Journal of Artificial Intelligence Research 40 (2011) 571-598 Submitted 11/10; published 03/11

Multiagent Learning in Large Anonymous Games

Ian A. Kash KASH@SEAS.HARVARD.EDU
Center for Research on Computation and Society
Harvard University

Eric J. Friedman EJF27@CORNELL.EDU
Department of Operations Research
and Information Engineering
Cornell University

Joseph Y. Halpern HALPERN@CS.CORNELL.EDU
Department of Computer Science
Cornell University

Abstract

In large systems, it is important for agents to learn to act effectively, but sophisticated multi-agent learning algorithms generally do not scale. An alternative approach is to find restricted classes of games where simple, efficient algorithms converge. It is shown that stage learning efficiently converges to Nash equilibria in large anonymous games if best-reply dynamics converge. Two features are identified that improve convergence. First, rather than making learning more difficult, more agents are actually beneficial in many settings. Second, providing agents with statistical information about the behavior of others can significantly reduce the number of observations needed.

1. Introduction

Designers of distributed systems are frequently unable to determine how an agent in the system should behave, because optimal behavior depends on the user's preferences and the actions of others. A natural approach is to have agents use a learning algorithm. Many multiagent learning algorithms have been proposed including simple strategy update procedures such as *fictitious play* (Fudenberg & Levine, 1998), multiagent versions of *Q-learning* (Watkins & Dayan, 1992), and *no-regret algorithms* (Cesa-Bianchi & Lugosi, 2006).

Our goal in this work is to help the designers of distributed systems understand when learning is practical. As we discuss in Section 2, existing algorithms are generally unsuitable for large distributed systems. In a distributed system, each agent has a limited view of the actions of other agents. Algorithms that require knowing, for example, the strategy chosen by every agent cannot be implemented. Furthermore, the size of distributed systems requires fast convergence. Users may use the system for short periods of time and conditions in the system change over time, so a practical algorithm for a system with thousands or millions of users needs to have a convergence rate that is sublinear in the number of agents. Existing algorithms tend to provide performance guarantees that are polynomial or even exponential. Finally, the large number of agents in the system guarantees that there will be noise. Agents will make mistakes and will behave in unexpectedly. Even if no agent changes his strategy, there can still be noise in agent payoffs. For example, a gossip protocol will match different agents from round to round; congestion in the underlying network may effect message delays between agents. A learning algorithm needs to be robust to this noise.

While finding an algorithm that satisfies these requirements for arbitrary games may be difficult, distributed systems have characteristics that make the problem easier. First, they involve a large number of agents. Having more agents may seem to make learning harder—after all, there are more possible interactions. However, it has the advantage that the outcome of an action typically depends only weakly on what other agents do. This makes outcomes robust to noise. Having a large number of agents also make it less useful for an agent to try to influence others; it becomes a better policy to try to learn an optimal response. In contrast, with a small number of agents, an agent can attempt to guide learning agents into an outcome that is beneficial for him.

Second, distributed systems are often *anonymous*; it does not matter *who* does something, but rather *how many* agents do it. For example, when there is congestion on a link, the experience of a single agent does not depend on who is sending the packets, but on how many are being sent. Anonymous games have a long history in the economics literature (e.g., Blonski, 2001) and have been a subject of recent interest in the computer science literature (Daskalakis & Papadimitriou, 2007; Gradwohl & Reingold, 2008).

Finally, and perhaps most importantly, in a distributed system the system designer controls the game agents are playing. This gives us a somewhat different perspective than most work, which takes the game as given. We do not need to solve the hard problem of finding an efficient algorithm for all games. Instead, we can find algorithms that work efficiently for interesting classes of games, where for us "interesting" means "the type of games a system designer might wish agents to play." Such games should be "well behaved," since it would be strange to design a system where an agent's decisions can influence other agents in pathological ways.

In Section 3, we show that *stage learning* (Friedman & Shenker, 1998) is robust, implementable with minimal information, and converges efficiently for an interesting class of games. In this algorithm, agents divide the rounds of the game into a series of stages. In each stage, the agent uses a fixed strategy except that he occasionally explores. At the end of a stage, the agent chooses as his strategy for the next stage whatever strategy had the highest average reward in the current stage. We prove that, under appropriate conditions, a large system of stage learners will follow (approximate) best-reply dynamics[1] despite errors and exploration.

For games where best-reply dynamics converge, our theorem guarantees that learners will play an approximate Nash equilibrium. In contrast to previous results, where the convergence guarantee scales poorly with the number of agents, our theorem guarantees convergence in a finite amount of time with an infinite number of agents. While the assumption that best-reply dynamics converge is a strong one, many interesting games converge under best-reply dynamics, including dominance-solvable games, games with monotone best replies, and max-solvable games (Nisan, Schapira, & Zohar, 2008). The class of max-solvable games in particular includes many important games such as Transmission Control Protocol (TCP) congestion control, interdomain routing with the Border Gateway Protocol (BGP), cost-sharing games, and stable-roommates games (Nisan, Schapira, Valiant, & Zohar, 2011).

1. In this paper, we consider best-reply dynamics where all agents update their strategy at the same time. Some other results about best-reply dynamics assume agents update their strategy one at a time.

Marden, Arslan, and Shamma (2007a) have observed that convergence of best-reply dynamics is often a property of games that humans design (although their observation was for a slightly different notion of best-reply dynamics). Moreover, convergence of best-reply dynamics is a weaker assumption than a common assumption made in the mechanism design literature, that the games of interest have dominant strategies (each agent has a strategy that is optimal no matter what other agents do).

Simulation results, presented in Section 4, show that convergence is fast in practice: a system with thousands of agents can converge in a few thousand rounds. Furthermore, we identify two factors that determine the rate and quality of convergence. One is the number of agents: having more agents makes the noise in the system more consistent so agents can learn using fewer observations. The other is giving agents statistical information about the behavior of other agents; this can speed convergence by an order of magnitude. Indeed, even noisy statistical information about agent behavior, which should be relatively easy to obtain and disseminate, can significantly improve performance.

While our theoretical results are limited to stage learning, they provide intuition about why other "well behaved" learning algorithms should also converge. Our simulations, which include two other learning algorithms, bear this out. Furthermore, to demonstrate the applicability of stage learning in more realistic settings, we simulate the results of learning in a scrip system (Kash, Friedman, & Halpern, 2007). Our results demonstrate that stage learning is robust to factors such as churn (agents joining and leaving the system) and asynchrony (agents using stages of different lengths). However, stage learning is not robust to all changes. We include simulations of games with a small number of agents, games that are not anonymous, and games that are not continuous. These games violate the assumptions of our theoretical results; our simulations show that, in these games, stage learning converges very slowly or not at all.

Finally, not all participants in a system will necessarily behave as expected. For learning to be useful in a real system, it needs to be robust to such behavior. In Section 5, we show that the continuity of utility functions is a key property that makes stage learning robust to Byzantine behavior by a small fraction of agents.

2. Related Work

One approach to learning to play games is to generalize reinforcement learning algorithms such as Q-learning (Watkins & Dayan, 1992). One nice feature of this approach is that it can handle games with state, which is important in distributed systems. In Q-learning, an agent associates a value with each state-action pair. When he chooses action a in state s, he updates the value $Q(s,a)$ based on the reward he received and the best value he can achieve in the resulting state s' ($\max_{a'} Q(s',a')$). When generalizing to multiple agents, s and a become vectors of the state and action of every agent and the max is replaced by a prediction of the behavior of other agents. Different algorithms use different predictions; for example, Nash-Q uses a Nash equilibrium calculation (Hu & Wellman, 2003). See the work of Shoham, Powers, and Grenager (2003) for a survey.

Unfortunately, these algorithms converge too slowly for a large distributed system. The algorithm needs to experience each possible action profile many times to guarantee convergence. So, with n agents and k strategies, the naive convergence time is $O(k^n)$. Even with a better representation for anonymous games, the convergence time is still $O(n^k)$ (typically $k \ll n$). There is also a more fundamental problem with this approach: it assumes information that an agent is unlikely to have. In order to know which value to update, the agent must learn the action chosen by every other agent. In practice, an agent will learn something about the actions of the agents with whom he directly interacts, but is unlikely to gain much information about the actions of other agents.

Another approach is *no-regret learning*, where agents choose a strategy for each round that guarantees that the regret of their choices will be low. Hart and Mas-Colell (2000) present such a learning procedure that converges to a *correlated equilibrium*[2] given knowledge of what the payoffs of every action would have been in each round. They also provide a variant of their algorithm that requires only information about the agent's actual payoffs (Hart & Mas-Colell, 2001). However, to guarantee convergence to within ϵ of a correlated equilibrium requires $O(kn/\epsilon^2 \log kn)$, still too slow for large systems. Furthermore, the convergence guarantee is that the distribution of play converges to equilibrium; the strategies of individual learners will not converge. Many other no-regret algorithms exist (Blum & Mansour, 2007). In Section 4, we use the Exp3 algorithm (Auer, Cesa-Bianchi, Freund, & Schapire, 2002). They can achieve even better convergence in restricted settings. For example, Blum, Even-Dar, and Ligett (2006) showed that in routing games a continuum of no-regret learners will approximate Nash equilibrium in a finite amount of time. Jafari, Greenwald, Gondek, and Ercal (2001) showed that no-regret learners converge to Nash equilibrium in dominance solvable, constant sum, and general sum 2×2 games.

Foster and Young (2006) use a stage-learning procedure that converges to Nash equilibrium for two-player games. Germano and Lugosi (2007) showed that it converges for generic n-player games (games where best replies are unique). Young (2009) uses a similar algorithm without explicit stages that also converges for generic n-player games. Rather than selecting best replies, in these algorithms agents choose new actions randomly when not in equilibrium. Unfortunately, these algorithms involve searching the whole strategy space, so their convergence time is exponential. Another algorithm that uses stages to provide a stable learning environment is the ESRL algorithm for coordinated exploration (Verbeeck, Nowé, Parent, & Tuyls, 2007).

Marden, Arslan, and Shamma (2007b) and Marden, Young, Arslan, and Shamma (2009) use an algorithm with experimentation and best replies but without explicit stages that converges for *weakly acyclic games*, where best-reply dynamics converge when agents move one at a time, rather than moving all at once, as we assume here. Convergence is based on the existence of a sequence of exploration moves that lead to equilibrium. With n agents who explore with probability ϵ, this analysis gives a convergence time of $O(1/\epsilon^n)$. Furthermore, the guarantee requires ϵ to be sufficiently small that agents essentially explore one at a time, so ϵ needs to be $O(1/n)$.

Adlakha, Johari, Weintraub, and Goldsmith (2010) have independently given conditions for the existence of an "oblivious equilibrium," or "mean field equilibrium," in stochastic games. Just as in our model they require that the game be large, anonymous, and continuous. In an oblivious equilibrium, each player reacts only to the "average" states and

2. Correlated equilibrium is a more general solution concept than Nash equilibrium (see Osborne & Rubenstein, 1994); every Nash equilibrium is a correlated equilibrium, but there may be correlated equilibria that are not Nash equilibria.

strategies of other players rather than their exact values. However, this model assumes that a player's payoff depends only on the state of other players and not their actions. Adlakha and Johari (2010) consider stochastic games with strategic complementarities and show that mean field equilibria exist, best-reply dynamics converge, and "myopic learning dynamics" (which require only knowledge of the aggregate states of other players) can find them.

There is a long history of work examining simple learning procedures such as *fictitious play* (Fudenberg & Levine, 1998), where each agent makes a best response assuming that each other player's strategy is characterized by the empirical frequency of his observed moves. In contrast to algorithms with convergence guarantees for general games, these algorithms fail to converge in many games. But for classes of games where they do converge, they tend to do so rapidly. However, most work in this area assumes that the actions of agents are observed by all agents, agents know the payoff matrix, and payoffs are deterministic. A recent approach in this tradition is based on the Win or Learn Fast principle, which has limited convergence guarantees but often performs well in practice (Bowling & Veloso, 2001). Hopkins (1999) showed that many such procedures converge in symmetric games with an infinite number of learners, although his results provide no guarantees about the rate of convergence.

There is also a body of empirical work on the convergence of learning algorithms in multiagent settings. Q-learning has had empirical success in pricing games (Tesauro & Kephart, 2002), n-player cooperative games (Claus & Boutilier, 1998), and grid world games (Bowling, 2000). Greenwald at al. (2001) showed that a number of algorithms, including stage learning, converge in a variety of simple games. Marden et al. (2009) found that their algorithm converged must faster in a congestion game than the theoretical analysis would suggest. Our theorem suggests an explanation for these empirical observations: best-reply dynamics converge in all these games. While our theorem applies directly only to stage learning, it provides intuition as to why algorithms that learn "quickly enough" and change their behavior "slowly enough" rapidly converge to Nash equilibrium in practice.

3. Theoretical Results

In this section we present the theoretical analysis of our model. We then provide support from simulations in the following section.

3.1 Large Anonymous Games

We are interested in anonymous games with countably many agents. Assuming that there are countably many agents simplifies the proofs; it is straightforward to extend our results to games with a large finite number of agents. Our model is adapted from that of Blonski (2001). Formally, a *large anonymous game* is characterized by a tuple $\Gamma = (\mathbb{N}, A, P, \Pr)$.

- $\mathbb{N}$ is the countably infinite set of agents.
- A is a finite set of actions from which each agent can choose (for simplicity, we assume that each agent can choose from the same set of actions).
- $\Delta(A)$, the set of probability distributions over A, has two useful interpretations. The first is as the set of mixed actions. For $a \in A$ we will abuse notation and denote the

mixed action that is a with probability 1 as a. In each round each agent chooses one of these mixed actions. The second interpretation of $\rho \in \Delta(A)$ is as the fraction of agents choosing each action $a \in A$. This is important for our notion of anonymity, which says an agent's utility should depend only on how many agents choose each action rather than who chooses it.

- $G = \{g : \mathbb{N} \to \Delta(A)\}$ is the set of (mixed) action profiles (i.e. which action each agent chooses). Given the mixed action of every agent, we want to know the fraction of agents that end up choosing action a. For $g \in G$, let $g(i)(a)$ denote the probability with which agent i plays a according to $g(i) \in \Delta(A)$. We can then express the fraction of agents in g that choose action a as $\lim_{n\to\infty}(1/n)\sum_{i=0}^{n} g(i)(a)$, if this limit exists. If the limit exists for all actions $a \in A$, let $\rho_g \in \Delta(A)$ give the value of the limit for each a. The profiles g that we use are all determined by a simple random process. For such profiles g, the strong law of large numbers (SLLN) guarantees that with probability 1 ρ_g is well defined. Thus it will typically be well defined (using similar limits) for us to talk about the fraction of agents who do something.

- $P \subset \mathbb{R}$ is the finite set of payoffs that agents can receive.

- $\Pr : A \times \Delta(A) \to \Delta(P)$ denotes the distribution over payoffs that results when the agent performs action a and other agents follow action profile ρ. We use a probability distribution over payoffs rather than a payoff to model the fact that agent payoffs may change even if no agent changes his strategy. The expected utility of an agent who performs mixed action s when other agents follow action distribution ρ is $u(s,\rho) = \sum_{a\in A}\sum_{p\in P} p s(a) \Pr_{a,\rho}(p)$. Our definition of Pr in terms of $\Delta(A)$ rather than G ensures the game is anonymous. We further require that Pr (and thus u) be *Lipschitz continuous*.[3] For definiteness, we use the L1 norm as our notion of distance when specifying continuity (the L1 distance between two vectors is the sum of the absolute values of the differences in each component). Note that this formulation assumes all agents share a common utility function. This assumption can be relaxed to allow agents to have a finite number of types, which we show in Appendix A.

An example of a large anonymous game is one where, in each round, each agent plays a two-player game against an opponent chosen at random. Such random matching games are common in the literature (e.g., Hopkins, 1999), and the meaning of "an opponent chosen at random" can be made formal (Boylan, 1992). In such a game, A is the set of actions of the two-player game and P is the set of payoffs of the game. Once every agent chooses an action, the distribution over opponent actions is characterized by some $\rho \in \Delta(A)$. Let $p_{a,a'}$ denote the payoff for the agent if he plays a and the other agent plays a'. Then the utility of mixed action s given distribution ρ is

$$u(s,\rho) = \sum_{a,a' \in A^2} s(a)\rho(a')p_{a,a'}.$$

3. Lipschitz continuity imposes the additional constraint that there is some constant K such that $|\Pr(a,\rho) - \Pr(a,\rho')|/||\rho - \rho'||_1 \le K$ for all ρ and ρ'. Intuitively, this ensures that the distribution of outcomes does not change "too fast." This is a standard assumption that is easily seen to hold in the games that have typically been considered in the literature.

3.2 Best-Reply Dynamics

Given a game Γ and an action distribution ρ, a natural goal for an agent is to play the action that maximizes his expected utility with respect to ρ: $\text{argmax}_{a \in A} u(a, \rho)$. We call such an action a *best reply* to ρ. In a practical amount of time, an agent may have difficulty determining which of two actions with close expected utilities is better, so we will allow agents to choose actions that are close to best replies. If a is a best reply to ρ, then a' is an *η-best reply* to ρ if $u(a', \rho) + \eta \geq u(a, \rho)$. There may be more than one η-best reply; we denote the set of η-best replies $ABR_\eta(\rho)$.

We do not have a single agent looking for a best reply; every agent is trying to find a one at the same time. If agents start off with some action distribution ρ_0, after they all find a best reply there will be a new action distribution ρ_1. We assume that $\rho_0(a) = 1/|A|$ (agents choose their initial strategy uniformly at random), but our results apply to any distribution used to determine the initial strategy. We say that a sequence $(\rho_0, \rho_1, \ldots)$ is an *η-best-reply sequence* if the support of ρ_{i+1} is a subset of $ABR_\eta(\rho_i)$; that is ρ_{i+1} gives positive probability only to approximate best replies to ρ_i. A η best-reply sequence *converges* if there exists some t such that for all $t' > t$, $\rho_{t'} = \rho_t$. Note that this is a particularly strong notion of convergence because we require the ρ_t to converge in finite time and not merely in the limit. A game may have infinitely many best-reply sequences, so we say that *approximate best-reply dynamics converge* if there exists some $\eta > 0$ such that every η-best-reply sequence converges. The limit distribution ρ_t determines a mixed strategy that is an η-Nash equilibrium (i.e. the support of ρ_t is a subset of $ABR_\eta(\rho_t)$).

Our main result shows that learners can successfully learn in large anonymous games where approximate best-reply dynamics converge. The number of stages needed to converge is determined by the number of best replies needed before the sequence converges. It is possible to design games that have long best-reply sequences, but in practice most games have short sequences. One condition that guarantees this is if ρ_0 and all the degenerate action distributions $a \in A$ (i.e., distributions that assign probability 1 to some $a \in A$) have unique best replies. In this case, there can be at most $|A|$ best replies before equilibrium is reached, because we have assumed that all agents have the same utility function. Furthermore, in such games the distinction between η-best replies and best replies is irrelevant; for sufficiently small η, a η-best reply is a best reply. It is not hard to show that the property that degenerate strategies have unique best replies is generic; it holds for almost every game.

3.3 Stage Learners

An agent who wants to find a best reply may not know the set of payoffs P, the mapping from actions to distributions over payoffs Pr, or the action distribution ρ (and, indeed, ρ may be changing over time), so he will have to use some type of learning algorithm to learn it. Our approach is to divide the play of the game into a sequence of stages. In each stage, the agent almost always plays some fixed action a, but also explores other actions. At the end of the stage, he chooses a new a' for the next stage based on what he has learned. An important feature of this approach is that agents maintain their actions for the entire stage, so each stage provides a stable environment in which agents can learn. To simplify our results, we specify a way of exploring and learning within a stage (originally described in Friedman & Shenker, 1998), but our results should generalize to any "reasonable" learning algorithm used to learn within a stage. (We discuss what is "reasonable" in Section 6.) In this section, we show that, given a suitable parameter, at each stage most agents will have learned a best reply to the environment of that stage.

Given a game Γ, in each round t agent i needs to select a mixed action $s_{i,t}$. Our agents use strategies that we denote a_ϵ, for $a \in A$, where $a_\epsilon(a) = 1 - \epsilon$ and $a_\epsilon(a' \neq a) = \epsilon/(|A| - 1)$. Thus, with a_ϵ, an agent almost always plays a, but with probability ϵ explores other strategies uniformly at random. Thus far we have not specified what information an agent can use to choose $s_{i,t}$. Different games may provide different information. All that we require is that an agent know all of his previous actions and his previous payoffs. More precisely, for all $t' < t$, he knows his action $a_{t'}(i)$ (which is determined by $s_{i,t'}$) and his payoffs $p_{t'}(i)$ (which is determined by $\Pr(a_{i,t'}, \rho_{t'})$, where $\rho_{t'}$ is the action distribution for round t'; note that we do not assume that the agent knows $\rho_{t'}$.) Using this information, we can express the average value of an action over the previous $\tau = \lceil 1/\epsilon^2 \rceil$ rounds (the length of a stage).[4] Let $H(a, i, t) = \{t - \tau \leq t' < t \mid a_{t'}(i) = a\}$ be the set of recent rounds in which a was played by i. Then the average value is $V(a, i, t) = \sum_{t' \in H(a,i,t)} p_{t'}(i)/|H(a, i, t)|$ if $|H(a, i, t)| > 0$ and 0 otherwise. While we need the value of H only at times that are multiples of τ, for convenience we define it for arbitrary times t.

We say that an agent is an *ϵ-stage learner* if he chooses his actions as follows. If $t = 0$, s_t is chosen at random from $\{a_\epsilon \mid a \in A\}$. If t is a nonzero multiple of τ, $s_{i,t} = a(i, t)_\epsilon$ where $a(i, t) = \text{argmax}_{a \in A} V(a, i, t)$. Otherwise, $s_{i,t} = s_{i,t-1}$. Thus, within a stage, his mixed action is fixed; at the end of a stage he updates it to use the action with the highest average value during the previous stage.

The evolution of a game played by stage learners is not deterministic; each agent chooses a random $s_{i,0}$ and the sequence of $a_t(i)$ and $p_t(i)$ he observes is also random. However, with a countably infinite set of agents, we can use the SLLN to make statements about the overall behavior of the game. Let $g_t(i) = s_{i,t}$. A *run* of the game consists of a sequence of triples (g_t, a_t, p_t). The SLLN guarantees that with probability 1 the fraction of agents who choose a strategy a in a_t is $\rho_{g_t}(a)$. Similarly, the fraction of agents who chose a in a_t that receive payoff p will be $\Pr(a, \rho_{g_t})(p)$ with probability 1.

To make our notion of a stage precise, we refer to the sequence of tuples $(g_{n\tau}, a_{n\tau}, p_{n\tau}) \dots (g_{(n+1)\tau-1}, a_{(n+1)\tau-1}, p_{(n+1)\tau-1})$ as stage n of the run. During stage n there is a stationary action distribution that we denote $\rho_{g_{n\tau}}$. If $s_{i,(n+1)\tau} = a_\epsilon$ and $a \in ABR_\eta(g_{n\tau})$, then we say that agent i has *learned an η-best reply* during stage n of the run. As the following lemma shows, for sufficiently small ϵ, most agents will learn an η-best reply.

Lemma 3.1. *For all large anonymous games Γ, action profiles, approximations $\eta > 0$, and probabilities of error $e > 0$, there exists an $\epsilon^* > 0$ such that for $\epsilon < \epsilon^*$ and all n, if all agents are ϵ-stage learners, then at least a $1 - e$ fraction of agents will learn an η-best reply during stage n.*

Proof. (Sketch) On average, an agent using strategy a_ϵ plays action a $(1 - \epsilon)\tau$ times during a stage and plays all other actions $\epsilon\tau/(n - 1)$ times each. For τ large, the realized number of times played will be close to the expectation value with high probability. Thus, if $\epsilon\tau$ is sufficiently large, then the average payoff from each action will be exponentially close to the

4. The use of the exponent 2 is arbitrary. We require only that the expected number of times a strategy is explored increases as ϵ decreases.

true expected value (via a standard Hoeffding bound on sums of i.i.d. random variables), and thus each learner will correctly identify an action with approximately the highest expected payoff with probability at least $1-e$. By the SLLN, at least a $1-e$ fraction of agents will learn an η-best reply. A detailed version of this proof in a more general setting can be found in the work by Friedman and Shenker (1998). □

3.4 Convergence Theorem

Thus far we have defined large anonymous games where approximate best-reply dynamics converge. If all agents in the game are ϵ-stage learners, then the sequence $\hat{\rho}_0, \hat{\rho}_1, \ldots$ of action distributions in a run of the game is not a best-reply sequence, but it is close. The action used by most agents most of the time in each $\hat{\rho}_n$ is the action used in ρ_n for some approximate best reply sequence.

In order to prove this, we need to define "close." Our definition is based on the error rate e and exploration rate ϵ that introduces noise into $\hat{\rho}_n$. Intuitively, distribution $\hat{\rho}$ is close to ρ if, by changing the strategies of an e fraction of agents and having all agents explore an ϵ fraction of the time, we can go from an action profile with corresponding action distribution ρ to one with corresponding distribution $\hat{\rho}$. Note that this definition will not be symmetric.

In this definition, g identifies what (pure) action each agent is using that leads to ρ, g' allows an e fraction of agents to use some other action, and $\hat{g}$ incorporates the fact that each agent is exploring, so each strategy is an a_ϵ (the agent usually plays a but explores with probability ϵ).

Definition 3.2. Action distribution $\hat{\rho}$ is (e, ϵ)-*close* to ρ if there exist g, g', and $\hat{g} \in G$ such that:

- $\rho = \rho_g$ and $\hat{\rho} = \rho_{\hat{g}}$;
- $g(i) \in A$ for all $i \in \mathbb{N}$;
- $||\rho_g - \rho_{g'}||_1 \leq 2e$ (this allows an e fraction of agents in g' to play a different strategy from g);
- for some $\epsilon' \leq \epsilon$, if $g'(i) = a$ then $\hat{g}(i) = a_{\epsilon'}$. □

The use of ϵ' in the final requirement ensures that if two distributions are (e, ϵ)-close then they are also (e', ϵ')-close for all $e' \geq e$ and $\epsilon' \geq \epsilon$. As an example of the asymmetry of this definition, a_ϵ is $(0, \epsilon)$ close to a, but the reverse is not true. While (e, ϵ)-closeness is a useful distance measure for our analysis, it is an unnatural notion of distance for specifying the continuity of u, where we used the L1 norm. The following simple lemma shows that this distinction is unimportant; if $\hat{\rho}$ is sufficiently (e, ϵ)-close to ρ then it is close according to the L1 measure as well.

Lemma 3.3. *If $\hat{\rho}$ is (e, ϵ)-close to ρ, then $||\hat{\rho} - \rho||_1 \leq 2(e + \epsilon)$.*

Proof. Since $\hat{\rho}$ is (e, ϵ)-close to ρ, there exist g, g', and $\hat{g}$ as in Definition 3.2. Consider the distributions $\rho_g = \rho$, $\rho_{g'}$, and $\rho_{\hat{g}} = \hat{\rho}$. We can view these three distributions as vectors, and calculate their L1 distances. By Definition 3.2, $||\rho_g - \rho_{g'}||_1 \leq 2e$. $||\rho_{g'} - \rho_{\hat{g}}||_1 \leq 2\epsilon$ because an ϵ fraction of agents explore. Thus by the triangle inequality, the L1 distance between ρ and $\hat{\rho}$ is at most $2(e + \epsilon)$. □

We have assumed that approximate best reply sequences of ρ_n converge, but during a run of the game agents will actually be learning approximate best replies to $\hat{\rho}_n$. The following lemma shows that this distinction does not matter if ρ and $\hat{\rho}$ are sufficiently close.

Lemma 3.4. *For all η there exists a d_η such that if $\hat{\rho}$ is (e,ϵ)-close to ρ, $e > 0$, $\epsilon > 0$, and $e + \epsilon < d_\eta$ then $ABR_{(\eta/2)}(\hat{\rho}) \subseteq ABR_\eta(\rho)$.*

Proof. Let K be the maximum of the Lipschitz constants for all $u(a, \cdot)$ (one constant for each a) and $d_\eta = \eta/(8K)$. Then for all $\hat{\rho}$ that are (e,ϵ)-close to ρ and all a, $|u(a,\hat{\rho}) - u(a,\rho)| \leq ||\hat{\rho} - \rho||_1 K \leq 2\eta/(8K)K = \eta/4$ by Lemma 3.3.

Let $a \notin ABR_\eta(\rho)$ and $a' \in \text{argmax}_{a'}\, u(a', \rho)$. Then $u(a,\rho) + \eta < u(a',\rho)$. Combining this with the above gives $u(a,\hat{\rho}) + \eta/2 < u(a',\hat{\rho})$. Thus $a \notin ABR_{\eta/2}(\hat{\rho})$. □

Lemmas 3.1 and 3.4 give requirements on (e,ϵ). In the statement of the theorem, we call (e,ϵ) *η-acceptable* if they satisfy the requirements of both lemmas for $\eta/2$ and all η-best-reply sequences converge in Γ.

Theorem 3.5. *Let Γ be a large anonymous game where approximate best-reply dynamics converge and let (e,ϵ) be η-acceptable for Γ. If all agents are ϵ-stage learners then, for all runs, there exists an η-best-reply sequence $\rho_0, \rho_1, \ldots$ such that in stage n at least a $1 - e$ fraction will learn a best reply to ρ_n with probability 1.*

Proof. $\rho_0 = \hat{\rho}_0$ (both are the uniform distribution), so $\hat{\rho}_0$ is (e,ϵ)-close to ρ. Assume $\hat{\rho}_n$ is (e,ϵ)-close to ρ. By Lemma 3.1 at least a $1-e$ fraction will learn a $\eta/2$-best reply to $\hat{\rho}_n$. By Lemma 3.4, this is a η-best reply to ρ_n. Thus $\hat{\rho}_{n+1}$ will be (e,ϵ)-close to ρ_{n+1}. □

Theorem 3.5 guarantees that after a finite number of stages, agents will be close to an approximate Nash equilibrium profile. Specifically, $\hat{\rho}_n$ will be (e,ϵ)-close to an η-Nash equilibrium profile ρ_n. Note that this means that $\hat{\rho}_n$ is actually an η'-Nash equilibrium for a larger η' that depends on η,e,ϵ, and the Lipschitz constant K.

Our three requirements for a practical learning algorithm were that it require minimal information, converge quickly in a large system, and be robust to noise. Stage learning requires only that an agent know his own payoffs, so the first condition is satisfied. Theorem 3.5 shows that it satisfies the other two requirements. Convergence is guaranteed in a finite number of stages. While the number of stages depends on the game, in Section 3.2 we argued that in many cases it will be quite small. Finally, robustness comes from tolerating an e fraction of errors. While in our proofs we assumed these errors were due to learning, the analysis is the same if some of this noise is from other sources such as churn or agents making errors. We discuss this issue more in Section 6.

4. Simulation Results

In this section, we discuss experimental results that demonstrate the practicality of learning in large anonymous games. Theorem 3.5 guarantees convergence for a sufficiently small exploration probability ϵ, but decreasing ϵ also increases τ, the length of a stage. Our first set of experiments shows that the necessary values of ϵ and τ are quite reasonable in practice. While our theorem applies only to stage learning, the analysis provides intuition as to why a reasonable algorithm that changes slowly enough that other learners have a chance to learn best replies should converge as well. To demonstrate this, we also implemented two other learning algorithms, which also quickly converged.

Our theoretical results make two significant predictions about factors that influence the rate of convergence. Lemma 3.1 tells us that the length of a stage is determined by the number of times each strategy needs to be explored to get an accurate estimate of its value. Thus, the amount of information provided by each observation has a large effect on the rate of convergence. For example, in a random matching game, an agent's payoff provides information about the strategy of one other agent. On the other hand, if he receives his expected payoff for being matched, a single observation provides information about the entire distribution of strategies. In the latter case the agent can learn with many fewer observations. A related prediction is that having more agents will lead to faster convergence, particularly in games where payoffs are determined by the average behavior of other agents, because variance in payoffs due to exploration and mistakes decreases as the number of agents increases. Our experimental results illustrate both of these phenomena.

The game used in our first set of experiments, like many simple games used to test learning algorithms, is symmetric. Hopkins (1999) showed that many learning algorithms are well behaved in symmetric games with large populations. To demonstrate that our main results are due to something other than symmetry, we also tested stage learning on an asymmetric game, and observed convergence even with a small population.

To explore the applicability of stage learning in a more practical setting that violates a number of the assumptions of our theorem, we implemented a variant of stage learning for a game based on a scrip system (Kash et al., 2007). To demonstrate the applicability of this approach to real systems, we included experiments where there is churn (agents leaving and being replaced by new agents) and agents learning at different rates.

Finally, we give examples of games that are not large, not anonymous, and not continuous, and provide simulations showing that stage learners learn far more slowly in these games than in those that satisfy the hypotheses of Theorem 3.5, or do not learn to play equilibrium at all. These examples demonstrate that these assumptions are essential for our results.

4.1 A Contribution Game

In our first set of experiments, agents play a contribution game (also called a Diamond-type search model in the work by Milgrom & Roberts, 1990). In the contribution game, two agents choose strategies from 0 to 19, indicating how much effort they contribute to a collective enterprise. The value to an agent depends on how much he contributes, as well as how much the other agent contributes. If he contributes x and the contribution of the other agent is y, then his utility is $4xy - (x-5)^3$. In each round of our game, each agent is paired with a random agent and they play the contribution game. In this game, best-reply dynamics converge within 4 stages from any starting distribution.

We implemented three learning algorithms to run on this game. Our implementation of stage learners is as described in Section 3.3, with $\epsilon = 0.05$. Rather than taking the length of stage τ to be $1/\epsilon^2$, we set $\tau = 2500$ to have sufficiently long stages for this value of ϵ, rather than decreasing ϵ until stages are long enough. Our second algorithm is based on that of

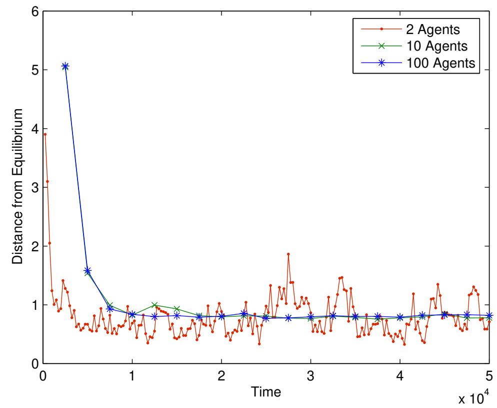

Figure 1: Stage learners with random matching.

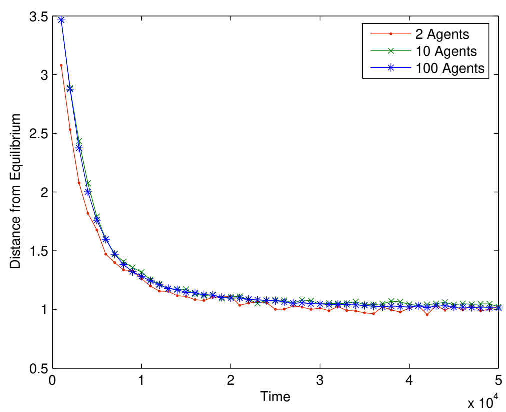

Figure 2: Hart and Mas-Colell with random matching.

Hart and Mas-Colell (2001), with improvements suggested by Greenwald, Friedman, and Shenker (2001). This algorithm takes parameters M and δ (the exploration probability). We used $M = 16$ and $\delta = 0.05$. Our final learning algorithm is Exp3 (Auer et al., 2002). We set γ, the exploration probability, to 0.05. This algorithm requires that payoffs be normalized to lie in $[0, 1]$. Since a few choices of strategies lead to very large negative payoffs, a naive normalization leads to almost every payoff being close to 1. For better performance, we normalized payoffs such that most payoffs fell into the range $[0, 1]$ and any that were outside were set to 0 or 1 as appropriate.

The results of these three algorithms are shown in Figures 1, 2, and 3. Each curve shows the distance from equilibrium as a function of the number of rounds of a population of agents of a given size using a given learning algorithm. The results were averaged over ten runs. Since the payoffs for nearby strategies are close, we want our notion of distance to take into account that agents playing 7 are closer to equilibrium (8) than those playing zero. Therefore, we consider the expected distance of ρ from equilibrium: $\sum_a \rho(a)|a - 8|$. To determine ρ, we counted the number of times each action was taken over the length of

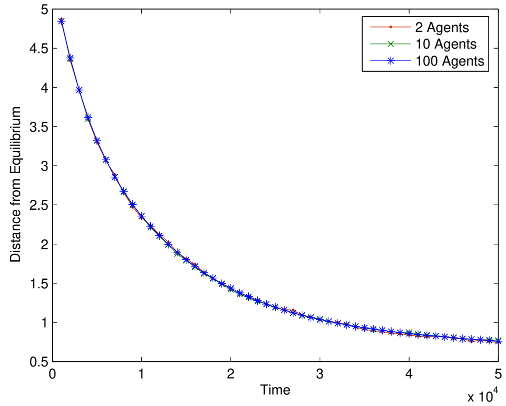

Figure 3: Exp3 with random matching.

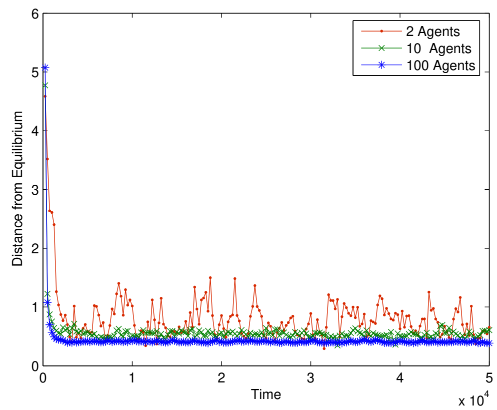

Figure 4: Stage learning with average-based payoffs.

a stage, so in practice the distance will never be zero due to mistakes and exploration. For ease of presentation, the graph shows only populations of size up to 100; similar results were obtained for populations up to 5000 agents.

For stage learning, increasing the population size has a dramatic impact. With two agents, mistakes and best replies to the results of these mistakes cause behavior to be quite chaotic. With ten agents, agents successfully learn, although mistakes and suboptimal strategies are quite frequent. With one hundred agents, all the agents converge quickly to near equilibrium strategies and significant mistakes are rare.

Despite a lack of theoretical guarantees, our other two algorithms also converge, although somewhat more slowly. The long-run performance of Exp3 is similar to stage learning. Hart and Mas-Colell's algorithm only has asymptotic convergence guarantees, and tends to converge slowly in practice if tuned for tight convergence. So to get it to converge in a reasonable amount of time we tuned the parameters to accept somewhat weaker convergence (although for the particular game shown here the difference in convergence is not dramatic).

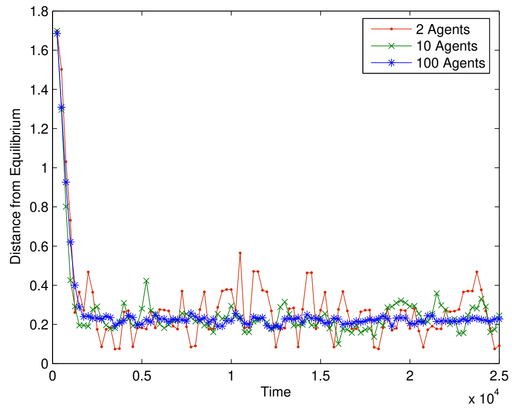

Figure 5: Stage learners in a congestion game.

Convergence of stage learning in the random-matching game takes approximately 10,000 rounds, which is too slow for many applications. If a system design requires this type of matching, this makes learning problematic. However, the results of Figure 4 suggest that the learning could be done much faster if the system designer could supply agents with more information. This suggests that collecting statistical information about the behavior of agents may be a critical feature for ensuring fast convergence. To model such a scenario, consider a related game where, rather than being matched against a random opponent, all agents contribute to the same project and their reward is based on the average contribution of the other agents. The results of stage learning in this game are shown in Figure 4. With so much more information available to agents from each observation, we were able to cut the length of a stage by a factor of 10. The number of stages needed to reach equilibrium remained essentially the same. Convergence was tighter as well; mistakes were rare and almost all of the distance from equilibrium is due to exploration.

4.2 A Congestion Game

For a different game, we tested the performance of stage learners in a congestion game. This game models a situation where two agents share a network link. They gain utility proportional to their transmission rate over the link, but are penalized based on the resulting congestion they experience. The game is asymmetric because the two different types of agents place different values on transmission rate. The game is described in detail by Greenwald, Friedman, and Shenker (2001), who showed that no-regret learners are able to find the equilibrium of this game. An extension of our theoretical results to games with multiple types is presenting in Appendix A.

Figure 5 shows that stage learners were able to learn very quickly in this game, using stages of length 250 even though they were being randomly matched against a player of the other type. Because the different types of agents had different equilibrium strategies, the distance measure we use is to treat the observed distribution of strategies and the equilibrium distribution as vectors and compute their L1 distance.

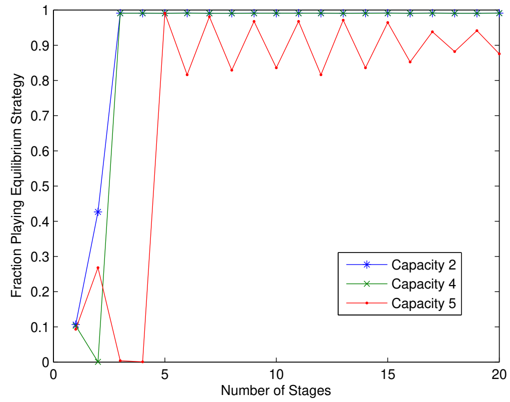

Figure 6: Stage learners in a TCP-like game.

4.3 A TCP-like Game

In the previous example, we considered a random-matching game where two agents of different types share a link. We now consider a game where a large number of agents share several links (this game is a variant of the congestion control game studied in Nisan et al., 2011).

There are three types of agents using a network. Each agent chooses an integer rate at which to transmit between 0 and 10. Links in the network have a maximum average rate at which agents can transmit; if this is exceeded they share the capacity evenly among agents. An agent's utility is his overall transmission rate through the network minus a penalty for traffic that was dropped due to congestion. If an agent attempts to transmit at a rate of x and has an actual rate of y the penalty is $0.5(x-y)$.[5]

All agents share a link with an average capacity of 5. One third of agents are further constrained by sharing a link with an average capacity of 2 and another third share a link with average capacity of 4. This game has the unique equilibrium where agents in the first third choose a rate of 2, agents in the second third choose a rate of 4, and agents in the final third choose a rate of 9 (so that the overall average rate is 5). This results in a game where best-reply dynamics converge in five stages from a uniform starting distribution.

Figure 6 shows the results for 90 learners (30 of each type) with $\tau = 50000$ and $\epsilon = 0.01$, averaged over ten runs. Agents constrained by an average capacity of two quickly learn their equilibrium strategy, followed by those with an average capacity of four. Agents constrained by an average capacity of five learn their equilibrium strategy, but have a "sawtooth" pattern where a small fraction alternately plays 10 rather than 9. This is because, with exploration, it is actually optimal for a small number of agents to play 10. Once a noticeable fraction does so, 9 is uniquely optimal. This demonstrates that, strictly speaking, this game does not satisfy our continuity requirement. In equilibrium, the demand for bandwidth is exactly equal to the supply. Thus, small changes in the demand of other agents due to exploration can have a large effect on the amount that can actually be demanded and thus on the payoffs

5. This penalty is not used in the work by Nisan et al. (2011); using it avoids the tie-breaking issues they consider.

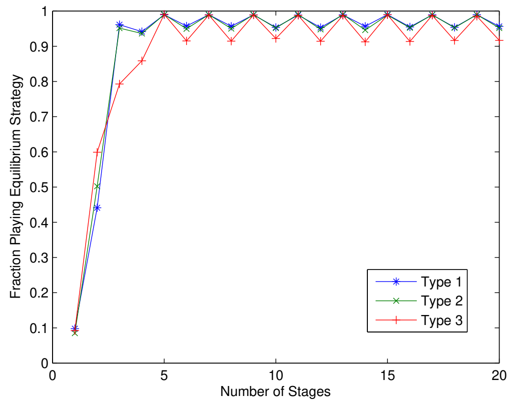

Figure 7: Stage learners in random TCP-like games.

of various strategies. However, the structure of the game is such that play still tends to remain "close" to the equilibrium in terms of the rates agents choose.

In addition to the specific parameters mentioned above, we also ran 100 simulations where each of the three capacities was a randomly chosen integer between 0 and 10. Figure 7 shows that, on average, the results were similar. All three types of agents share a common constraint; type 1 and type 2 each have an additional constraint. Unsurprisingly, since these two types are symmetric their results are almost identical. All three types demonstrate the sawtooth behavior, with type 3 doing so in more runs due to examples like Figure 6 where having fewer constraints gives agents more flexibility. This primarily comes from runs where type 1 and type 2 have constraints that are larger than the overall constraint (i.e. only the overall constraint matters). Thus all three types have the ability to benefit from resources not demanded when other agents explore.

4.4 A Scrip System Game

Our motivation for this work is to help the designers of distributed systems understand when learning is practical. In order to demonstrate how stage learning could be applied in such a setting, we tested a variant of stage learners in the model of a scrip system used by Kash et al. (2007). In the model, agents pay other agents to provide them service and in turn provide service themselves to earn money to pay for future service. Agents may place different values on receiving service (γ), incur different costs to provide service (α), discount future utility at different rates (δ), and have different availabilities to provide service (β). We used a single type of agent with parameters $\gamma = 1.0$, $\alpha = 0.05$, $\delta = 0.9$, $\beta = 1$, average amount of money per agent $m = 1$, and stages of 200 rounds per agent (only one agent makes a request each round).

This model is not a large anonymous game because whether an agent should provide service depends on how much money he currently has. Thus, stage learning as specified does not work, because it does not take into account the current state of the (stochastic) game. Despite this, we can still implement a variant of stage learning: fix a strategy during each stage and then at the end of the stage use an algorithm designed for this game to determine

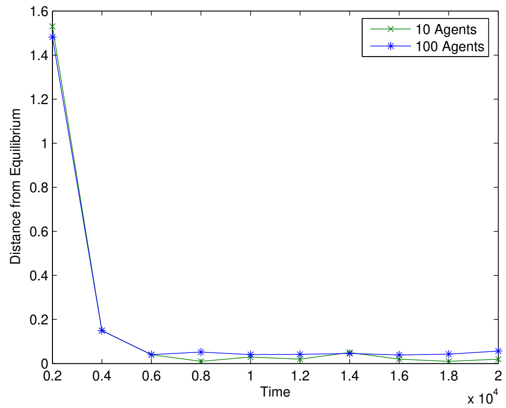

Figure 8: Stage learners in a scrip system.

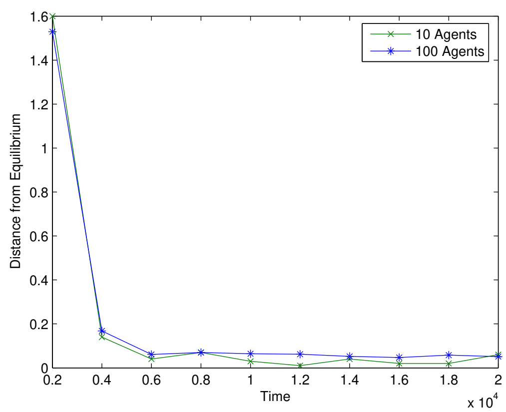

Figure 9: A scrip system with churn.

a new strategy that is a best reply to what the agent observed. Our algorithm works by estimating the agent's probabilities of making a request and being chosen as a volunteer in each round, and then uses these probabilities to compute an optimal policy. Figure 8 shows that this is quite effective. The distance measure used is based on directly measuring the distance of the agent's chosen (threshold) strategy from the equilibrium strategy, since unlike the previous games it is impossible to directly infer the agent's strategy in each round solely from his decision whether or not to volunteer. Note that the number of rounds has been normalized based on the number of agents in Figure 8 and later figures; stages actually lasted ten times as long with 100 agents.

Real systems do not have a static population of learning agents. To demonstrate the robustness of stage learning to churn, we replaced ten percent of the agents with new agents with randomly chosen initial strategies at the end of each period. As Figure 9 shows, this has essentially no effect on convergence.

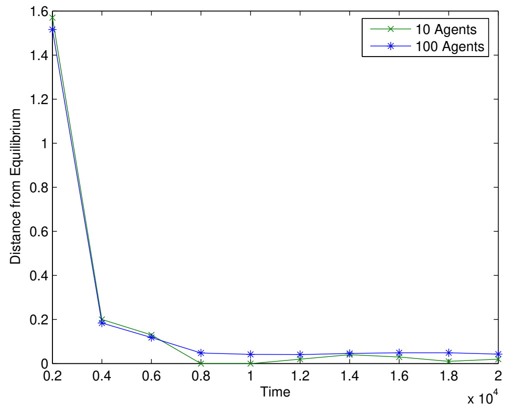

Figure 10: A scrip system with different stage lengths.

Finally, in a real system it is often unreasonable to expect all agents to be able to update their strategies at the same time. Figure 10 shows that having half the agents use stages of 222 rounds per agent rather than 200 did not have a significant effect on convergence.[6]

4.5 Learning Counterexamples

At first glance, Theorem 3.5 may seem trivial. In a game where best-reply dynamics are guaranteed to converge, it seems obvious that agents who attempt to find best replies should successfully find them and reach equilibrium. However, as we show in this section, this fact alone is not sufficient. In particular, all three of the key features of the games we study—that they are large, anonymous, and continuous—are required for the theorem to hold.

First, if the game has only a small number of agents, a mistake made by a single agent could be quite important, to the point where learning essentially has to start over. So, while our results can be converted into results about the probability that none of a finite number of agents will make a mistake in a given stage, the expected time to reach equilibrium following this algorithm can be significantly longer than the best-reply dynamics would suggest. The following is an example of a game where the number of best replies needed to reach equilibrium is approximately the number of strategies, but our experimental results show that the number of stages needed by stage learners to find the equilibrium is significantly longer. (We conjecture that in fact the learning time is exponentially longer.) In contrast, Theorem 3.5 guarantees that, for games satisfying our requirements, the number of stages needed is equal to the number of best replies.

Consider a game with three agents, where A, the set of actions, is $\{0, 1, \ldots, k\}$. The utility functions of the agents are symmetric; the first agent's utility function is given by the following table:

6. In general we expect that small variations in stage lengths will not affect convergence; however large enough differences can result in non-Nash convergence. See the work by Greenwald et al. (2001) for some simulations and analysis.

actions	payoff	conditions
$(0, y, z)$	1	if $y \neq z$ and either $y > 1$ or $z > 1$
(x, y, z)	0	if $y \neq z$ and $x > 0$ and either $y > 1$ or $z > 1$
$(0, 1, 0)$	0	
$(0, 0, 1)$	0	
$(1, 1, 0)$	1	
$(1, 0, 1)$	1	
(x, y, y)	1	if $x = y + 1$
(x, y, y)	0	if $x \neq y + 1$ and $y < k$
(k, k, k)	1	
(x, k, k)	0	if $x \neq k$

Agents learning best replies can be viewed as "climbing a ladder." The best reply to (x, x, x) is $(x + 1, x + 1, x + 1)$ until agents reach (k, k, k), which is a Nash equilibrium. However, when a mistake is made, agents essentially start over. To see how this works, suppose that agents are at $(3, 3, 3)$ and for the next stage one makes a mistake and they select $(5, 4, 4)$. This leads to the best reply sequence $(5, 0, 0)$, $(1, 0, 0)$, $(1, 1, 1)$, at which point agents can begin climbing again. The somewhat complicated structure of payoffs near 0 ensures that agents begin climbing again from arbitrary patterns of mistakes. In a typical run, $k+2$ stages of best replies are needed to reach equilibrium: one stage with the initial randomly-chosen strategies, one stage where all three agents switch to strategy 0, and k stages of climbing. The exact number of stages can vary if two or more agents choose the same initial strategy, but can never be greater than $k + 3$.

The following table gives the number of rounds (averaged over ten runs) for stage learners in this game to first reach equilibrium. As the number of strategies varies, the length of a stage is $\tau = 100(k + 1)$, with exploration probability $\epsilon = 0.05$.

k	rounds to reach k
4	7.0
9	19.3
14	25.8
19	39.5
24	37.3
29	102.7
34	169.4
39	246.6

With $k = 4$, stage learners typically require $k+2$ stages, with an occasional error raising the average slightly. With k between 9 and 24, a majority of runs feature at least one agent making a mistake, so the number of stages required is closer to $2k$. With $k = 29$ and up, there are many opportunities for agents to make a mistake, so the number of stages required on average is in the range of $3k$ to $6k$. Thus learning is slower than best-reply dynamics, and the disparity grows as the number of strategies increases.

A small modification of this example shows the problems that arise in games that are not anonymous. In a non-anonymous game with a large number of agents, payoffs can depend entirely on the actions of a small number of agents. For example, we can split the set N of agents into three disjoint sets, N_0, N_1, and N_2, and choose agents $0 \in N_0$, $1 \in N_1$,

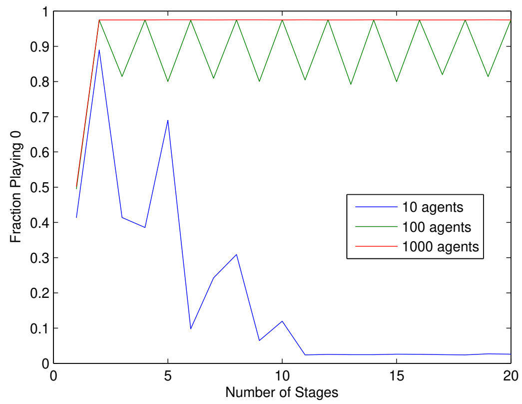

Figure 11: Stage learners in a discontinuous game.

and $2 \in N_2$. Again, each agent chooses an action in $\{0, \dots, k\}$. The payoffs of agents 0, 1, and 2 are determined as above; everyone in N_0 gets the same payoff as 0, everyone in N_1 gets the same payoff as 1, and everyone in N_2 gets the same payoff as 2. Again, convergence to equilibrium will be significantly slower than with best-reply dynamics.

Finally, consider the following game, which is large and anonymous, but does not satisfy the continuity requirement. The set of actions is $A = \{0, 1\}$, and each agent always receives a payoff in $P = \{0, 1, 10\}$. If an agent chooses action 0, his payoff is always 1 ($\Pr_{0,\rho}(1) = 1$). If he chooses action 1, his payoff is 10 if every other agent chooses action 1, 10 if every other agent chooses action 0, and 0 otherwise ($\Pr_{1,(1,0)}(10) = 1$, $\Pr_{1,(0,1)}(10) = 1$, and $\Pr_{1,\rho}(0) = 1$) for $\rho \notin \{(1,0), (0,1)\}$).

In this game, suppose approximate best-reply dynamics start at $(0.5, 0.5)$ (each action is chosen by half of the agents). As they have not coordinated, the unique approximate best reply for all agents is action 0, so after one best reply, the action distribution will be $(1, 0)$. Since agents have now coordinated, another round of approximate best replies leads to the equilibrium $(0, 1)$. If the agents are stage learners, after the first stage they will learn an approximate best reply to $(0.5, 0.5)$ (exploration does not change the action profile in this case), so most will adopt the mixed action 0_ϵ: playing 0 with probability $1 - \epsilon$ and 1 with probability ϵ. Thus, even if no agents make a mistake, the action distribution for the next stage will have at least an ϵ fraction playing action 1. Thus the unique approximate best reply will be action 0; stage learners will be "stuck" at 0, and never reach the equilibrium of 1.

Figure 11 shows the fraction of times strategy 0 was played during each stage (averaged over ten runs) for 10, 100, and 1000 agents ($\tau = 100$ and $\epsilon = 0.05$). With ten agents, some initial mistakes are made, but after stage 10 strategy 0 was played about 2.5% of the time in all runs, which corresponds to the fraction of time we expect to see it simply from exploration. With 100 agents we see another "sawtooth" pattern where most agents are stuck playing 0, but in alternating rounds a small fraction plays 1. This happens because, in rounds where all are playing 0, a small fraction are lucky and explore 1 when no other agents explore. As a result, they adopt strategy 1 for the next stage. However, most do not, so in the following stage agents return to all playing 0. Such oscillating behavior has been observed in other learning contexts, for example among competing myopic pricebots (Kephart, Hanson, & Greenwald, 2000). With 1000 agents, such lucky agents are quite rare, so essentially all agents are constantly stuck playing 0.

5. Learning with Byzantine Agents

In practice, learning algorithms need to be robust to the presence of other agents not following the algorithm. We have seen that stage learning in large anonymous games is robust to agents who do not learn and instead follow some fixed strategy in each stage. In the analysis, these agents can simply be treated as agents who made a mistake in the previous stage. However, an agent need not follow some fixed strategy; an agent attempting to interfere with the learning of other for malicious reasons or personal gain will likely adapt his strategy over time. However, as we show in this section, stage learning can also handle such manipulation in large anonymous games.

Gradwohl and Reingold (2008) examined several classes of games and introduced the notion of a "stable" equilibrium as one in which a change of strategy by a small fraction of agents only has a small effect on the payoff of other agents. Their definition is for games with a finite number of agents, but it can easily be adapted to our notion of a large anonymous game. We take this notion a step further and characterize the game, rather than an equilibrium, as stable if every strategy is stable.

Definition 5.1. A large anonymous game Γ is (η, β)-stable if for all $\rho, \rho' \in \Delta(A)$ such that $||\rho - \rho'||_1 \leq \beta$ and all $a \in A$, $|u(a, \rho) - u(a, \rho')| \leq \eta$. □

One class of games they consider is λ-continuous games. λ-continuity is essentially a version of Lipschitz continuity for finite games, so it easy to show that large anonymous games are stable, where the amount of manipulation that can be tolerated depends on the Lipschitz constants for agents' utility functions.

Lemma 5.2. *For all large anonymous games Γ, there exists a constant K_Γ such that for all η, Γ is $(\eta, \eta/K_\Gamma)$-stable*

Proof. For all a, $u(a, \cdot)$ is Lipschitz continuous with a constant K_a such that $|u(a, \rho) - u(a, \rho')|/||\rho - \rho'||_1 \leq K_a$. Take $K_\Gamma = \max_a K_a$. Then for all ρ and ρ' such that $||\rho - \rho'||_1 \leq \eta/K_\Gamma$,

$$\begin{aligned}
|u(a, \rho) - u(a, \rho')| &\leq K_a ||\rho - \rho'||_1 \\
&\leq K_\Gamma ||\rho - \rho'||_1 \\
&\leq (K_\Gamma) \left(\frac{\eta}{K_\Gamma} \right) \\
&\leq \eta.
\end{aligned}$$

□

Gradwohl and Reingold (2008) show that stable equilibria have several nice properties. If a small fraction of agents deviate, payoffs for the other agent will not decrease very much relative to equilibrium. Additionally, following those strategies will still be an approximate equilibrium despite the deviation. Finally, this means that the strategies still constitute an approximate equilibrium even if asynchronous play causes the strategies of a fraction of agents to be revealed to others.

We show that, if the game is stable, then learning is also robust to the actions of a small fraction of Byzantine agents. The following lemma adapts Lemma 3.1 to show that, in each stage, agents can learn approximate best replies despite the actions of Byzantine agents. Thus agents can successfully reach equilibrium, as shown by Theorem 3.5.

To state the lemma, we need to define the actions of a Byzantine agent. If there were no Byzantine agents, then in stage n there would be some stationary strategy $g^*_{n\tau}$ and corresponding fraction of agents choosing each action $\rho^*_{n\tau}$. A β fraction of Byzantine agents can change their actions arbitrarily each round, but doing so will have no effect on the actions of the other agents. Thus, the Byzantine agents can cause the observed fraction of agents choosing each strategy in round t to be any ρ_t such that $||\rho^*_{n\tau} - \rho_t||_1 < 2\beta$. We refer to a sequence $\rho_{n\tau}, \ldots, \rho_{n(\tau+1)-1}$ such that this condition holds for each t as a *consistent sequence*. When we say that agents learn an η-best reply during stage n, we mean that the strategy that they learn is an approximate best reply to $g^*_{n\tau}$, the actions that the players would have used had there been no Byzantine players, not the actual action profile, which includes the strategies used by the Byzantine players.

Lemma 5.3. *For all large anonymous games Γ, action distributions $\rho_{g_{\tau n}}$, approximations $\eta > 0$, probabilities of error $e > 0$, and fractions of agents $\beta < \eta/6K_\Gamma$, there exists an $\epsilon^* > 0$ such that for $\epsilon < \epsilon^*$, all n, and all consistent sequences $\rho_{n\tau}, \ldots, \rho_{n(\tau+1)-1}$, if all agents are ϵ-stage learners, then at least a $1-e$ fraction of agents will learn an η-best reply during stage n despite a β fraction of Byzantine agents.*

Proof. Consider an agent i and round t in stage n. If all agents were stage learners then the action distribution would be $\rho^* = \rho_{g_{\tau n}}$. However, the Byzantine agents have changed it such that $||\rho^* - \rho_t|| \leq 2\beta$. Fix an action a. By Lemma 5.2,

$$|u(a, \rho^*) - u(a, \rho_t)| \leq K_\Gamma 2\beta < \frac{2\eta}{6K_\Gamma} K_\Gamma = \frac{\eta}{3}$$

This means that Byzantine agents can adjust an agent's expected estimate of the value of an action by at most $\eta/3$. Let a^* be a best reply to $g_{\tau n}$ (the action used by stage learners during stage n). In each round t of stage n,

$$u(a^*, \rho^*) - u(a^*, \rho_t) < \frac{\eta}{3}.$$

For any action a that is not an η-best reply,

$$u(a^*, \rho^*) - u(a, \rho_t) = (u(a^*, \rho^*) - u(a, \rho^*) + (u(a, \rho^*) - u(a, \rho_t)) > \eta - \frac{\eta}{3} = \frac{2\eta}{3}.$$

Thus, regardless of the actions of the β fraction of Byzantine agents, agent i's expected estimate of the value of a^* exceeds his expected estimate of the value of a by at least $\eta/3$. Using Hoeffding bounds as before, for sufficiently large ϵ, i's estimates will be exponentially close to these expectations, so with probability at least $1-e$, he will not select as best any action that is not an η-best reply. By the SLLN, this means that at least a $1-e$ fraction of agents will learn an η-best reply. $\square$

Thus, as Lemma 5.3 shows, not only can stage learners learn despite some agents learning incorrect values, they can also tolerate a sufficiently small number of agents behaving arbitrarily.

6. Discussion

While our results show that a natural learning algorithm can learn efficiently in an interesting class of games, there are many further issues that merit exploration.

6.1 Other Learning Algorithms

Our theorem assumes that agents use a simple rule for learning within each stage: they average the value of payoffs received. However, there are certainly other rules for estimating the value of an action; any of these can be used as long as the rule guarantees that errors can be made arbitrarily rare given sufficient time. It is also not necessary to restrict agents to stage learning. Stage learning guarantees a stationary environment for a period of time, but such strict behavior may not be needed or practical. Other approaches, such as exponentially discounting the weight of observations (Greenwald et al., 2001; Marden et al., 2009) or Win or Learn Fast (Bowling & Veloso, 2001) allow an algorithm to focus its learning on recent observations and provide a stable environment in which other agents can learn.

6.2 Other Update Rules

In addition to using different algorithms to estimate the values of actions, a learner could also change the way he uses those values to update his behavior. For example, rather than basing his new strategy on only the last stage, he could base it on the entire history of stages and use a rule in the spirit of fictitious play. Since there are games where fictitious play converges but best-reply dynamics do not, this could extend our results to another interesting class of games, as long as the errors in each period do not accumulate over time. Another possibility is to update probabilistically or use a tolerance to determine whether to update (see, e.g., Foster & Young, 2006; Hart & Mas-Colell, 2001). This could allow convergence in games where best-reply dynamics oscillate or decrease the fraction of agents who make mistakes once the system reaches equilibrium.

6.3 Model Assumptions

Our model makes several unrealistic assumptions, most notably that there are countably many agents who all share the same utility function. Essentially the same results holds with a large, finite number of agents, adding a few more "error terms." In particular, since there is always a small probability that every agent makes a mistake at the same time, we can prove only that no more than a $1 - e$ fraction of the agents make errors in most rounds, and that agents spend most of their time playing equilibrium strategies.

We have also implicitly assumed that the set of agents is fixed. As Figure 9 shows, we could easily allow for churn. A natural strategy for newly arriving agents is to pick a random a_ϵ to use in the next stage. If all agents do this, it follows that convergence is unaffected: we can treat the new agents as part of the e fraction that made a mistake in the last stage. Furthermore, this tells us that newly arriving agents "catch up" very quickly. After a single stage, new agents are guaranteed to have learned a best reply with probability at least $1 - e$.

Finally, we have assumed that all agents have the same utility function. Our results can easily be extended to include a finite number of different types of agents, each with their own utility function, since the SLLN can be applied to each type of agent. This extension is discussed in Appendix A. We believe that our results hold even if the set of possible types is infinite. This can happen, for example, if an agent's utility depends on a valuation drawn from some interval. However, some care is needed to define best-reply sequences in this case.

6.4 State

One common feature of distributed systems not addressed in the theoretical portion of this work is state. As we saw with the scrip system in Section 4.4, an agent's current state is often an important factor in choosing an optimal action.

In principle, we could extend our framework to games with state: in each stage each agent chooses a policy to usually follow and explores other actions with probability ϵ. Each agent could then use some *off-policy algorithm* (one where the agent can learn without controlling the sequence of observations; see Kaelbling, Littman, & Moore, 1996 for examples) to learn an optimal policy to use in the next stage. One major problem with this approach is that standard algorithms learn too slowly for our purposes. For example, Q-learning (Watkins & Dayan, 1992) typically needs to observe each state-action pair hundreds of times in practice. The low exploration probability means that the expected $|S||A|/\epsilon$ rounds needed to explore each pair even once is large. Efficient learning requires more specialized algorithms that can make better use of the structure of a problem. However, the use of specialized algorithms makes providing a general guarantee of convergence more difficult. Another problem is that, even if an agent explores each action for each of his possible local states, the payoff he receives will depend on the states of the other agents, and thus the actions they chose. We need some property of the game to guarantee that this distribution of states is in some sense "well behaved." Adlakha and Johari's (2010) work on mean field equilibria gives one such condition. In this setting, the use of publicly available statistics might provide a solution to these problems.

6.5 Mixed Equilibria

Another restriction of our results is that our agents only learn pure strategies. One way to address this is to discretize the mixed strategy space (see, e.g., Foster & Young, 2006). If one of the resulting strategies is sufficiently close to an equilibrium strategy and best-reply dynamics converge with the discretized strategies, then we expect agents to converge to a near-equilibrium distribution of strategies. We have had empirical success using this approach to learn to play rock-paper-scissors.

7. Conclusion

Learning in distributed systems requires algorithms that are scalable to thousands of agents and can be implemented with minimal information about the actions of other agents. Most general-purpose multiagent learning algorithms fail one or both of these requirements. We have shown here that stage learning can be an efficient solution in large anonymous games where approximate best-reply dynamics lead to approximate pure strategy Nash equilibria. Many interesting classes of games have this property, and it is frequently found in designed games. In contrast to previous work, the time to convergence guaranteed by the theorem does not increase with the number of agents. If system designers can find an appropriate game satisfying these properties on which to base their systems, they can be confident that nodes can efficiently learn appropriate behavior.

Our results also highlight two factors that aid convergence. First, having more learners often improves performance. With more learners, the noise introduced into payoffs by exploration and mistakes becomes more consistent. Second, having more information typically improves performance. Publicly available statistics about the observed behavior of agents can allow an agent to learn effectively while making fewer local observations. Our simulations demonstrate the effects of these two factors, as well how our results generalize to situations with other learning algorithms, churn, asynchrony, and Byzantine behavior.

7.1 Acknowledgments

Most of the work was done while IK was at Cornell University. EF, IK, and JH are supported in part by NSF grant ITR-0325453. JH is also supported in part by NSF grant IIS-0812045 and by AFOSR grants FA9550-08-1-0438 and FA9550-05-1-0055. EF is also supported in part by NSF grant CDI-0835706.

Appendix A. Multiple Types

In this section, we extend our definition of a large anonymous game to settings where agents may have different utility functions. To do so, we introduce the notion of a type. Agents' utilities may depend on their type and the fraction of each type taking each action. As our results rely on the strong law of large numbers, we restrict the set of types to be finite. Formally, a *large anonymous game with types* is characterized by a tuple $\Gamma = (\mathbb{N}, T, \tau, A, P, \Pr)$. We define $\mathbb{N}$, A, G, and P as before. For the remaining terms:

- T is a finite set of agent *types*.
- $\tau : \mathbb{N} \to T$ is a function mapping each agent to his type.
- As before, $\Delta(A)$ is the set of probability distributions over A, and can be viewed as the set of mixed actions available to an agent. But now, to describe the fraction of agents of each type choosing each action, we must use element of $\Delta(A)^T$.
- $\Pr : A \times T \times \Delta(A)^T \to \Delta(P)$ determines the distribution over payoffs that results when an agent of type t performs action a and other agents follow action profile ρ. The expected utility of an agent of type t who performs mixed action s when other agents

follow action distribution ρ is $u(s,t,\rho) = \sum_{a \in A} \sum_{p \in P} ps(a) \Pr_{a,t,\rho}(p)$. As before, we further require that Pr (and thus u) be *Lipschitz continuous*.

The revised definitions of an η-best reply, an η-Nash equilibrium, an η-best-reply sequence, convergence of approximate best-reply dynamics, and (e, ϵ)-close follow naturally from the revised definitions of ρ and u. Lemma 3.1 now applies to each type of agent separately, and shows that all but a small fraction of each type will learn an approximate best reply in each stage. Lemma 3.3 and Lemma 3.4 hold given the revised definitions of ρ and u. Thus Theorem 3.5, which combines these, also still holds.

References

Adlakha, S., & Johari, R. (2010). Mean field equilibrium in dynamic games with complementarities. In *IEEE Conference on Decision and Control (CDC)*.

Adlakha, S., Johari, R., Weintraub, G. Y., & Goldsmith, A. (2010). Mean field analysis for large population stochastic games. In *IEEE Conference on Decision and Control (CDC)*.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, *32*(1), 48–77.

Blonski, M. (2001). Equilibrium characterization in large anonymous games. Tech. rep., U. Mannheim.

Blum, A., Even-Dar, E., & Ligett, K. (2006). Routing without regret: on convergence to Nash equilibria of regret-minimizing algorithms in routing games. In *25th ACM Symp. on Principles of Distributed Computing (PODC)*, pp. 45–52.

Blum, A., & Mansour, Y. (2007). Learning, regret minimization, and equilibria. In Nisan, N., Roughgarden, T., Tardos, É., & Vazirani, V. (Eds.), *Algorithmic Game Theory*, pp. 79–102. Cambridge University Press.

Bowling, M. H. (2000). Convergence problems of general-sum multiagent reinforcement learning. In *17th Int. Conf. on Machine Learning (ICML 2000)*, pp. 89–94.

Bowling, M. H., & Veloso, M. M. (2001). Rational and convergent learning in stochastic games. In *17th Int. Joint Conference on Artificial Intelligence (IJCAI 2001)*, pp. 1021–1026.

Boylan, R. T. (1992). Laws of large numbers for dynamical systems with randomly matched indviduals. *Journal of Economic Theory*, *57*, 473–504.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, Learning and Games*. Cambridge University Press.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI-97 Workshop on Multiagent Learning*, pp. 746–752.

Daskalakis, C., & Papadimitriou, C. H. (2007). Computing equilibria in anonymous games. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pp. 83–93.

Foster, D. P., & Young, P. (2006). Regret testing: Learning to play Nash equilibrium without knowing you have an opponent. *Theoretical Economics*, *1*, 341–367.

Friedman, E. J., & Shenker, S. (1998). Learning and implementation on the internet. Tech. rep., Cornell University.

Fudenberg, D., & Levine, D. (1998). *Theory of Learning in Games*. MIT Press.

Germano, F., & Lugosi, G. (2007). Global Nash convergence of Foster and Young's regret testing. *Games and Economic Behavior*, *60*(1), 135–154.

Gradwohl, R., & Reingold, O. (2008). Fault tolerance in large games. In *Proc. 9th ACM Conference on Electronic Commerce (EC 2008)*, pp. 274–283.

Greenwald, A., Friedman, E. J., & Shenker, S. (2001). Learning in networks contexts: Experimental results from simulations. *Games and Economic Behavior*, *35*(1-2), 80–123.

Hart, S., & Mas-Colell, A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, *68*(5), 1127–1150.

Hart, S., & Mas-Colell, A. (2001). A reinforecement learning procedure leading to correlated equilibrium. In Debreu, G., Neuefeind, W., & Trockel, W. (Eds.), *Economic Essays*, pp. 181–200. Springer.

Hopkins, E. (1999). Learning, matching, and aggregation. *Games and Economic Behavior*, *26*, 79–110.

Hu, J., & Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, *4*, 1039–1069.

Jafari, A., Greenwald, A. R., Gondek, D., & Ercal, G. (2001). On no-regret learning, fictitious play, and nash equilibrium. In *Proc. Eighteenth International Conference on Machine Learning (ICML)*, pp. 226–233.

Kaelbling, L. P., Littman, M. L., & Moore, A. P. (1996). Reinforcement learning: A survey. *J. Artif. Intell. Res. (JAIR)*, *4*, 237–285.

Kash, I. A., Friedman, E. J., & Halpern, J. Y. (2007). Optimizing scrip systems: Efficiency, crashes, hoarders and altruists. In *Eighth ACM Conference on Electronic Commerce (EC 2007)*, pp. 305–315.

Kephart, J. O., Hanson, J. E., & Greenwald, A. R. (2000). Dynamic pricing by software agents. *Computer Networks*, *32*(6), 731–752.

Marden, J. R., Arslan, G., & Shamma, J. S. (2007a). Connections between cooperative control and potential games. In *Proc. 2007 European Control Conference (ECC)*.

Marden, J. R., Arslan, G., & Shamma, J. S. (2007b). Regret based dynamics: convergence in weakly acyclic games. In *6th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 42–49.

Marden, J. R., Young, H. P., Arslan, G., & Shamma, J. S. (2009). Payoff-based dynamics for multi-player weakly acyclic games. *SIAM Journal on Control and Optimization*, *48*(1), 373–396.

Milgrom, P., & Roberts, J. (1990). Rationalizability, learning, and equilibrium in games with strategic complement- arities. *Econometrica*, *58*(6), 1255–1277.

Nisan, N., Schapira, M., Valiant, G., & Zohar, A. (2011). Best-response mechanisms. In *Proc. Second Syposium on Innovations in Computer Science (ICS)*. To Appear.

Nisan, N., Schapira, M., & Zohar, A. (2008). Asynchronous best-reply dynamics. In *Proc. 4th International Workshop on Internet and Netwrok Economics (WINE)*, pp. 531–538.

Osborne, M., & Rubenstein, A. (1994). *A Course in Game Theory*. MIT Press.

Shoham, Y., Powers, R., & Grenager, T. (2003). Multi-agent reinforcement learning: a critical survey. Tech. rep., Stanford.

Tesauro, G., & Kephart, J. O. (2002). Pricing in agent economies using multi-agent Q-learning. *Autonomous Agents and Multi-Agent Systems*, *5*(3), 289–304.

Verbeeck, K., Nowé, A., Parent, J., & Tuyls, K. (2007). Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Journal of Autonomous Agents and Multi-agent Systems*, *14*, 239–269.

Watkins, C. J., & Dayan, P. (1992). Technical note Q-learning. *Machine Learning*, *8*, 279–292.

Young, H. P. (2009). Learning by trial and error. *Games and Economic Behavior*, *65*(2), 626–643.