \epsilon$. Conversely, by Lemmas 1, 2, and 3, we know that if we have, for any $1 \leq r \leq R$, $\sum_{j=1}^{r+1} p_j(r-j+1)/r + \sum_{j=r+2}^{2r+1} p_j(j-r-1)/r + \sum_{j=2r+2}^{\infty} p_j \leq \epsilon$, then these p_i guarantee player 1 a suboptimality of at most ϵ. Thus, to find the worst-case optimal p_i, we need to solve the following linear program:

minimize ϵ
subject to
$(\forall 1 \leq r \leq R)$ $\sum_{j=1}^{r+1} p_j(r-j+1)/r + \sum_{j=r+2}^{2r+1} p_j(j-r-1)/r + \sum_{j=2r+2}^{\infty} p_j - \epsilon \leq 0$
$(\forall j \geq 1)$ $p_j - p_{j+1} \geq 0$
$\sum_{j=1}^{\infty} p_j \geq 1$

One feasible solution is to set $p_1 = c_R$, $p_i = c_R/(i-1)$ for $2 \leq i \leq R+1$, and $p_i = 0$ everywhere else, where $c_R = 1/(1+\sum_{i=2}^{R+1} 1/(i-1)) = 1/(1+\sum_{i=1}^{R} 1/i) \geq 1/(2+\ln R)$. In this case, for any $r \leq R$ we have $\sum_{j=1}^{r+1} p_j(r-j+1)/r + \sum_{j=r+2}^{2r+1} p_j(j-r-1)/r + \sum_{j=2r+2}^{\infty} p_j \leq \sum_{j=1}^{r+1} p_j(r-j+1)/r + \sum_{j=r+2}^{\infty} p_j = 1 - \sum_{j=1}^{r+1} p_j(j-1)/r = 1 - \sum_{j=2}^{r+1} c_R/r = 1 - c_R$. So we obtain $\epsilon = 1 - c_R \leq 1 - 1/(2+\ln R)$. (The reader may be troubled by the fact that this assumes that player 1 is able to put positive probability on s_{R+1}, whereas it is assumed that player 2 will not put positive probability on t_{R+1}. If we do not allow player 1 to put positive probability on s_{R+1}, this makes only a vanishing difference as R becomes large; notationally, it is more convenient to allow for the probability on s_{R+1}.)

To see how close to optimal this solution is, let us first note that, because the p_j must be nonincreasing, we have, for $r+2 \leq j \leq 2r+1$, that $p_j(j-r-1)/r = p_j(1+(j-2r-1)/r) \geq p_j + p_{2r-j+2}(j-2r-1)/r = p_j + p_{2r-j+2}(-(2r-j+2)+1)/r)$. Hence, we find that $\sum_{j=1}^{r+1} p_j(r-j+1)/r + \sum_{j=r+2}^{2r+1} p_j(j-r-1)/r + \sum_{j=2r+2}^{\infty} p_j \geq \sum_{j=1}^{r+1} p_j(r-2j+2)/r + \sum_{j=r+2}^{\infty} p_j$. Thus, if we replace the main constraint in the above linear program by $\sum_{j=1}^{r+1} p_j(r-2j+2)/r + \sum_{j=r+2}^{\infty} p_j - \epsilon \leq 0$, it can only make the space of feasible solutions larger, and therefore the optimal solution can only improve. Similarly, if we drop the constraint that the p_j must be nonincreasing, the optimal solution can only improve. We thus obtain the linear program:

minimize ϵ
subject to
$(\forall 1 \leq r \leq R)$ $\sum_{j=1}^{r+1} p_j(r-2j+2)/r + \sum_{j=r+2}^{\infty} p_j - \epsilon \leq 0$
$\sum_{j=1}^{\infty} p_j \geq 1$

Now, if we take the dual of this linear program, we obtain:

maximize π
subject to
$(\forall 1 \leq j \leq \infty)$ $-\pi + \sum_{1 \leq r \leq j-2} q_r + \sum_{j-1 \leq r \leq R} q_r(r-2j+2)/r \geq 0$
$\sum_{r=1}^{R} q_r \leq 1$

Note that in the shorthand $j-1 \leq r \leq R$, r is not allowed to take the value 0. q_r can be interpreted as the weight that the adversary places on the example corresponding to r from Lemma 4.

One feasible solution to this dual linear program is to set $q_r = k_R/r$ for $1 \leq r \leq R$, where $k_R = 1/(\sum_{r=1}^{R} 1/r)$. We have $\sum_{1 \leq r \leq j-2} q_r + \sum_{j-1 \leq r \leq R} q_r(r-2j+2)/r = \sum_{r \geq 1} q_r + \sum_{j-1 \leq r \leq R} q_r(-2j+2)/r = 1 + \sum_{j-1 \leq r \leq R} q_r(-2j+2)/r = 1 + \sum_{j-1 \leq r \leq R} k_R(-2j+2)/r^2$. Now, $\sum_{j-1 \leq r \leq R} k_R(-2j+2)/r^2 \geq -2k_R + \sum_{j \leq r \leq R} k_R(-2j+2)/r^2 \geq -2k_R + \int_{r=j-1}^{\infty} k_R(-2(j-1))/r^2 dr = -4k_R$. So, we can set π to $1-4k_R \geq 1-4/\ln R$. This gives us a lower bound on the worst-case suboptimality of player 1 that nearly matches the upper bound given above.

The following theorem summarizes the above development:

Theorem 2: The optimal approximation guarantee that can be obtained using fictitious play against another player using fictitious play, who uses a worst-case $1 \leq r \leq R$ (where the first player only knows R), is $1-(1/\Theta(\log R))$. This guarantee can be achieved by playing s_i with probability proportional to:

- 1 for $i = 1$,

- $1/(i-1)$ for $2 \leq i \leq R+1$.

VI. Experiment: comparing the heuristic to the optimal LP solution

In the previous section, we advocated the heuristic of setting $p_1 = c_R$, $p_i = c_R/(i-1)$ for $2 \leq i \leq R+1$, and $p_i = 0$ everywhere else (where $c_R = 1/(1 + \sum_{i=1}^{R} 1/i)$). An alternative approach is to simply solve the original linear program to optimality using a linear program solver, and use those probabilities. In this section, we compare these two approaches by solving the linear program for values of R up to 1000.

In Figure 2, we compare, for different values of R, the ϵ corresponding to the optimal LP solution, the function $1 - 1/(2 + \ln R)$ which is an upper bound on the ϵ that our heuristic obtains, and the lower bound $1 - 4/\ln R$. As can be seen from the figure, the optimal LP solution does only slightly better than the heuristic, and the lower bound is somewhat loose.

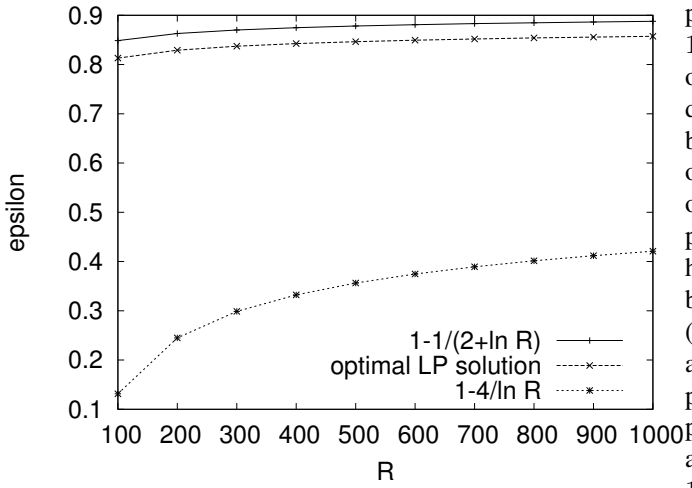

Fig. 2. As a function of R, the ϵ found by solving the original linear program to optimality, compared to the ϵ for our heuristic and the lower bound.

In Figure 3, we compare the probabilities placed on the pure strategies in the optimal LP solution to the probabilities in our heuristic. As can be seen, the probabilities are very close; the only difference is that the LP probabilities have more of a step-function nature.

Hence, there appears to be very little difference between using the heuristic and solving the LP to optimality. Because solving the LP becomes difficult for large R—using the GNU Linear Programming Kit (version 4.9), $R = 1000$ already takes 80 seconds to solve—it seems that using the heuristic makes more sense.

VII. Conclusions

In this paper, we studied fictitious play as an algorithm for generating approximately optimal strategies. We showed that if both players use the same r (that is, they each randomize uniformly over their actions in the first r rounds of fictitious play), then the result is an ϵ-equilibrium, where $\epsilon = (r+1)/(2r)$. (Since we are examining only a constant number of pure strategies, we know that $\epsilon < 1/2$ is impossible, due to a result of Feder *et al.* [16].) We showed that this bound is tight in the worst case; however, with an experiment on random games, we illustrated that fictitious play usually obtains a much better approximation. We then considered the possibility that the players choose different r. We showed how to obtain the optimal approximation guarantee when both the opponent's r and the game are adversarially chosen (but there is an upper bound R on the opponent's r), using a linear program formulation. Specifically, if the action played in the ith round of fictitious play is chosen with probability proportional to: 1 for $i = 1$ and $1/(i-1)$ for all $2 \leq i \leq R+1$, this gives an approximation guarantee of $1 - 1/(2 + \ln R)$. This provides an actionable prescription for how long to run fictitious play. We also obtained a lower bound of $1 - 4/\ln R$.

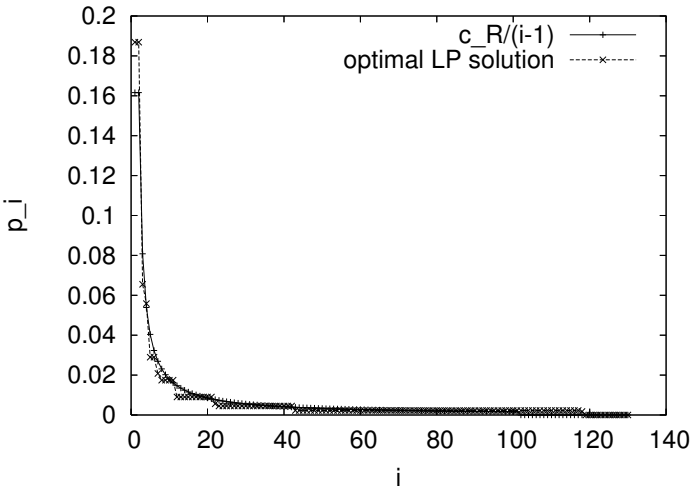

Fig. 3. For $R = 100$, the optimal p_i obtained by the linear program compared to the p_i from the heuristic.

While algorithms for computing approximate Nash equilibria that obtain slightly better approximation guarantees than fictitious play have already been found, it should be noted that fictitious play has many other desirable properties, and can also easily be used in games whose normal form has exponential size (as long as best responses can be computed). Moreover, fictitious play often performs well in practice, as illustrated by our experiment as well as by other results in the literature. Finally, unlike the other approximation algorithms, fictitious play is a natural algorithm for learning in games. It is encouraging that this often-used and relatively simple algorithm has reasonably good approximation properties.

Acknowledgments

This work is supported by NSF award number IIS-0812113 and a Research Fellowship from the Alfred P. Sloan Foundation.

REFERENCES

[1] I. Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199(1):339–355, 1994.

[2] B. Banerjee and J. Peng. $\mathrm{Rv}_{\sigma(t)}$: A unifying approach to performance and convergence in online multiagent learning. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 798–800, Hakodate, Japan, 2006.

[3] H. Bosse, J. Byrka, and E. Markakis. New algorithms for approximate Nash equilibria in bimatrix games. In *Workshop on Internet and Network Economics (WINE)*, pages 17–29, San Diego, CA, USA, 2007.

[4] M. Bowling. Convergence and no-regret in multiagent learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 209–216, Vancouver, Canada, 2005.

[5] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[6] G. W. Brown. Iterative solutions of games by fictitious play. In T. Koopmans, editor, *Activity Analysis of Production and Allocation*. New York: Wiley, 1951.

[7] X. Chen and X. Deng. 3-Nash is PPAD-complete. *Electronic Colloquium on Computational Complexity*, Report No. 134, 2005.

[8] X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–272, 2006.

[9] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, May 2007.

[10] V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008.

[11] C. Daskalakis, P. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 71–78, 2006.

[12] C. Daskalakis, A. Mehta, and C. H. Papadimitriou. A note on approximate Nash equilibria. In *Workshop on Internet and Network Economics (WINE)*, pages 297–306, Patras, Greece, 2006.

[13] C. Daskalakis, A. Mehta, and C. H. Papadimitriou. Progress in approximate Nash equilibria. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 355–358, San Diego, CA, USA, 2007.

[14] C. Daskalakis and C. H. Papadimitriou. Three-player games are hard. *Electronic Colloquium on Computational Complexity*, Report No. 139, 2005.

[15] W. Dudziak. Using fictitious play to find pseudo-optimal solutions for full-scale poker. In *Proceedings of the 2006 International Conference on Artificial Intelligence (ICAI06)*, pages 374–380, Las Vegas, Nevada, USA, 2006.

[16] T. Feder, H. Nazerzadeh, and A. Saberi. Approximating Nash equilibria using small-support strategies. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 352–354, San Diego, CA, USA, 2007.

[17] S. Ganzfried and T. Sandholm. Computing an approximate jam/fold equilibrium for 3-player no-limit Texas Hold'em tournaments. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 919–925, Estoril, Portugal, 2008.

[18] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[19] A. Gilpin and T. Sandholm. Optimal Rhode Island Hold'em poker. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1684–1685, Pittsburgh, PA, USA, 2005. Intelligent Systems Demonstration.

[20] S. C. Kontogiannis and P. G. Spirakis. Efficient algorithms for constant well supported approximate equilibria in bimatrix games. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP-07)*, pages 595–606, Wroclaw, Poland, 2007.

[21] C. Lemke and J. Howson. Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, 12:413–423, 1964.

[22] R. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 36–41, San Diego, CA, USA, 2003.

[23] H. B. McMahan and G. J. Gordon. A fast bundle-based anytime algorithm for poker and other convex games. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*, San Juan, Puerto Rico, 2007.

[24] K. Miyasawa. On the convergence of the learning process in a 2 x 2 nonzero sum two-person game. Technical report, Research memo 33, Princeton University, 1961.

[25] D. Monderer and L. S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68:258–265, 1996.

[26] J. Nachbar. Evolutionary selection dynamics in games: Convergence and limit properties. *International Journal of Game Theory*, 19:59–89, 1990.

[27] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008.

[28] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296–301, 1951.

[29] T. Sandholm. Perspectives on multiagent learning. *Artificial Intelligence*, 171(7):382–391, 2007.

[30] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 495–501, Pittsburgh, PA, USA, 2005.

[31] R. Savani and B. von Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74:397–429, 2006.

[32] L. S. Shapley. Some topics in two-person games. In M. Drescher, L. S. Shapley, and A. W. Tucker, editors, *Advances in Game Theory*. Princeton University Press, 1964.

[33] J. Shi and M. Littman. Abstraction methods for game theoretic poker. In *Computers and Games*, pages 333–345. Springer-Verlag, 2001.

[34] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.

[35] P. G. Spirakis. Approximate equilibria for strategic two person games. In *Proceedings of the First Symposium on Algorithmic Game Theory (SAGT-08)*, pages 5–21, Paderborn, Germany, 2008.

[36] H. Tsaknakis and P. G. Spirakis. An optimization approach for approximate Nash equilibria. In *Workshop on Internet and Network Economics (WINE)*, pages 42–56, San Diego, CA, USA, 2007.