Mach Learn (2007) 67:23–43
DOI 10.1007/s10994-006-0143-1

AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents

Vincent Conitzer · Tuomas Sandholm

Received: 8 September 2005 / Revised: 16 March 2006 / Accepted: 21 June 2006/
Published online: 18 September 2006

Abstract Two *minimal* requirements for a satisfactory multiagent learning algorithm are that it 1. learns to play optimally against stationary opponents and 2. converges to a Nash equilibrium in self-play. The previous algorithm that has come closest, WoLF-IGA, has been proven to have these two properties in 2-player 2-action (repeated) games—assuming that the opponent's mixed strategy is observable. Another algorithm, ReDVaLeR (which was introduced after the algorithm described in this paper), achieves the two properties in games with arbitrary numbers of actions and players, but still requires that the opponents' mixed strategies are observable. In this paper we present AWESOME, the first algorithm that is guaranteed to have the two properties in games with arbitrary numbers of actions and players. It is still the only algorithm that does so while only relying on observing the other players' actual actions (not their mixed strategies). It also learns to play optimally against opponents that *eventually become* stationary. The basic idea behind AWESOME (*Adapt When Everybody is Stationary, Otherwise Move to Equilibrium*) is to try to adapt to the others' strategies when they appear stationary, but otherwise to retreat to a precomputed equilibrium strategy. We provide experimental results that suggest that AWESOME converges fast in practice. The techniques used to prove the properties of AWESOME are fundamentally different from those used for previous algorithms, and may help in analyzing future multiagent learning algorithms as well.

Keywords Game theory · Learning in games · Nash equilibrium

Editors: Amy Greenwald and Michael Littman

V. Conitzer (✉) · T. Sandholm
Carnegie Mellon University, Computer Science Department, Pittsburgh, PA 15213
e-mail: conitzer@cs.cmu.edu

T. Sandholm
e-mail: sandholm@cs.cmu.edu

1 Introduction

Learning from experience is a key capability in AI, because it can be difficult to program a system in advance to act appropriately. Learning is especially important in multiagent settings where the other agents' behavior is not known in advance. Multiagent learning (learning in games) is complicated by the fact that the other agents may be learning as well, thus making the environment nonstationary for a learner.

Multiagent learning has been studied with different objectives as well as with different restrictions on the game and on what the learner can observe (e.g., Tan, 1993; Littman, 1994; Sandholm & Crites, 1996; Sen & Weiss, 1998). Two *minimal* desirable properties of a good multiagent learning algorithm are

- Learning to play optimally against stationary opponents (or even opponents that eventually become stationary).[1]
- Convergence to a Nash equilibrium in self-play (that is, when all the agents use the same learning algorithm).

These desiderata are *minimal* in the sense that any multiagent learning algorithm that fails at least one of these properties is, in a sense, unsatisfactory. Of course, one might also want the algorithm to have additional properties.[2] We discuss alternative objectives for learning in games in Section 7.

The WoLF-IGA (Bowling & Veloso, 2002) algorithm (an improvement over an earlier algorithm (Singh, Kearns, & Mansour, 2000)) constituted a significant step forward in this line of research. It is guaranteed to have both of the properties in general-sum (repeated) games under the following assumptions:

- (a) there are at most 2 players,
- (b) each player has at most 2 actions to choose from,
- (c) the opponent's mixed strategy (distribution over actions) is observable, and
- (d) gradient ascent of infinitesimally small step sizes can be used.[3]

Another algorithm, ReDVaLeR, was proposed more recently (Banerjee & Peng, 2004) (after the introduction of the AWESOME algorithm, described in this paper, at the International Conference on Machine Learning, 2003). ReDVaLeR achieves the two properties in general-sum games with arbitrary numbers of actions and opponents, but still requires assumptions (c) and (d). In addition, for a different setting of a parameter of the algorithm, ReDVaLeR achieves constant-bounded regret. An interesting aspect of this algorithm is that it explicitly checks whether the opponents' strategies are stationary or not, and proceeds differently depending on the result of this check. This serves to demonstrate just how powerful assumption (c) really is, in that it allows one to achieve the two properties separately: if the result of the check is positive, one can focus on converging to a best response, and if it is negative, one can focus on

[1] This property has sometimes been called *rationality* (Bowling & Veloso, 2002), but we avoid that term because it has an established, different meaning in economics.

[2] It can be argued that the two properties are not even strong enough to constitute a "minimal" set of requirements, in the sense that we would still not necessarily be satisfied with an algorithm if it has these properties. However, we would likely not be satisfied with any algorithm that did *not* meet these two requirements, even if it had other properties. This is the sense in which we use the word "minimal".

[3] Bowling and Veloso also defined a more generally applicable algorithm based on the same idea, but only gave experimental justification for it.

converging to an equilibrium. Without assumption (c), this approach is not possible, because the opponents' empirical distributions of play will change over time even if the opponents' actual mixed strategies are stationary.

In this paper we present AWESOME—the first algorithm that achieves both of the properties in general repeated games, and still the only algorithm that achieves them without any of the assumptions (a), (b), (c), and (d). As per the above observations, especially the fact that (c) is not required is significant. In fact, the sole purpose of many of the techniques in this paper is precisely to avoid assuming (c). It has the two properties with any finite number of agents and any finite number of actions; it only requires being able to observe other players' actions (rather than the distribution that the actions are drawn from); and it does not rely on infinitesimal updates. As in WoLF-IGA and ReDVaLeR, our notion of convergence is that the *stage-game* strategy converges to the desired strategy (not just the long-term empirical distribution of play).

AWESOME still makes some of the same assumptions that were made in the other theoretical work attempting to attain both of the properties (Singh, Kearns, & Mansour, 2000; Bowling & Veloso, 2002; Banerjee & Peng, 2004). First, it only deals with repeated games—that is, stochastic games with a single state. Second, it assumes that the structure of the game is known (has already been learned). This assumption is made in much (though not all) of the game theory literature on learning (for a review, see (Fudenberg & Levine, 1998)), but a significant amount of other research in multiagent learning in computer science does attempt to have the agents learn the game as well (Littman, 1994; Littman & Szepesvári, 1996; Hu & Wellman, 1998; Claus & Boutilier, 1998; Brafman & Tennenholtz, 2000; Banerjee, Sen, & Peng, 2001; Littman, 2001; Pivazyan & Shoham, 2002; Wang & Sandholm, 2002; Brafman & Tennenholtz, 2003; Greenwald & Hall, 2003; Conitzer & Sandholm, 2003a; Wang & Sandholm, 2003; Conitzer & Sandholm, 2004). However, so far this research has not been able to make claims of the kind made in this paper. In any case, a recent result shows that (for continuous-time dynamics) some knowledge of the other players' payoffs is necessary to converge to Nash equilibrium (Hart & Mas-Colell, 2003). If the game is not known initially, but the agents can observe the realized payoffs of all agents, then, given that all the agents are using the same learning algorithm, they could conceivably collaboratively explore the game and learn the game structure, and then learn how to play. The third assumption is that the agents can compute a Nash equilibrium.[4] It is still unknown whether a Nash equilibrium can be found in worst-case polynomial time (Papadimitriou, 2001), but it is known that certain related questions are hard in the worst case (Gilboa & Zemel, 1989; Conitzer & Sandholm, 2003b). However, in practice Nash equilibria can be found for reasonably large games, using a variety of algorithms (Lemke & Howson, 1964; Porter, Nudelman, & Shoham, 2004; Sandholm, Gilpin, & Conitzer, 2005).[5]

The basic idea behind AWESOME (*Adapt When Everybody is Stationary, Otherwise Move to Equilibrium*) is to try to adapt to the other agents' strategies when they appear stationary, but otherwise to retreat to a precomputed equilibrium strategy. At any point in time, AWESOME maintains either of two null hypotheses: that the others are playing the precomputed equilibrium, or that the others are stationary. Whenever both of these hypotheses are rejected,

[4] We assume that when there are multiple AWESOME players, they compute the same Nash equilibrium. This is natural since they share the same algorithm.

[5] Some of the literature on learning in games has been concerned with reaching the equilibrium through some simple dynamics (not using a separate algorithm to compute it). This is certainly a worthwhile objective in our opinion. However, in this paper the focus is on learning to play appropriately with respect to the opponent's algorithm.

AWESOME restarts completely. AWESOME may reject either of these hypotheses based on actions played in an *epoch*. Over time, the epoch length is carefully increased and the criterion for hypothesis rejection tightened to obtain the convergence guarantee. The AWESOME algorithm is also self-aware: when it detects that its own actions signal nonstationarity to the others, it restarts itself for synchronization purposes.

The techniques used in proving the properties of AWESOME are fundamentally different from those used for previous algorithms, because the requirement that the opponents' mixed strategies can be observed is dropped. These techniques may also be valuable in the analysis of other learning algorithms in games.

It is important to emphasize that, when attempting to converge to an equilibrium, as is common in the literature, our goal is to eventually learn the equilibrium of the *one-shot* game, which, when played repeatedly, will also constitute an equilibrium of the repeated game. The advantage of such equilibria is that they are natural and simple, always exist, and are robust to changes in the discounting/averaging schemes. Nevertheless, in repeated games it is possible to also have equilibria that are fundamentally different from repetitions of the one-shot equilibrium; such equilibria rely on a player conditioning its future behavior on the opponents' current behavior. Interestingly, a recent paper shows that when players try to maximize the limit of their average payoffs, such equilibria can be constructed in worst-case polynomial time (Littman & Stone, 2003).

The rest of the paper is organized as follows. In Section 2, we define the setting. In Section 3, we motivate and define the AWESOME algorithm and show how to set its parameters soundly. In Section 4, we show that AWESOME converges to a best response against opponents that (eventually) play stationary strategies. In Section 5, we show that AWESOME converges to a Nash equilibrium in self-play. In Section 6, we experimentally compare AWESOME to fictitious play. In Section 7, we discuss alternative objectives for learning in games (and, in the process, we also discuss a large body of related research). In Sections 8 and 9, we present conclusions and directions for future research.

2 Model and definitions

We study multiagent learning in a setting where a fixed finite number of agents play the same finite stage game repeatedly. We first define the stage game and then the repeated game.

2.1 The stage game

Definition 1 (Stage game). A *stage game* is defined by a finite set of agents $\{1, 2, \ldots, n\}$, and for each agent i, a finite action set A_i, and a utility function $u_i : A_1 \times A_2 \times \cdots \times A_n \to \mathbb{R}$. The agents choose their actions independently and concurrently.

We now define strategies for a stage game.

Definition 2 (Strategy). A *strategy* for agent i (in the stage game) is a probability distribution π_i over its action set A_i, indicating what the probability is that the agent will play each action. In a *pure strategy*, all the probability mass is on one action. Strategies that are not pure are called *mixed strategies*.

The agents' strategies are said to be in equilibrium if no agent is motivated to unilaterally change its strategy given the others' strategies:

Definition 3 (Nash equilibrium). A strategy profile $(\pi_1^*, \pi_2^*, \ldots, \pi_n^*)$ is a *Nash equilibrium* (of the stage game) if, for every agent i and for every strategy π_i,

$$E_{(\pi_1^*,\ldots,\pi_{i-1}^*,\pi_i^*,\pi_{i+1}^*,\pi_2^*,\ldots,\pi_n^*)} u_i(a_1, a_2, \ldots, a_n) \geq E_{(\pi_1^*,\ldots,\pi_{i-1}^*,\pi_i,\pi_{i+1}^*,\pi_2^*,\ldots,\pi_n^*)} u_i(a_1, a_2, \ldots, a_n)$$

We call a Nash equilibrium a *pure-strategy Nash equilibrium* if all the individuals' strategies in it are pure. Otherwise, we call it a *mixed-strategy Nash equilibrium*.

2.2 The repeated game

The agents play the stage game repeatedly (forever). As usual, we assume that the agents observe each others' actions. An agent may learn from previous rounds, so its strategy in a stage game may depend on how the earlier stage games have been played.

In the next section we present our learning algorithm for this setting, which has the desirable properties that it learns a best-response strategy against opponents that (eventually) are stationary, and it converges to a Nash equilibrium in self-play.

3 The AWESOME algorithm

In this section we present the AWESOME algorithm. We first give the high-level idea, and discuss some additional specifications and their motivation. We then give the actual algorithm and the space of valid parameter vectors for it.

3.1 The high-level idea

Roughly, the idea of the algorithm is the following. When the others appear to be playing stationary strategies, AWESOME adapts to play the best response to those apparent strategies. When the others appear to be adapting their strategies, AWESOME retreats to an equilibrium strategy. (Hence, AWESOME stands for *Adapt When Everybody is Stationary, Otherwise Move to Equilibrium*.)

3.2 Additional specifications

While the basic idea is simple, we need a few more technical specifications to enable us to prove the desired properties.

- To make the algorithm well-specified, we need to specify which equilibrium strategy AWESOME retreats to. We let AWESOME compute an equilibrium in the beginning, and it will retreat to its strategy in that equilibrium every time it retreats. To obtain our guarantee of convergence in self-play, we also specify that each AWESOME agent computes the

same equilibrium.[6] We observe that *any* equilibrium will work here (e.g., a social welfare maximizing one), but AWESOME might not converge to *that* equilibrium in self-play—that is, it may converge to another equilibrium.

- When retreating to the equilibrium strategy, AWESOME forgets everything it has learned. So, retreating to an equilibrium is a complete restart. (This may be wasteful in practice, but makes the analysis easier.)
- Best-responding to strategies that are close to the precomputed equilibrium strategies, but slightly different, can lead to rapid divergence from the equilibrium. To avoid this, AWESOME at various stages has a null hypothesis that the others are playing the precomputed equilibrium. AWESOME will not reject this hypothesis unless presented with significant evidence to the contrary.
- AWESOME rejects the equilibrium hypothesis also when its own actions, chosen according to its mixed equilibrium strategy, happen to appear to indicate a nonequilibrium strategy (even though the underlying mixed strategy is actually the equilibrium strategy). This will help in proving convergence in self-play by making the learning process synchronized across all AWESOME players. (Since the other AWESOME players will restart when they detect such nonstationarity, this agent restarts itself to stay synchronized with the others.)
- After AWESOME rejects the equilibrium hypothesis, it randomly picks an action and changes its strategy to always playing this action. At the end of an epoch, if another action would perform *significantly* better than this action against the strategies the others appeared to play in the last epoch, it switches to this action. (The significant difference is necessary to prevent the AWESOME player from switching back and forth between multiple best responses to the actual strategies played.)
- Because the others' strategies are unobservable (only their actions are observable), we need to specify how an AWESOME agent can reject, based on others' actions, the hypothesis that the others are playing the precomputed equilibrium strategies. Furthermore, we need to specify how an AWESOME agent can reject, based on others' actions, the hypothesis that the others are drawing their actions according to stationary (mixed) strategies. We present these specifications in the next subsection.

3.3 Verifying whether others are playing the precomputed equilibrium and detecting nonstationarity

We now discuss the problem of how to reject, based on observing the others' actions, the hypothesis that the others are playing according to the precomputed equilibrium strategies. AWESOME proceeds in epochs: at the end of each epoch, for each agent i in turn (including itself), it compares the actual distribution, h_i, of the actions that i played in the epoch (i.e. what percentage of the time each action was played) against the (mixed) strategy π_i^* from the precomputed equilibrium. AWESOME concludes that the actions are drawn from the equilibrium strategy if and only if the distance between the two distributions is small: $\max_{a_i \in A_i} |p_{h_i}^{a_i} - p_{\pi_i^*}^{a_i}| < \epsilon_e$, where p_ϕ^a is the percentage of time that action a is played in ϕ.

When detecting whether or not an agent is playing a stationary (potentially mixed) strategy, AWESOME uses the same idea, except that in the closeness measure, in place of π_i^* it uses

[6] This is at least somewhat reasonable since they share the same algorithm. If there are only finitely many equilibria, then one way to circumvent this assumption is to let each agent choose a random equilibrium after each restart, so that there is at least some probability that the computed equilibria coincide.

the actual distribution, h_i^{prev}, of actions played in the epoch just preceding the epoch that just ended. Also, a different threshold may be used: ϵ_s in place of ϵ_e. So, AWESOME maintains the stationarity hypothesis if and only $\max_{a_i \in A_i} |p_{h_i}^{a_i} - p_{h_i^{prev}}^{a_i}| < \epsilon_s$.

The naïve implementation of this keeps the number of iterations N in each epoch constant, as well as ϵ_e and ϵ_s. Two problems are associated with this naïve approach. First, even if the actions are actually drawn from the equilibrium distribution (or a stationary distribution when we are trying to ascertain stationarity), there is a fixed nonzero probability that the actions taken in any given epoch, by chance, do not appear to be drawn from the equilibrium distribution (or, when ascertaining stationarity, that the actual distributions of actions played in consecutive epochs do not look alike).[7] Thus, with probability 1, AWESOME would eventually restart. So, AWESOME could never converge (because it will play a random action between each pair of restarts). Second, AWESOME would not be able to distinguish a strategy from the precomputed equilibrium strategy if those strategies are within ϵ_e of each other. Similarly, AWESOME would not be able to detect nonstationarity if the distributions of actions played in consecutive epochs are within ϵ_s.

We can fix both of these problems by letting the distance ϵ_e and ϵ_s decrease each epoch, while simultaneously increasing the epoch length N. If we increase N sufficiently fast, the probability that the equilibrium distribution would by chance produce a sequence of actions that does not appear to be drawn from it will decrease each epoch in spite of the decrease in ϵ_e. (Similarly, the probability that a stationary distribution will, in consecutive epochs, produce action distributions that are further than ϵ_s apart will decrease in spite of the decrease in ϵ_s.) In fact, these probabilities can be decreased so fast that there is nonzero probability that the equilibrium hypothesis (resp. stationarity hypothesis) will *never* be rejected over an infinite number of epochs. Chebyshev's inequality, which states that $P(|X - E(X)| \geq t) \leq \frac{Var(X)}{t^2}$, will be a crucial tool in demonstrating this.

3.4 The algorithm skeleton

We now present the backbone of the algorithm for repeated games.

First we describe the variables used in the algorithm. Me refers to the AWESOME player. π_p^* is player p's equilibrium strategy. ϕ is the AWESOME player's current strategy. h_p^{prev} and h_p^{curr} are the histories of actions played by player p in the previous epoch and the epoch just played, respectively. (h_{-Me}^{curr} is the vector of all h_p^{curr} besides the AWESOME player's.) t is the current epoch (reset to 0 every restart). *APPE* (all players playing equilibrium) is *true* if the equilibrium hypothesis has not been rejected. *APS* (all players stationary) is *true* if the stationarity hypothesis has not been rejected. β is *true* if the equilibrium hypothesis was just rejected (and gives one epoch to adapt before the stationarity hypothesis can be rejected). $\epsilon_e^t, \epsilon_s^t, N^t$ are the values of those variables for epoch t. n is the number of players, $|A|$ the maximum number of actions for a single player, μ (also a constant) the utility difference between the AWESOME player's best and worst outcomes in the game.

Now we describe the functions used in the algorithm. ComputeEquilibriumStrategy computes the equilibrium strategy for a player. Play takes a strategy as input, and plays an action drawn from that distribution. Distance computes the distance (as defined above) between strategies (or histories). V computes the expected utility of playing a given strategy or action against a given strategy profile for the others.

[7] This holds for all distributions except those that correspond to pure strategies.

We are now ready to present the algorithm.

```
AWESOME()
 1. for each p
 2.   π*_p := ComputeEquilibriumStrategy(p)
 3. repeat {// beginning of each restart
 4.   for each player p {
 5.     InitializeToEmpty(h_p^prev)
 6.     InitializeToEmpty(h_p^curr)
 7.   }
 8.   APPE := true
 9.   APS := true
10.   β := false
11.   t := 0
12.   φ := π*_Me
13.   while APS { // beginning of each epoch
14.     repeat N^t times {
15.       Play(φ)
16.       for each player p
17.         Update(h_p^curr)
18.     }
19.     if APPE = false {
20.      if β = false
21.       for each player p
22.        if (Distance(h_p^curr, h_p^prev) > ε_s^t)
23.          APS := false
24.      β := false
25.      a := arg max V(a, h_-Me^curr)
26.      if V(a, h_-Me^curr) > V(φ, h_-Me^curr) +n|A|ε_s^{t+1}μ
27.        φ := a
28.     }
29.     if APPE = true
30.      for each player p
31.       if (Distance(h_p^curr, π*_p) > ε_e^t) {
32.         APPE := false
33.         φ := RandomAction()
34.         β := true
35.       }
36.     for each player p {
37.       h_p^prev := h_p^curr
38.       InitializeToEmpty(h_p^curr)
39.     }
40.     t := t + 1
41.   }
42. }
```

We still need to discuss how to set the schedule for $(\epsilon_e^t, \epsilon_s^t, N^t)$. This is the topic of the next section.

3.5 Valid schedules

We now need to consider more precisely what good schedules are for changing the epochs' parameters. It turns out that the following conditions on the schedule for decreasing ϵ_e and ϵ_s and increasing N are sufficient for the desirable properties to hold. The basic idea is to make

N go to infinity relatively fast compared to the ϵ_e and ϵ_s. The reason for this exact definition will become clear from the proofs in the next section.

Definition 4. A schedule $\{(\epsilon_e^t, \epsilon_s^t, N^t)\}_{t\in\{0,1,2,\ldots\}}$ is *valid* if

- ϵ_s^t, ϵ_e^t decrease monotonically and converge to 0.
- $N^t \to \infty$.
- $\prod_{t\in\{1,2,\ldots\}}(1 - |A|_\Sigma \frac{1}{N^t(\epsilon_s^{t+1})^2}) > 0$ (with all factors > 0), where $|A|_\Sigma$ is the total number of actions summed over all players.
- $\prod_{t\in\{1,2,\ldots\}}(1 - |A|_\Sigma \frac{1}{N^t(\epsilon_e^t)^2}) > 0$ (with all factors > 0).

The next theorem shows that a valid schedule always exists.

Theorem 1. *A valid schedule always exists.*

Proof: Let $\{\epsilon_e^t = \epsilon_s^{t+1}\}_{t\in\{0,1,2,\ldots\}}$ be any decreasing sequence going to 0. Then let $N^t = \lceil \frac{|A|_\Sigma}{(1-\frac{1}{2^{(\frac{1}{t})^2}})(\epsilon_e^t)^2} \rceil$ (which indeed goes to infinity). Then, $\prod_{t\in\{1,2,\ldots\}} 1 - |A|_\Sigma \frac{1}{N^t(\epsilon_s^{t+1})^2} = \prod_{t\in\{1,2,\ldots\}} 1 - |A|_\Sigma \frac{1}{N^t(\epsilon_e^t)^2} \geq \prod_{t\in\{1,2,\ldots\}} \frac{1}{2^{(\frac{1}{t})^2}}$ (we also observe that all factors are > 0). Also, $\prod_{t\in\{1,2,\ldots\}} \frac{1}{2^{(\frac{1}{t})^2}} = 2^{\sum_{t\in\{1,2,\ldots\}} \log \frac{1}{2^{(\frac{1}{t})^2}}} = 2^{\sum_{t\in\{1,2,\ldots\}} -(\frac{1}{t})^2}$. Because the sum in the exponent converges, it follows that this is positive. □

4 AWESOME learns a best-response against eventually stationary opponents

In this section we show that if the other agents use fixed (potentially mixed) strategies, then AWESOME learns to play a best-response strategy against the opponents. This holds even if the opponents are nonstationary first (e.g., because they are learning themselves), as long as they become stationary at some time.

Theorem 2. *With a valid schedule, if all the other players play fixed strategies forever after some round, AWESOME converges to a best response with probability 1.*

Proof: We prove this in two parts. First, we prove that after any given restart, with nonzero probability, the AWESOME player never restarts again. Second, we show that after any given restart, the probability of never restarting again without converging on the best response is 0. It follows that with probability 1, we will eventually converge.

To show that after any given restart, with nonzero probability, the AWESOME player never restarts again: consider the probability that for all t (t being set to 0 right after the restart), we have $max_{p\neq Me}\{d(\phi_p^t, \phi_p)\} \leq \frac{\epsilon_s^{t+1}}{2}$ (where the AWESOME player is player Me, ϕ_p^t is the distribution of actions actually played by p in epoch t, and ϕ_p is the (stationary) distribution that p is actually playing from). This probability is given by $\prod_{t\in\{1,2,\ldots\}}(1 - P(max_{p\neq Me}\{d(\phi_p^t, \phi_p)\} > \frac{\epsilon_s^{t+1}}{2}))$, which is greater than $\prod_{t\in\{1,2,\ldots\}}(1 - \sum_{p\neq Me} P(d(\phi_p^t, \phi_p) > \frac{\epsilon_s^{t+1}}{2}))$, which in turn is greater than $\prod_{t\in\{1,2,\ldots\}}(1 - \sum_{p\neq Me}\sum_a P(|\phi_p^t(a) - \phi_p(a)| > \frac{\epsilon_s^{t+1}}{2}))$ (where $\phi_p(a)$ is the probability ϕ_p places on a). Because $E(\phi_p^t(a)) = \phi_p(a)$, and observing $Var(\phi_p^t(a)) \leq \frac{1}{4N^t}$, we can now apply Chebyshev's inequality and conclude that the whole product is greater than $\prod_{t\in\{1,2,\ldots\}} 1 - |A|_\Sigma \frac{1}{N^t(\epsilon_s^{t+1})^2}$,

where $|A|_\Sigma$ is the total number of actions summed over all players.[8] But for a valid schedule, this is greater than 0.

Now we show that if this event occurs, then *APS* will not be set to *false* on account of the stationary players. This is because

$$d\left(\phi_p^t, \phi_p^{t-1}\right) > \epsilon_s^t \Rightarrow d\left(\phi_p^t, \phi_p\right) + d\left(\phi_p^{t-1}, \phi_p\right) > \epsilon_s^t \Rightarrow$$

$$d\left(\phi_p^t, \phi_p\right) > \frac{\epsilon_s^t}{2} \vee d\left(\phi_p^{t-1}, \phi_p\right) > \frac{\epsilon_s^t}{2} \Rightarrow d\left(\phi_p^t, \phi_p\right) > \frac{\epsilon_s^{t+1}}{2} \vee d\left(\phi_p^{t-1}, \phi_p\right) > \frac{\epsilon_s^t}{2}$$

(using the triangle inequality and the fact that the ϵ_s are stricly decreasing).

All that is left to show for this part is that, given that this happens, *APS* will, with some nonzero probability, not be set to *false* on account of the AWESOME player. Certainly this will not be the case if *APPE* remains *true* forever, so we can assume that this is set to *false* at some point. Then, with probability at least $\frac{1}{|A|}$, the first action b that the AWESOME player will choose after *APPE* is set to *false* is a best response to the stationary strategies. (We are making use of the fact that the stationary players' actions are independent of this choice.) We now claim that if this occurs, then *APS* will not be set to *false* on account of the AWESOME player, because the AWESOME player will play b forever. This is because the expected utility of playing any action a against players who play from distributions ϕ_{-Me}^t (call this $u_{Me}(a, \phi_{-Me}^t)$) can be shown to differ at most $n|A| \max_{p \neq Me} d(\phi_p, \phi_p^t)\mu$ from the expected utility of playing action a against players who play from distributions ϕ_{-Me} (call this $u_{Me}(a, \phi_{-Me})$). Thus, for any t and any a, we have

$$u_{Me}\left(a, \phi_{-Me}^t\right) \leq u_{Me}(a, \phi_{-Me}) + n|A|\epsilon_s^{t+1}\mu \leq u_{Me}(b, \phi_{-Me}) + n|A|\epsilon_s^{t+1}\mu$$

(because b is a best-response to ϕ_{-Me}), and it follows that the AWESOME player will never change its strategy.

Now, to show that after any given restart, the probability of never restarting again without converging on the best response is 0: there are two ways in which this could happen, namely with *APPE* being set to *true* forever, or with it set to *false* at some point. In the first case, we can assume that the stationary players are not actually playing the precomputed equilibrium (because in this case, the AWESOME player would actually be best-responding forever). Let $p \neq Me$ and a be such that $\phi_p(a) \neq \pi_p^*(a)$, where $\pi_p^*(a)$ is the equilibrium probability p places on a. Let $d = |\phi_p(a) - \pi_p^*(a)|$. By Chebyshev's inequality, the probability that $\phi_p^t(a)$ is within $\frac{d}{2}$ of $\phi_p(a)$ is at least $1 - \frac{1}{N^t d^2}$, which goes to 1 as t goes to infinity (because N^t goes to infinity). Because ϵ_e^t goes to 0, at some point $\epsilon_e^t < \frac{d}{2}$, so $|\phi_p^t(a) - \phi_p(a)| < \frac{d}{2} \Rightarrow |\phi_p^t(a) - \pi_p^*(a)| > \epsilon_e^t$. With probability 1, this will be true for some $\phi_p^t(a)$, and at this point *APPE* will be set to *false*. So the first case happens with probability 0. For the second case where *APPE* is set to *false* at some point, we can assume that the AWESOME player is not playing any best-response b forever from some point onwards, because in this case the AWESOME player would have converged on a best response. All we have to show is that from any epoch t onwards, with probability 1, the AWESOME player will eventually switch actions (because starting at some epoch t, ϵ_s will be small enough that this will cause *APS* to be set to *false*). If playing an action a against the true profile ϕ_{-Me} gives expected utility k less than playing b, then by continuity, for some ϵ, for any strategy profile ϕ'_{-Me} within distance ϵ of the true profile ϕ_{-Me}, playing a against ϕ'_{-Me} gives expected utility at least $\frac{k}{2}$ less than playing b. By an argument similar to that made in the first case, the

[8] We used the fact that the schedule is valid to assume that the factors are greater than 0 in the manipulation.

probability of ϕ^t_{-Me} being within ϵ of the true profile ϕ_{-Me} goes to 1 as t goes to infinity; and because eventually, $n|A|\epsilon_s^{t+1}\mu$ will be smaller than $\frac{k}{2}$, this will cause the AWESOME player to change actions. □

5 AWESOME converges to a Nash equilibrium in self-play

In this section we show that AWESOME converges to a Nash equilibrium when all the other players are using AWESOME as well.

Theorem 3. *With a valid schedule, AWESOME converges to a Nash equilibrium in self-play with probability 1.*

Proof: We first observe that the values of *APPE* and *APS* are always the same for all the (AWESOME) players, due to the synchronization efforts in the algorithm. It can be shown in a manner similar to the proof of Theorem 2 that after any restart, with nonzero probability, we have, for all t, $\max_p\{d(\phi^t_p, \pi^*_p)\} \leq \epsilon^t_e$ (where ϕ^t_p is the distribution of actions actually played by p in epoch t, and π^*_p is the equilibrium strategy for p). In this case, *APPE* is never set to *false* and the players play the equilibrium forever.

All that is left to show is that, after any restart, the probability of never restarting while not converging to an equilibrium is 0. This can only happen if *APPE* is set to *false* at some point, and the players do not keep playing a pure-strategy equilibrium forever starting at some point after this. As in the proof of Theorem 2, all we have to show is that from any epoch t onwards, with probability 1, some player will eventually switch actions (because starting at some epoch t, ϵ_s will be small enough that this will cause *APS* to be set to false). Because we can assume that at least one player is not best-responding to the others' actions, the proof of this claim is exactly identical to that given in the proof of Theorem 2. □

It is interesting to observe that even in self-play, it is possible (with nonzero probability) that AWESOME players converge to an equilibrium other than the precomputed equilibrium. Consider a game with a pure-strategy equilibrium as well as a mixed-strategy equilibrium where every action is played with positive probability. If the mixed-strategy equilibrium is the one that is precomputed, it is possible that the equilibrium hypothesis (by chance) is rejected, and that each player (by chance) picks its pure-strategy equilibrium action after this. Because from here on, the players will always be best-responding to what the others are doing, they will never change their strategies, the stationarity hypothesis will never be rejected, and we have converged on the pure-strategy equilibrium.

6 Experimental results

In this section, we present an experimental evaluation of the AWESOME algorithm. We compare AWESOME's performance to the performance of *fictitious play*. Fictitious play is one of the simplest algorithms for learning in games: the learner simply plays the best response to the opponents' historical distribution of play. In spite of its simplicity, fictitious play has several properties that make it a natural algorithm for comparison: it can be applied to arbitrary games, it does not require the learner to know the opponents' actual mixed strategies, and it converges to a best response against stationary opponents. In addition, when both players use fictitious play, then the players' empirical (marginal) distributions of play converge to a Nash equilibrium under various conditions—for example, when the game is zero-sum (Robinson, 1951), or has generic payoffs and is 2×2 (Miyasawa, 1961), or is

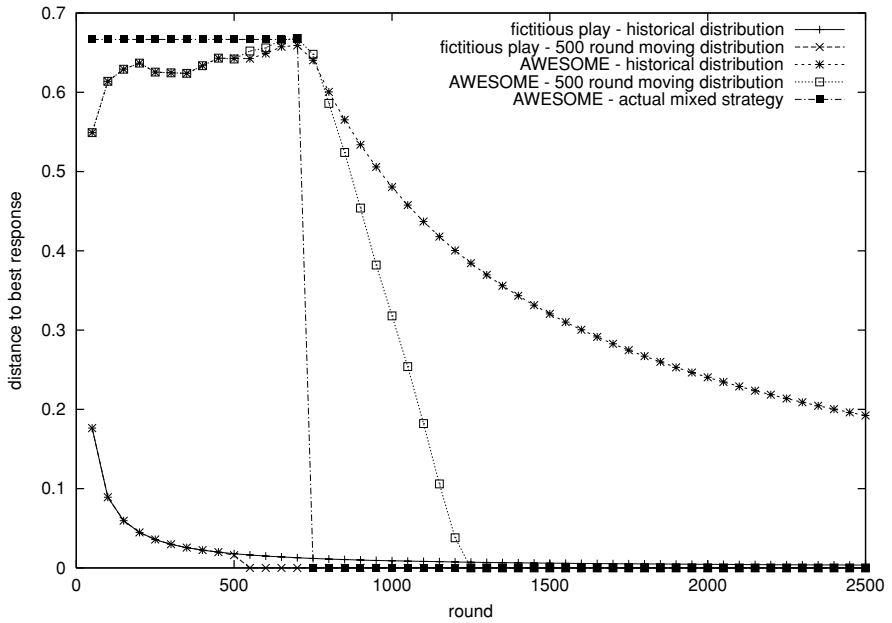

Fig. 1 Play against the stationary player (0.4, 0.6, 0) in rock-paper-scissors

solvable by iterated strict dominance (Nachbar, 1990). However, there are also games in which the distributions do not converge under fictitious play (Shapley, 1964).

In our experiments, we study the convergence of (1) the empirical distribution of play (that is, the entire history of actions that were played), (2) the empirical distribution over the last 500 rounds only (a "moving average"), and (3) the actual mixed strategy used in a specific round. We only show 3) for AWESOME, because fictitious play chooses actions deterministically and therefore will never converge to a mixed strategy in this sense. As our distance measure between two distributions p_1 and p_2 over a set S, we use $d(p_1, p_2) = \max_{s \in S} |p_1(s) - p_2(s)|$. We use a valid schedule for AWESOME: we set $\epsilon_s^t = 1/t$ and define the other parameters as in the proof of Theorem 1.

6.1 Rock-paper-scissors

The first game that we study is the well-known rock-paper-scissors game, which is often used as an example to illustrate how fictitious play can be effective (Fudenberg & Levine, 1998).

0.5, 0.5	0, 1	1, 0
1, 0	0.5, 0.5	0, 1
0, 1	1, 0	0.5, 0.5

Rock-paper-scissors.

In the unique Nash equilibrium of this game, each player plays each action with probability 1/3. In Fig. 1, we show experimental results for playing against a stationary opponent that uses the mixed strategy (0.4, 0.6, 0).[9] Fictitious play converges to the best response very

[9] There is no particular reason for using this distribution (or, for that matter, for using any other distribution).

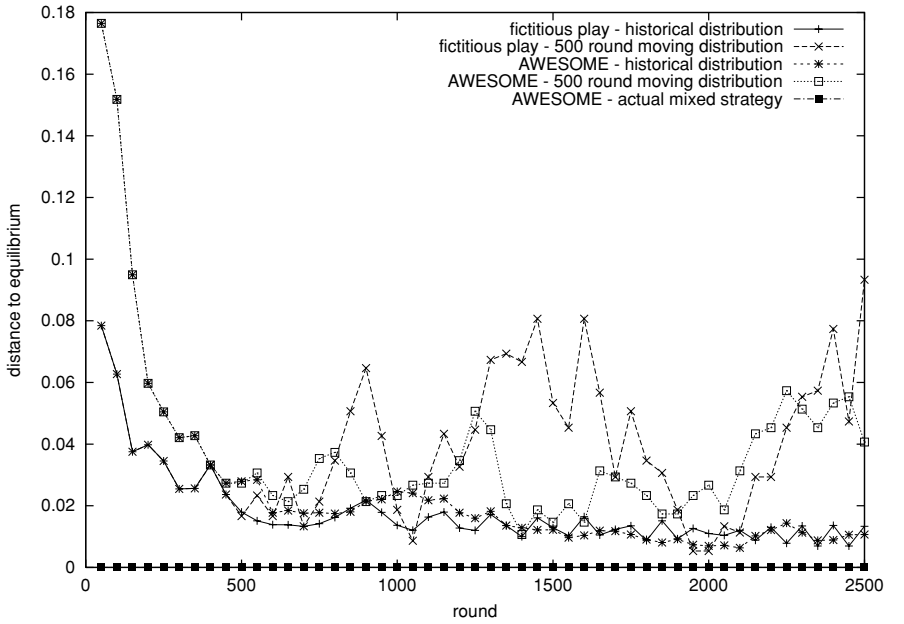

Fig. 2 Self-play in rock-paper-scissors

rapidly. This is not surprising, as fictitious play is an ideal algorithm for playing against a stationary player: it best-responds against the best estimate of the opponent's strategy. AWESOME initially plays the equilibrium strategy, but eventually rejects the equilibrium hypothesis, and from that point plays the best response. It takes a large number of rounds for AWESOME's historical distribution to converge because of the other actions it played before it rejected the equilibrium hypothesis; but the moving distribution converges rapidly once AWESOME starts best-responding.

In Fig. 2, we show experimental results for self-play (in which each algorithm plays against a copy of itself). Both algorithms perform well here (note the changed scale on the y-axis). For fictitious play, it is known that the players' empirical distributions of play converge to the equilibrium distributions in all zero-sum games (Robinson, 1951), and rock-paper-scissors is a zero-sum game. AWESOME never rejects the equilibrium hypothesis and therefore always plays according to the Nash equilibrium. We note that the 500-round moving distribution cannot be expected to converge exactly: when drawing from a mixed strategy a fixed number of times only, the empirical distribution will rarely coincide exactly with the actual distribution.

6.2 Shapley's game

The other game that we study is Shapley's game, which is often used as an example to illustrate how fictitious play can fail (Fudenberg & Levine, 1998).

0, 1	0, 0	1, 0
0, 0	1, 0	0, 1
1, 0	0, 1	0, 0

Shapley's game.

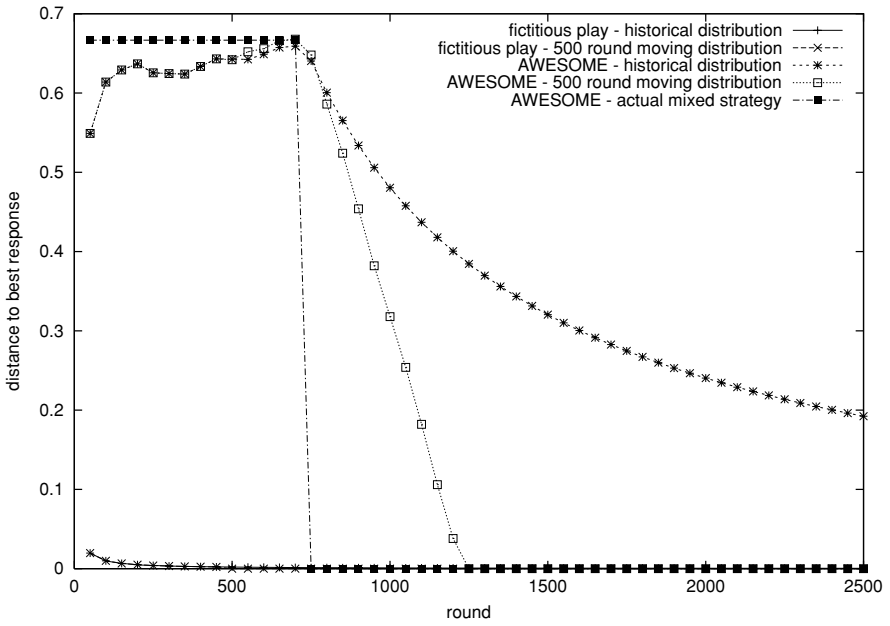

Fig. 3 Play against the stationary player (0.4, 0.6, 0) in Shapley's game

Again, in the unique Nash equilibrium of this game, each player plays each action with probability 1/3. In Fig. 3, we show experimental results for playing against a stationary opponent that uses the mixed strategy (0.4, 0.6, 0). The results are similar to those for rock-paper-scissors.

Finally, in Fig. 4, we show experimental results for self-play. Fictitious play now cycles and the empirical distributions never converge (as was first pointed out by Shapley himself (Shapley, 1964)). Because the length of the cycles increases over time, the 500-round moving distribution eventually places all of the probability on a single action and is therefore as far away from equilibrium as possible. AWESOME, on the other hand, again never rejects the equilibrium hypothesis and therefore continues to play the equilibrium.

7 Discussion of alternative learning objectives

In this section, we discuss alternative objectives that one may pursue for learning in games, and we argue for the importance of the objectives that we pursued (and that AWESOME achieves).

7.1 Convergence to equilibrium in self-play

In self-play, AWESOME converges to a Nash equilibrium of the stage game. One may well wonder whether this requirement is unnecessarily strong: various weaker notions are available. For instance, one may consider *correlated* equilibrium (Aumann, 1974), in which the players receive correlated random signals (before playing the stage game), on which they can then base their play. This is not unreasonable, and it has the advantages that correlated

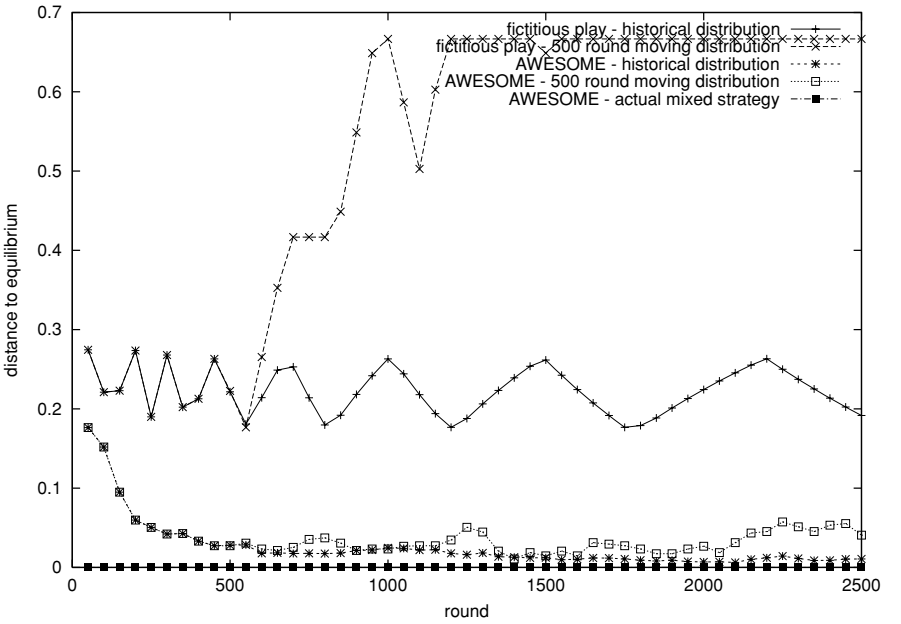

Fig. 4 Self-play in Shapley's game

equilibria can support better outcomes, and that correlated equilibria can be computed in polynomial time using linear programming. However, it does rely on the presence of signals in the world that allow the correlated equilibria to be played. The fewer the signals that are available in the world, the fewer the correlated equilibria that can be played; and, in the extreme case where no correlated signals are available, the set of correlated equilibria that can be played coincides with the set of Nash equilibria. Thus, any learning algorithm that is robust to the absence of correlated signals must be able to converge to Nash equilibrium. We note that there is a family of algorithms that converge to the set of correlated equilibria in terms of the *empirical distribution of play* (Foster & Vohra, 1997; Fudenberg & Levine, 1999; Hart & Mas-Colell, 2000; Cahn, 2000; Greenwald & Jafari, 2003; Kakade & Foster, 2004). This is a much weaker criterion and it does not require the presence of any external correlated signals. We will compare convergence of the stage-game strategies with convergence of the empirical distribution of play in Subsection 7.3.

One may also consider (Nash) equilibria *of the repeated game* that do not correspond to an equilibrium of the stage game. For instance, cooperation in the Prisoner's Dilemma—a dominated strategy in the stage game—can correspond to equilibrium play in the repeated game, if the players believe that once they defect, the opponent will defect in the future. Again, this is a very reasonable solution concept, and, as the Prisoner's Dilemma example shows, it may even lead to better outcomes (as in the case of correlated equilibrium). Nash equilibria of the repeated game can also be constructed in polynomial time, if players wish to maximize the limit of their average payoff (Littman & Stone, 2003). Nevertheless, there are various reasons to prefer an algorithm that converges to a Nash equilibrium of the stage game. First of all, it is in general impossible to state whether strategies constitute a Nash equilibrium of the repeated game, unless we know how the agents value future outcomes. One way in which future outcomes may be valued is to take the limit of the average payoff.

This has the odd property that the agent will not care about the outcomes of any finite set of periods of play. Another way to value future outcomes is through discounting. In the limit case where the future is extremely discounted, only the outcome of the current period matters, and thus any equilibrium of the repeated game is an equilibrium of the stage game. Thus, any algorithm that is robust to extreme discounting must be able to converge to an equilibrium of the stage game. Another reason to prefer learning algorithms that converge to an equilibrium of the stage game is the following: equilibria of the repeated game do not make much sense in scenarios in which the game is repeated only for learning purposes, *e.g.* when we are training our agent to play (say) soccer by having it play over and over again. In such scenarios, it does not make much sense to talk about discount factors and the like. Finally, equilibria of the repeated game require the agent to have complex beliefs over what the other agents would do in various future scenarios, and it is less clear where such beliefs might come from than it is for simple beliefs over what the other agents will play next.

We do not think that these relaxed equilibrium notions should be dismissed for the purpose of learning in games, since they (for example) may lead to better outcomes. Nevertheless, we believe that the arguments above show that learning algorithms should be able to converge to a Nash equilibrium of the stage game *at a minimum*, because at least in some settings this is the only sensible outcome.

One may also criticize the Nash equilibrium concept as being too *weak*—for example, one may require that play converges to a Pareto optimal Nash equilibrium. This falls under the category of requiring the algorithm to have additional properties, and we have already acknowledged that this may be desirable. Similarly, convergence against a wider class of learning opponents, rather than just in self-play, is a desirable goal.

7.2 Best-responding against stationary opponents

One may also wonder whether the other requirement—eventual best response against stationary opponents—is really necessary. Stationary agents are irrational (at least those that continue to play a strategy that is not optimal for themselves), so why should they ever occur? There are various possible reasons for this. First, especially for complex games, humans often design agents by crafting a strategy by hand that the human believes will perform reasonably well (but that is definitely suboptimal). Learning to take advantage of such suboptimal opponents (in competitive scenarios) or to perform reasonably well in spite of such opponents' suboptimality (in cooperative scenarios) is an important capability for an agent. As another reason, the process that controls the opponent's actions may not actually correspond to a rational agent; rather, our agent may in reality be playing against (indifferent) Nature. (This will also happen if the opponent agent is unable to change its strategy, for example because the agent is defective.)

Another possible reason is that the other agents may merely be *satisficing* (Simon, 1982), pursuing a level of utility that they consider satisfactory. Indeed, satisficing approaches in learning in games have been proposed (Stimpson, Goodrich, & Walters, 2001). In this case they will be content to continue playing any strategy that gives them this desired level of utility, even if it does not maximize their utility. If the desired level of utility is low enough, these agents will be content to play any mixed strategy; thus, any learning algorithm that is robust to this extreme satisficing scenario needs to be able to converge to a best response against any stationary opponents. (Of course, this is assuming that we ourselves do not take a satisficing approach.)

Again, one may criticize this criterion as being too weak and suggest something stronger. For example, we may wish the average *regret* against any opponents to go to zero.[10] This is an admirable objective that has been pursued (and achieved) in a significant body of research (Littlestone & Warmuth, 1994; Auer et al., 1995; Fudenberg & Levine, 1995; Freund & Schapire, 1999; Hart & Mas-Colell, 2000; Jafari et al., 2001; Greenwald & Jafari, 2003; Zinkevich, 2003; Bowling, 2005). Alternatively, we may wish to learn to best-respond against larger classes of opponents (Powers & Shoham, 2005a,b).

7.3 Convergence of stage-game strategies versus convergence of empirical distribution of play

In self-play, we require that the agents' *stage-game strategies* converge to a Nash equilibrium. This is much stronger than, for example, convergence of the *empirical distribution of play*, where the distribution of the obtained outcomes converges to some equilibrium. A variety of approaches is available for converging, in terms of the empirical distribution of play, to the set of correlated equilibria (Foster & Vohra, 1997; Fudenberg & Levine, 1999; Hart & Mas-Colell, 2000; Cahn, 2000; Greenwald & Jafari, 2003) (or even to the set of convex combinations of Nash equilibria (Kakade & Foster, 2004), which is contained in the set of correlated equilibria). We believe that it is important that the agents' stage-game strategies converge, for the following reason. Convergence in the empirical distribution of play may be achieved by an agent whose stage-game strategy is entirely unlike the empirical distribution of play. For example, an agent that never randomizes over the next action to take may converge to a mixed-strategy equilibrium in the empirical distribution of play. It does not seem that the agent has actually learned how to play the game in this case, because even in the end the agent is not actually playing a mixed strategy. The agent could conceivably eventually take a step back, consider the empirical distribution of play, and play a (mixed) strategy corresponding to this distribution. This has the rather awkward effect of separating the process into a learning phase, during which the agent's strategy remains unlike the desired strategy, and an execution phase, in which the agent stops the learning process and decides to play according to what it has learned. Stopping the learning process may prevent the agent from converging completely to the desired strategy—for example, it is well-known that certain games have only equilibria with irrational probabilities (Nash, 1950), which cannot correspond to the fraction of time that an action was played during a finite number of rounds. Another disadvantage of such an approach is that once the learning phase has ended, the agent will no longer be able to adapt to changes in the opponent's strategy (for example, the opponent may irrationally stop playing a best response because it is defective, or, alternatively, the opponent may switch to another best response, to which the former agent's equilibrium strategy is not a best response).

7.4 Thinking about the learning process strategically

A final criticism of our approach, one that goes beyond discussions of which variant of a definition should be used, is that *the learning process itself should be viewed strategically*. Thus, we should apply equilibrium notions to the learning process itself, or at the very least attempt to obtain a good result relative to our opponents' strategies (not just the strategies used in rounds of the game, but their entire *learning* strategies). For example, in this view,

[10] Technically this is not a stronger criterion because one may play any strategy (even one far away from any best response) infinitely often and still have the average regret go to 0, but in practice this is unlikely to occur.

having our agent rapidly converge to an equilibrium that is bad for it is not as desirable as trying to steer the learning process towards an equilibrium that is good for it, even if this may fail with some probability. This line of reasoning has been advocated for at least a decade, and some results down that avenue have recently been derived (Brafman & Tennenholtz, 2004, 2005).

While this argument is entirely sensible, the question that immediately arises is what the role of learning is in this setting, and why this does not correspond to "simply" computing an equilibrium of the repeated game. One possibility is that (parts of) the payoff matrices are not known and have to be learned (although, in principle, this can still be modeled as an (even larger) game of incomplete information). Interestingly, if agents have beliefs directly over their opponents' strategies and update these beliefs using Bayes' rule, then a well-known result states that play will converge to Nash equilibrium, *if* every measurable set of outcomes that has positive probability under their actual strategies has positive probability under each agent's prior belief (Kalai & Lehrer, 1993). However, it has been argued that this last requirement is unreasonably restrictive (Nachbar, 1997, 2001), and that there exist in fact games where such "rational learning" will never come close to a Nash equilibrium (Foster & Young, 2001). The notion of *Efficient Learning Equilibrium* (Brafman & Tennenholtz, 2004, 2005) has been proposed as an alternative to the use of prior distributions over either the game being played or the opponents' strategies. Under this equilibrium definition, deviations must become irrational after a polynomial number of steps, and the payoffs must approach those of a Nash equilibrium after a polynomial number of steps if everyone sticks to the learning algorithm.

A few of the difficulties that we pointed out in the context of learning an equilibrium of the repeated game occur in the context of strategic learning as well. It is not always clear how future payoffs should be compared to current ones (such as in the setting where we are merely training our agent). Also, strategic learning requires the agent to have very sophisticated models of its opponents' future behavior. Still, it would be desirable to design an algorithm that achieves some notion of strategic rationality in addition to the properties pursued in this paper—to the extent that this is possible.

8 Conclusions

We have argued that a satisfactory multiagent learning algorithm should, *at a minimum*, learn to play optimally against stationary opponents, and converge to a Nash equilibrium in self-play. The previous algorithm that has come closest, WoLF-IGA, has been proven to have these two properties in 2-player 2-action repeated games—assuming that the opponent's mixed strategy is observable. Another algorithm, ReDVaLeR (which was introduced after the AWESOME algorithm), achieves the two properties in games with arbitrary numbers of actions and players, but still requires that the opponents' mixed strategies are observable. ReDVaLeR explicitly checks whether the opponents' strategies are stationary, and behaves differently based on the result of the check. Hence, the assumption that the mixed strategies are observable allows this algorithm to achieve each property separately.

In this paper we presented AWESOME, the first algorithm that is guaranteed to have the two properties in games with arbitrary numbers of actions and players, and still the only algorithm that does so while only relying on observing the other players' actual actions (not their mixed strategies). AWESOME also does not use infinitesimal step sizes, and it learns to play optimally against opponents that *eventually become* stationary.

The basic idea behind AWESOME (*Adapt When Everybody is Stationary, Otherwise Move to Equilibrium*) is to try to adapt to the other agents' strategies when they appear stationary, but otherwise to retreat to a precomputed equilibrium strategy. At any point in time, AWESOME maintains either of two null hypotheses: that the others are playing the precomputed equilibrium, or that the others are stationary. Whenever both of these hypotheses are rejected, AWESOME restarts completely. AWESOME may reject either of these hypotheses based on actions played in an epoch. Over time, the epoch length is carefully increased and the criterion for hypothesis rejection tightened to obtain the convergence guarantee. The AWESOME algorithm is also self-aware: when it detects that its own actions signal nonstationarity to the others, it restarts itself for synchronization purposes.

While the algorithm is primarily intended as a theoretical contribution, experimental results comparing AWESOME to fictitious play suggest that AWESOME actually converges quite fast in practice. Fictitious play converges to a best response against a stationary opponent faster than AWESOME, which is not surprising because fictitious play plays a best response against the best estimate of the opponent's strategy. However, in a game where fictitious play converges to a Nash equilibrium in self-play (in terms of the empirical distribution of play), both algorithms converge similarly fast. Unlike AWESOME, fictitious play does not always converge in self-play, and does not converge to a mixed *stage-game* strategy.

9 Future research

The techniques used in proving the properties of AWESOME are fundamentally different from those used for other algorithms pursuing the same properties, because the requirement that the opponents' mixed strategies can be observed is dropped. These techniques may also be valuable in the analysis of other learning algorithms in games.

The AWESOME algorithm itself can also serve as a stepping stone for future multiagent learning algorithm development. AWESOME can be viewed as a skeleton—that guarantees the satisfaction of the two minimal desirable properies—on top of which additional techniques may be used in order to guarantee further desirable properties (such as those discussed in Section 7).

There are several open research questions regarding AWESOME. First, it is important to determine which valid schedules give *fast* convergence. This could be studied from a theoretical angle, by deriving asymptotic bounds on the running time for families of schedules. It could also be studied experimentally for representative families of games. A related second question is whether there are any structural changes that can be made to AWESOME to improve the convergence time while maintaining the properties derived in this paper. For instance, maybe AWESOME does not need to forget the entire history when it restarts. A third question is whether one can integrate learning the structure of the game seamlessly into AWESOME (rather than first learning the structure of the game and then running AWESOME).

Acknowledgments We thank the anonymous reviewers, as well as Michael Bowling and Manuela Veloso for helpful discussions. This material is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, ITR IIS-0121678, and ITR IIS-0427858, a Sloan Fellowship, and an IBM Ph.D. Fellowship.

References

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)* (pp. 322–331).

Aumann, R. (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, *1*, 67–96.

Banerjee, B., & Peng, J. (2004). Performance bounded reinforcement learning in strategic interactions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 2–7). San Jose, CA, USA.

Banerjee, B., Sen, S., & Peng, J. (2001). Fast concurrent reinforcement learners. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 825–830). Seattle, WA.

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

Bowling, M., & Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, *136*, 215–250.

Brafman, R., & Tennenholtz, M. (2000). A near-optimal polynomial time algorithm for learning in certain classes of stochastic games. *Artificial Intelligence*, *121*, 31–47.

Brafman, R., & Tennenholtz, M. (2003). R-max—a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, *3*, 213–231.

Brafman, R., & Tennenholtz, M. (2004). Efficient learning equilibrium. *Artificial Intelligence*, *159*, 27–47.

Brafman, R., & Tennenholtz, M. (2005). Optimal efficient learning equilibrium: Imperfect monitoring in symmetric games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 726–731). Pittsburgh, PA, USA.

Cahn, A. (2000). *General procedures leading to correlated equilibria*. Discussion paper 216, Center for Rationality, The Hebrew University of Jerusalem, Israel.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 746–752). Madison, WI.

Conitzer, V., & Sandholm, T. (2003a). BL-WoLF: A framework for loss-bounded learnability in zero-sum games. In *International Conference on Machine Learning (ICML)* (pp. 91–98). Washington, DC, USA.

Conitzer, V., & Sandholm, T. (2003b). Complexity results about Nash equilibria. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 765–771). Acapulco, Mexico.

Conitzer, V., & Sandholm, T. (2004). Communication complexity as a lower bound for learning in games. In *International Conference on Machine Learning (ICML)* (pp. 185–192). Banff, Alberta, Canada.

Foster, D., & Vohra, R. (1997). Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, *21*, 40–55.

Foster, D. P., & Young, H. P. (2001). On the impossibility of predicting the behavior of rational agents. In *Proceedings of the National Academy of Sciences*, (Vol. 98, pp. 12848–12853).

Freund, Y., & Schapire, R. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, *29*, 79–103.

Fudenberg, D., & Levine, D. (1998). *The theory of learning in games*. MIT Press.

Fudenberg, D., & Levine, D. (1999). Conditional universal consistency. *Games and Economic Behavior*, *29*, 104–130.

Fudenberg, D., & Levine, D. K. (1995). Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, *19*, 1065–1089.

Gilboa, I., & Zemel, E. (1989). Nash and correlated equilibria: some complexity considerations. *Games and Economic Behavior*, *1*, 80–93.

Greenwald, A., & Hall, K. (2003). Correlated Q-learning. *International Conference on Machine Learning (ICML)* (pp. 242–249). Washington, DC, USA.

Greenwald, A., & Jafari, A. (2003). A general class of no-regret learning algorithms and game-theoretic equilibria. *Conference on Learning Theory (COLT)*. Washington, DC.

Hart, S., & Mas-Colell, A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, *68*, 1127–1150.

Hart, S., & Mas-Colell, A. (2003). Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, *93*, 1830–1836.

Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: theoretical framework and an algorithm. *International Conference on Machine Learning (ICML)* (pp. 242–250).

Jafari, A., Greenwald, A., Gondek, D., & Ercal, G. (2001). On no-regret learning, fictitious play, and Nash equilibrium. *International Conference on Machine Learning (ICML)* (pp. 226–233). Williams College, MA, USA.

Kakade, S., & Foster, D. (2004). Deterministic calibration and Nash equilibrium. In *Conference on Learning Theory (COLT)*. Banff, Alberta, Canada.
Kalai, E., & Lehrer, E. (1993). Rational learning leads to Nash equilibrium. *Econometrica*, *61*, 1019–1045.
Lemke, C., & Howson, J. (1964). Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, *12*, 413–423.
Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, *108*, 212–261.
Littman, M. (1994). Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)* (pp. 157–163).
Littman, M. (2001). Friend or foe Q-learning in general-sum Markov games. In *International Conference on Machine Learning (ICML)* (pp. 322–328).
Littman, M., & Stone, P. (2003). A polynomial-time Nash equilibrium algorithm for repeated games. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)* (pp. 48–54). San Diego, CA.
Littman, M., & Szepesvári, C. (1996). A generalized reinforcement-learning model: convergence and applications. In *International Conference on Machine Learning (ICML)* (pp. 310–318).
Miyasawa, K. (1961). *On the convergence of the learning process in a 2 × 2 nonzero sum two-person game*. Research memo 33, Princeton University.
Nachbar, J. (1990). Evolutionary selection dynamics in games: Convergence and limit properties. *International Journal of Game Theory*, *19*, 59–89.
Nachbar, J. (1997). Prediction, optimization, and learning in games. *Econometrica*, *65*, 275–309.
Nachbar, J. (2001). Bayesian learning in repeated games of incomplete information. *Social Choice and Welfare*, *18*, 303–326.
Nash, J. (1950). Equilibrium points in n-person games. In *Proc. of the National Academy of Sciences*, *36*, 48–49.
Papadimitriou, C. (2001). Algorithms, games and the Internet. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)* (pp. 749–753).
Pivazyan, K., & Shoham, Y. (2002). Polynomial-time reinforcement learning of near-optimal policies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. Edmonton, Canada.
Porter, R., Nudelman, E., & Shoham, Y. (2004). Simple search methods for finding a Nash equilibrium. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 664–669). San Jose, CA, USA.
Powers, R., & Shoham, Y. (2005a). Learning against opponents with bounded memory. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Edinburgh, UK.
Powers, R., & Shoham, Y. (2005b). New criteria and a new algorithm for learning in multi-agent systems. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. Vancouver, Canada.
Robinson, J. (1951). An iterative method of solving a game. *Annals of Mathematics*, *54*, 296–301.
Sandholm, T., & Crites, R. (1996). Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, *37*, 147–166. Special issue on the Prisoner's Dilemma.
Sandholm, T., Gilpin, A., & Conitzer, V. (2005). Mixed-integer programming methods for finding Nash equilibria. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 495–501). Pittsburgh, PA, USA.
Sen, S., & Weiss, G. (1998). Learning in multiagent systems. In G. Weiss (Ed.), *Multiagent systems: a modern introduction to distributed artificial intelligence* (Chapter 6, pp. 259–298). MIT Press.
Shapley, L. S. (1964). Some topics in two-person games. In M. Drescher, L. S. Shapley & A. W. Tucker (Eds.), *Advances in game theory*. Princeton University Press.
Simon, H. A. (1982). *Models of bounded rationality*, vol. 2. MIT Press.
Singh, S., Kearns, M., & Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)* (pp. 541–548). Stanford, CA.
Stimpson, J., Goodrich, M., & Walters, L. (2001). Satisficing and learning cooperation in the prisoner's dilemma. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 535–540). Seattle, WA.
Tan, M. (1993). Multi-agent reinforcement learning: independent vs. cooperative agents. In *International Conference on Machine Learning (ICML)* (pp. 330–337).
Wang, X., & Sandholm, T. (2002). Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. Vancouver, Canada.
Wang, X., & Sandholm, T. (2003). Learning near-Pareto-optimal conventions in polynomial time. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. Vancouver, Canada.
Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)* (pp. 928–936). Washington, DC, USA.