# Correlated $Q$-Learning

**Amy Greenwald** AMY@BROWN.EDU
*Department of Computer Science*
*Brown University*
*Providence, RI 02912*

**Keith Hall** KEITH_HALL@JHU.EDU
*Department of Computer Science*
*Johns Hopkins University*
*Baltimore, MD 21201*

**Martin Zinkevich** MAZ@CS.BROWN.EDU
*Department of Computer Science*
*Brown University*
*Providence, RI 02912*

**Editor:** Leslie Pack Kaelbling

## Abstract

Recently, there have been several attempts to design multiagent $Q$-learning algorithms that learn equilibrium policies in general-sum Markov games, just as $Q$-learning learns optimal policies in Markov decision processes. We introduce *correlated* $Q$-learning, one such algorithm based on the correlated equilibrium solution concept. Motivated by a fixed point proof of the existence of stationary correlated equilibrium policies in Markov games, we present a generic multiagent $Q$-learning algorithm of which many popular algorithms are immediate special cases. We also prove that certain variants of correlated (and Nash) $Q$-learning are guaranteed to converge to stationary correlated (and Nash) equilibrium policies in two special classes of Markov games, namely zero-sum and common-interest. Finally, we show empirically that correlated $Q$-learning outperforms Nash $Q$-learning, further justifying the former beyond noting that it is less computationally expensive than the latter.

**Keywords:** Multiagent Learning, Reinforcement Learning, Markov Games

## 1. Introduction

Recently, there have been several attempts to design multiagent $Q$-learning algorithms that learn equilibrium policies in general-sum Markov games, just as $Q$-learning learns optimal policies in Markov decision processes. Hu and Wellman (2003) propose an algorithm called Nash-$Q$ that converges to Nash equilibrium policies in general-sum games under restrictive conditions. Littman's (2001) friend-or-foe-$Q$ (FF-$Q$) algorithm always converges, but only learns equilibrium policies in restricted classes of games. For example, Littman's (1994) minimax-$Q$ algorithm (equivalently, foe-$Q$) converges to minimax equilibrium policies in two-player, zero-sum games. This paper introduces correlated $Q$-learning (CE-$Q$), a multiagent $Q$-learning algorithm based on the correlated equilibrium solution concept (Aumann, 1974). Correlated-$Q$ generalizes Nash-$Q$ in general-sum games, in that the set of correlated equilibria contains the set of Nash equilibria. Correlated-$Q$ also generalizes minimax-$Q$ in zero-sum games, where the set of Nash and minimax equilibria coincide.

A Nash equilibrium is a vector of independent strategies, each of which is a probability distribution over actions, in which each agent's strategy is optimal given the strategies of the other agents. A correlated equilibrium is more general than a Nash equilibrium in that it allows for dependencies among agents' strategies: a correlated equilibrium is a joint distribution over actions from which no agent is motivated to deviate unilaterally.

An everyday example of a correlated equilibrium is a traffic signal. For two agents that meet at an intersection, the traffic signal translates into the joint probability distribution (STOP,GO) with probability $p$ and (GO,STOP) with probability $1 - p$. No probability mass is assigned to (GO,GO) or (STOP,STOP). An agent's optimal action given a red signal is to stop, while an agent's optimal action given a green signal is to go.

The set of correlated equilibria (CE) is a convex polytope; thus, unlike Nash equilibria (NE), the computation of which was recently shown to be PPAD-complete (Chen and Deng, 2005), CE can be computed efficiently via linear programming. In addition, CE that are not NE can achieve higher rewards than NE, by avoiding positive probability mass on less desirable outcomes (e.g., avoiding collisions at a traffic signal). Finally, CE is consistent with the usual model of independent agent behavior in artificial intelligence: after a private signal is observed, each agent chooses its action independently.

One of the difficulties in learning (Nash or) correlated equilibrium policies in general-sum Markov games stems from the fact that in general-sum one-shot games, there exist multiple equilibria with multiple values. Indeed, in any implementation of multiagent $Q$-learning, an equilibrium selection problem arises. We attempt to resolve this problem by introducing four variants of correlated-$Q$, based on four equilibrium selection mechanisms—utilitarian, egalitarian, plutocratic, and dictatorial. We analyze the convergence properties of these algorithms in two special classes of Markov games, zero-sum and common-interest.

In addition to our theoretical analyses, we demonstrate the following empirically:

1. In a stylized version of grid soccer, a zero-sum Markov game (a simpler setting than general-sum), $Q$-learning already fails to converge. This observation is one of the key motivations for the study of multiagent $Q$-learning algorithms.

2. In three stylized general-sum grid games, the variants of correlated-$Q$ studied perform on par with related variants of Nash-$Q$. The former outperforms the latter only in the game where there exists a CE that achieves higher rewards than any NE.

3. On a random test bed of general-sum Markov games, the variants of both correlated and Nash $Q$-learning studied often fail to converge. Still, the former is more successful at finding stationary equilibrium policies than the latter.

**Overview** This paper is organized as follows. We start by reviewing the definitions of correlated equilibrium in one-shot games and correlated equilibrium *policies* in Markov games. In Section 3, we define two versions of multiagent $Q$-learning, one centralized and one decentralized, and we show how correlated-$Q$, Nash-$Q$, and FF-$Q$ all arise as special cases of these generic algorithms. In Section 4, we include a theoretical discussion of zero-sum and common-interest Markov games, in which we prove that certain variants of correlated $Q$-learning are guaranteed to converge to stationary equilibrium policies. In Section 6, we describe simulation experiments that compare utilitarian, egalitarian, plutocratic, and dictatorial correlated $Q$-learning with $Q$-learning, FF-$Q$, and two variants of Nash-$Q$.

## 2. Correlated Equilibrium Policies in Markov Games

In this section, we review the definitions of correlated equilibrium in one-shot games and correlated equilibrium *policies* in Markov games. Also, by way of motivating the iterative algorithms developed in the next section, we sketch the relevant bits of a fixed point proof of the existence of stationary correlated equilibrium policies in Markov games.

We begin with some notation and terminology that we rely on to define Markov games. We adopt the following standard game-theoretic terminology: the term action (strategy, or policy) *profile* is used to mean a vector of actions (strategies, or policies), one per player. In addition, $\Delta(X)$ denotes the set of all probability distributions over finite set $X$.

**Definition 1** *A (finite, discounted)* ***Markov game*** *is a tuple* $\Gamma_\gamma = \langle N, S, A, P, R \rangle$ *in which*

- $N$ *is a finite set of* $n$ *players*
- $S$ *is a finite set of* $m$ *states*
- $A = \prod_{i \in N, s \in S} A_i(s)$*, where* $A_i(s)$ *is player* $i$*'s finite set of pure actions at state* $s$*; we define* $A(s) \equiv \prod_{i \in N} A_i(s)$ *and* $A_{-i}(s) = \prod_{j \neq i} A_j(s)$*, so that* $A(s) = A_{-i}(s) \times A_i(s)$*; we write* $a = (a_{-i}, a_i) \in A(s)$ *to distinguish player* $i$*, with* $a_i \in A_i(s)$ *and* $a_{-i} \in A_{-i}(s)$*; we also define* $\mathcal{A} = \bigcup_{s \in S} \bigcup_{a \in A(s)} \{(s, a)\}$*, the set of state-action pairs.*
- $P$ *is a system of transition probabilities: i.e., for all* $s \in S$*,* $a \in A(s)$*,* $P[s' \mid s, a] \geq 0$ *and* $\sum_{s' \in S} P[s' \mid s, a] = 1$*; we interpret* $P[s' \mid s, a]$ *as the probability that the next state is* $s'$ *given that the current state is* $s$ *and the current action profile is* $a$
- $R : \mathcal{A} \to [\alpha, \beta]^n$*, where* $R_i(s, a) \in [\alpha, \beta]$ *is player* $i$*'s reward at state* $s$ *and at action profile* $a \in A(s)$
- $\gamma \in [0, 1)$ *is a discount factor*

Let us imagine that in addition to the players, there is also a *referee*,[1] who can be considered to be a physical machine (i.e., the referee itself has no beliefs, desires, or intentions). At each time step, the referee sends to each player a private signal consisting of a recommended action for that player. We often assume the referee selects these actions according to a *stationary* policy $\pi \in \prod_{s \in S} \Delta(A(s))$ that depends on state, but not on time.

The dynamics of a discrete-time Markov game *with a referee* unfold as follows: at time $t = 1, 2, \ldots$, the players and the referee observe the current game state $s^t \in S$; following its policy $\pi$, the referee selects the distribution $\pi_{s^t}$, based on which it recommends an action, say $\alpha_i^t$, to each player $i$; given its recommendation, each player selects an action $a_i^t$, and the pure action profile $a^t = (a_1^t, \ldots, a_n^t)$ is played; based on the current state and action profile, each player $i$ now earns reward $R_i(s^t, a^t)$; finally, nature selects a successor state $s^{t+1}$ with transition probability $P[s^{t+1} \mid s^t, a^t]$; the process repeats at time $t + 1$.

[1] Note that the referee is not part of the definition of a Markov game. While a referee can be of assistance in the implementation of a correlated equilibrium, the concept can be defined without reference to this third party. In this section, we introduce the referee as a pedagogical device. In our experimental work, we sometimes rely on the referee to facilitate the implementation of correlated equilibria.

### 2.1 Correlated Equilibrium in One-Shot Games: A Review

A (finite) *one-shot game* is a tuple $\Gamma = \langle N, A, R \rangle$ in which $N$ is a finite set of $n$ players; $A = \prod_{i \in N} A_i$, where $A_i$ is player $i$'s finite set of pure actions; and $R : A \to \mathbb{R}^n$, where $R_i(a)$ is player $i$'s reward at action profile $a \in A$.

Once again, imagine a referee who selects an action profile $a$ according to some policy $\pi \in \Delta(A)$. The referee advises player $i$ to follow action $a_i$. Define $A_{-i} = \prod_{j \neq i} A_j$. Define $\pi(a_i) = \sum_{a_{-i} \in A_{-i}} \pi(a_{-i}, a_i)$ and $\pi(a_{-i} \mid a_i) = \frac{\pi(a_{-i}, a_i)}{\pi(a_i)}$ whenever $\pi(a_i) > 0$.

**Definition 2** *Given a one-shot game $\Gamma$, the referee's policy $\pi \in \Delta(A)$ is a **correlated equilibrium** if, for all $i \in N$, for all $a_i \in A_i$ with $\pi(a_i) > 0$, and for all $a_i' \in A_i$,*

$$\sum_{a_{-i} \in A_{-i}} \pi(a_{-i} \mid a_i) R_i(a_{-i}, a_i) \geq \sum_{a_{-i} \in A_{-i}} \pi(a_{-i} \mid a_i) R_i(a_{-i}, a_i') \tag{1}$$

If the referee chooses $a$ according to correlated equilibrium policy $\pi$, then the players are motivated to follow his advice, because the expression $\sum_{a_{-i} \in A_{-i}} \pi(a_{-i} \mid a_i) R(a_{-i}, a_i')$ computes player $i$'s expected reward for playing $a_i'$ when the referee advises him to play $a_i$.

Equivalently, for all $i \in N$ and for all $a_i, a_i' \in A_i$,

$$\sum_{a_{-i} \in A_{-i}} \pi(a_{-i}, a_i) R_i(a_{-i}, a_i) \geq \sum_{a_{-i} \in A_{-i}} \pi(a_{-i}, a_i) R_i(a_{-i}, a_i') \tag{2}$$

Equation 2 is Equation 1 multiplied by $\pi(a_i)$. Equation 2 holds trivially whenever $\pi(a_i) = 0$, because in such cases both sides equal zero. Given a one-shot game $\Gamma$, $R(a_{-i}, a_i)$ is known, which implies that Equation 2 is a system of linear inequalities, with $\pi(a_{-i}, a_i)$ unknown.

The set of all solutions to a system of linear inequalities is convex. Since these inequalities are not strict, this set is also closed. This set is bounded as well, because the set of all policies is bounded. Therefore, the set of correlated equilibria is compact and convex.

If the recommendations of the referee in a correlated equilibrium are independent (i.e., for all $i \in N$, for all $a_i, a_i' \in A_i$, for all $a_{-i} \in A_{-i}$, $\pi(a_{-i} \mid a_i) = \pi(a_{-i} \mid a_i')$, whenever $\pi(a_i), \pi(a_i') > 0$), then a correlated equilibrium is also a Nash equilibrium. In fact, any Nash equilibrium can be represented as a correlated equilibrium: the players can simply generate their own advice (independently). Existence of both types of equilibria is ensured by Nash's theorem (Nash, 1951). Therefore, the set of correlated equilibria is nonempty, as well. We have established the following (well-known) result.

**Theorem 3** *The set of correlated equilibria in a one-shot game is nonempty, compact, and convex.*

Finally, we note two important features of correlated equilibria. First, a correlated equilibrium in a one-shot game can be computed in polynomial time via linear programming. Equation 2 consists of $\sum_{i \in N} |A_i| (|A_i| - 1)$ linear inequalities, which is polynomial in the number of players, and $\prod_{i \in N} |A_i| - 1$ variables, which is exponential in the number of players, but polynomial in the size of the game. Second, correlated equilibrium rewards in a one-shot game can fall outside the convex hull of all Nash equilibrium rewards, and hence the former can make all players better off than the latter (Aumann, 1974).

### 2.2 Correlated Equilibrium Policies in Markov Games: Definition

Intuitively, it is straightforward to generalize the definition of correlated equilibrium in one-shot games to correlated equilibrium *policies* in Markov games:

**Definition 4** *Given a Markov game $\Gamma_\gamma$, a referee's policy $\pi$ is a **correlated equilibrium** if for any agent $i$, if all the other agents follow the advice of the referee, agent $i$ maximizes its expected utility by also following the advice of the referee.*

To operationalize this definition, we compute the expected utility of an agent when it follows the advice of the referee as well as the expected utility of an agent when it deviates, in both cases assuming all other agents follow the advice of the referee.

Given a Markov game $\Gamma_\gamma$, and a referee's policy $\pi$, consider the transition matrix $T^\pi$ such that $T^\pi_{ss'}$ is the probability of transitioning to state $s'$ from state $s$, given that the referee selects an action profile according to the distribution $\pi_s$ that the agents indeed follow:

$$T^\pi_{ss'} = \sum_{a \in A(s)} \pi_s(a) P[s' \mid s, a] \tag{3}$$

Exponentiating this matrix, the probability of transitioning to state $s'$ from state $s$ after $t$ time steps is given by $(T^\pi_{ss'})^t$. Now the value function $V_i^\pi(s)$ represents agent $i$'s expected reward, originating at state $s$, assuming all agents follow the referee's policy $\pi$:

$$V_i^\pi(s) = (1-\gamma) \sum_{t=0}^{\infty} \sum_{s' \in S} \gamma^t (T^\pi_{ss'})^t \sum_{a \in A(s')} \pi_{s'}(a) R_i(s', a) \tag{4}$$

The $Q$-value function $Q_i^\pi(s,a)$ represents agent $i$'s expected rewards if action profile $a$ is played in state $s$ and the referee's policy $\pi$ is followed thereafter:

$$Q_i^\pi(s,a) = (1-\gamma) \left( R_i(s,a) + \gamma \sum_{s' \in S} P[s'|s,a] \left( \sum_{t=0}^{\infty} \sum_{s'' \in S} \gamma^t (T^\pi_{s's''})^t \sum_{a \in A(s'')} \pi_{s''}(a) R_i(s'', a) \right) \right) \tag{5}$$

The normalization constant $1-\gamma$ ensures that the ranges of $V_i^\pi$ and $Q_i^\pi$ each fall in $[\alpha, \beta]$.

The following theorem, which we state without proof, follows from Equations 4 and 5 via the Markov property. (Note also that the referee's policy $\pi$ is stationary by assumption.)

**Theorem 5** *Given a Markov game $\Gamma_\gamma$, for any $V : S \to [\alpha,\beta]^n$, for any $Q : \mathcal{A} \to [\alpha,\beta]^n$, and for any stationary policy $\pi$, $V = V^\pi$ and $Q = Q^\pi$ if and only if for all $i \in N$,*

$$\begin{aligned} V_i(s) &= \sum_{a \in A(s)} \pi_s(a) Q_i(s,a) & (6) \\ Q_i(s,a) &= (1-\gamma) R_i(s,a) + \gamma \sum_{s' \in S} P[s'|s,a] V_i(s') & (7) \end{aligned}$$

Hereafter, in place of Equations 4 and 5, we define $V_i^\pi$ and $Q_i^\pi$ recursively as the unique pair of functions satisfying Equations 6 and 7.

Define $\pi_s(a_i) = \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i)$ and $\pi_s(a_{-i} \mid a_i) = \frac{\pi_s(a_{-i}, a_i)}{\pi_s(a_i)}$ whenever $\pi_s(a_i) > 0$.

**Remark 6** *Given a Markov game $\Gamma_\gamma$, a stationary policy $\pi$ is* **not** *a correlated equilibrium if there exists an $i \in N$, an $s \in S$, an $a_i \in A_i(s)$ with $\pi(a_i) > 0$, and an $a_i' \in A_i(s)$,* s.t.*:*

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} \mid a_i) Q_i^\pi(s,(a_{-i},a_i)) < \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} \mid a_i) Q_i^\pi(s,(a_{-i},a_i')) \tag{8}$$

Here, in state $s$, when it is recommended that agent $i$ play $a_i$, it would rather play $a_i'$, since the expected utility of $a_i'$ is greater than the expected utility of $a_i$. This is an example of a *one-shot deviation* (see, for example, Osborne and Rubinstein (1994)). The definition of correlated equilibrium in Markov games, however, permits arbitrarily complex deviations on the part of an agent: e.g., deviations could be nonstationary. The next theorem states that the converse of Remark 6 is also true, implying that it suffices to consider one-shot deviations. Together Remark 6 and Theorem 7 provide the necessary and sufficient conditions for $\pi$ to be a stationary correlated equilibrium policy in a Markov game.

**Theorem 7** *Given a Markov game $\Gamma_\gamma$, a stationary policy $\pi$ is a correlated equilibrium if for all $i \in N$, for all $s \in S$, for all $a_i \in A_i(s)$ with $\pi(a_i) > 0$, for all $a_i' \in A_i(s)$,*

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} \mid a_i) Q_i^\pi(s,(a_{-i},a_i)) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} \mid a_i) Q_i^\pi(s,(a_{-i},a_i')) \tag{9}$$

Here, in state $s$, when it is recommended that agent $i$ play $a_i$, it prefers to play $a_i$, because the expected utility of $a_i$ is greater than or equal to the expected utility of $a_i'$, for all $a_i'$.

Observe the following: if all of the other agents but agent $i$ play according to the referee's (stationary) policy $\pi$, then from the point of view of agent $i$, its environment is an MDP. Hence, the one-shot deviation principle for MDPs (see, for example, Greenwald and Zinkevich (2005)) establishes Theorem 7.

**Corollary 8** *Given a Markov game $\Gamma_\gamma$, a stationary policy $\pi$ is a correlated equilibrium if for all $i \in N$, for all $s \in S$, and for all $a_i, a_i' \in A_i(s)$,*

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i) Q_i^\pi(s,(a_{-i},a_i)) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i) Q_i^\pi(s,(a_{-i},a_i')) \tag{10}$$

Equation 10 is Equation 9 multiplied by $\pi_s(a_i)$.

Unlike in one-shot games where only the $\pi(a_{-i}, a_i)$'s are unknown (see Equation 2), here the $\pi_s(a_{-i}, a_i)$'s, and hence the $Q_i^\pi(s,(a_{-i},a_i))$'s, are unknown. In particular, Equation 10 is not a system of linear inequalities, but rather a system of nonlinear inequalities. Next, we discuss the existence of a solution to this nonlinear system.

### 2.3 Correlated Equilibrium Policies in Markov Games: Existence

In Theorem 5, Equation 6 is dependent upon the referee's policy $\pi$, whereas Equation 7 depends on the structure of the game (rewards and transition probabilities). Like value iteration and $Q$-learning, which are used to compute optimal policies in MDPs, we can try to leverage this separation to search for correlated equilibrium policies in Markov games.

Given a Markov game $\Gamma_\gamma$, with a particular $S$ and $\mathcal{A}$, define the following three spaces:

1. $\mathcal{V} = [\alpha, \beta]^{n \times S}$, the space of all functions of the form $V : S \rightarrow [\alpha, \beta]^n$
2. $\mathcal{Q} = [\alpha, \beta]^{n \times \mathcal{A}}$, the space of all functions of the form $Q : \mathcal{A} \rightarrow [\alpha, \beta]^n$
3. $\Pi = \prod_{s \in S} \Delta(A(s))$, the space of all policies

When viewed as subsets of Euclidean space, these are nonempty, compact, convex sets.

Let us express the value function defined in Equation 6 more precisely by making explicit its dependence on the $Q$-values as well as the policy $\pi$. Define $V_{\mathcal{Q} \times \Pi} : \mathcal{Q} \times \Pi \rightarrow \mathcal{V}$ such that for all $Q \in \mathcal{Q}$, for all $\pi \in \Pi$, for all $i \in N$, and for all $s \in S$,

$$(V_{\mathcal{Q} \times \Pi}(Q, \pi))_i(s) = \sum_{a \in A(s)} \pi_s(a) Q_i(s, a) \tag{11}$$

Similarly, let us also express the $Q$-value function defined in Equation 7 more precisely so that its dependence on the value function is made explicit. Define $Q_{\mathcal{V}} : \mathcal{V} \rightarrow \mathcal{Q}$ such that for all $V \in \mathcal{V}$, for all $i \in N$, for all $s \in S$, and for all $a \in A(s)$,

$$(Q_{\mathcal{V}}(V))_i(s, a) = (1 - \gamma) R_i(s, a) + \gamma \sum_{s' \in S} P[s'|s, a] V_i(s') \tag{12}$$

Here, we have highlighted the fact that the dependence of $Q$ on $\pi$ arises through $V$. Finally, we define the correspondence $\pi^*_{\mathcal{Q}} : \mathcal{Q} \rightarrow\rightarrow \Pi$ such that for all $Q \in \mathcal{Q}$, $\pi \in \Pi$ is in $\pi^*_{\mathcal{Q}}(Q)$ if and only if for all $s \in S$, for all $a_i, a_i' \in A_i(s)$:

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i) Q(s, (a_{-i}, a_i)) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i) Q(s, (a_{-i}, a_i')) \tag{13}$$

**Theorem 9** *Given a Markov game $\Gamma_\gamma$, for $V \in \mathcal{V}$, $Q \in \mathcal{Q}$, and $\pi \in \Pi$, if $V = V_{\mathcal{Q} \times \Pi}(Q, \pi)$, $Q = Q_{\mathcal{V}}(V)$, and $\pi \in \pi^*_{\mathcal{Q}}(Q)$, then $V = V^\pi$, $Q = Q^\pi$, and $\pi$ is a stationary correlated equilibrium policy.*

**Proof** By Equations 11 and 12, for all $i \in N$, for all $s \in S$, for all $a \in A(s)$:

$$\begin{aligned} V_i(s) &= \sum_{a \in A(s)} \pi_s(a) Q_i(s, a) && (14) \\ Q_i(s, a) &= (1 - \gamma) R_i(s, a) + \gamma \sum_{s' \in S} P[s'|s, a] V_i(s') && (15) \end{aligned}$$

Therefore, by Theorem 5, $V = V^\pi$ and $Q = Q^\pi$. Finally, since $\pi \in \pi^*_{\mathcal{Q}}(Q^\pi)$, it follows from Corollary 8 that $\pi$ is a stationary correlated equilibrium policy. ■

Define $\rho^* \equiv (I \times \pi^*_{\mathcal{Q}}) \circ Q_{\mathcal{V}} \circ V_{\mathcal{Q} \times \Pi}$, where $I$ is the identity function.

**Corollary 10** *Given a Markov game $\Gamma_\gamma$, a fixed point (if it exists) of the correspondence $\rho^* : \mathcal{Q} \times \Pi \rightarrow\rightarrow \mathcal{Q} \times \Pi$ is a pair consisting of a stationary correlated equilibrium policy and its associated $Q$-values.*

**Proof** Choose $(Q, \pi)$ to be a fixed point of $\rho^*$. Define $V = V_{\mathcal{Q}\times\Pi}(Q, \pi)$.

$$\begin{aligned} (Q,\pi) &\in \rho^*(Q,\pi) && (16)\\ &= \{(Q_{\mathcal{V}}(V_{\mathcal{Q}\times\Pi}(Q,\pi)),\pi') \mid \pi' \in \pi^*_{\mathcal{Q}}(Q_{\mathcal{V}}(V_{\mathcal{Q}\times\Pi}(Q,\pi)))\} && (17)\\ &= \{(Q_{\mathcal{V}}(V),\pi') \mid \pi' \in \pi^*_{\mathcal{Q}}(Q_{\mathcal{V}}(V))\} && (18) \end{aligned}$$

Therefore, $Q = Q_{\mathcal{V}}(V)$ so that $\pi \in \pi^*_{\mathcal{Q}}(Q)$. By Theorem 9, $V = V^\pi$, $Q = Q^\pi$, and $\pi$ is a stationary correlated equilibrium policy. ■

**Theorem 11** *The correspondence $\rho^* = (I \times \pi^*_{\mathcal{Q}}) \circ Q_{\mathcal{V}} \circ V_{\mathcal{Q}\times\Pi}$ has a fixed point.*

**Proof** See, for example, Greenwald and Zinkevich (2005). ■

**Theorem 12** *Every Markov game has a stationary correlated equilibrium policy.*

**Proof** By Theorem 11, the correspondence $\rho^* = (I \times \pi^*_{\mathcal{Q}}) \circ Q_{\mathcal{V}} \circ V_{\mathcal{Q}\times\Pi}$ has a fixed point. By Corollary 10, this fixed point is comprised of a stationary correlated equilibrium policy. ■

We have thus established the existence of a solution—a fixed point—to the nonlinear system of inequalities that characterize stationary correlated equilibrium policies in Markov games (Equation 10). Based on our derivation (specifically, Equations 11, 12, and 13), we propose a class of iterative algorithms intended to find such a fixed point in the next section. Then, in later sections, we investigate the question of whether or not any of the algorithms in our class succeed: i.e., converge to correlated equilibrium policies in Markov games. We obtain positive theoretical results in two special cases, and positive experimental results on certain games, but negative (experimental) results in general.

## 3. Multiagent $Q$-Learning

The derivation in the previous section suggests an iterative algorithm for computing *global* equilibrium policies based on *local* updates: given initial $Q$-values and an initial policy, update the values, $Q$-values, and policy at each state, and repeat.

In MDPs, the special case of Markov games with only a single agent, the corresponding local update procedure, known as value iteration, is well understood: Given $Q$-values at time $t$ for all $s \in S$ and for all $a \in A(s)$, namely $Q^t(s, a)$, at time $t + 1$,

$$\begin{aligned} V^{t+1}(s) &:= \max_{a\in A(s)} Q^t(s,a) && (19)\\ Q^{t+1}(s,a) &:= (1-\gamma)R(s,a) + \gamma \sum_{s'\in S} P[s'|s,a]V^{t+1}(s') && (20) \end{aligned}$$

This procedure converges to a unique fixed point $V^*$, a unique fixed point $Q^*$, and a globally optimal policy $\pi^*$, which is not necessarily unique (e.g., see Puterman (1994)).

More generally, in Markov games, given $Q$-values at time $t$ for all $i \in N$, for all $s \in S$, and for all $a \in A(s)$, namely $Q_i^t(s,a)$; given a policy $\pi^t$; and given a **selection mechanism** $f$, that is, a mapping from one-shot games into (sets of) joint distributions; at time $t+1$,

$$V_i^{t+1}(s) \quad := \quad \sum_{a \in A(s)} \pi_s^t(a) Q_i^t(s,a) \tag{21}$$

$$Q_i^{t+1}(s,a) \quad := \quad (1-\gamma) R_i(s,a) + \gamma \sum_{s' \in S} P[s'|s,a] V_i^{t+1}(s') \tag{22}$$

$$\pi_s^{t+1} \quad \in \quad f(Q^{t+1}(s)) \tag{23}$$

We now proceed to investigate the question of whether or not this procedure converges to equilibrium policies in Markov games, for various choices of the selection mechanism $f$.

Following the literature on this subject (e.g., Littman (1994, 2001); Hu and Wellman (2003)), we focus our study on "correlated $Q$-learning," in which state values and $Q$-values at state-action profile pairs are updated asynchronously (see Tables 1 and 2), rather than "correlated value iteration," in which these values are updated synchronously. The latter approach would be more faithful to Equations 21, 22, and 23.

### 3.1 Pseudocode: Centralized and Decentralized

The generalization of $Q$-learning from MDPs to Markov games is intuitively straightforward. However, one important application-specific issue arises: can we assume the existence of a trusted third party who can act as a referee, or central coordinator? Or need we decentralize the implementation of multiagent $Q$-learning? We present two generic formulations of multiagent $Q$-learning: one centralized (Table 1), and one decentralized (Table 2).

Given a Markov game, the multiagent $Q$-learning process is initialized at some state with some action profile, after which the game is played as follows: First (step 1), the current action profile is simulated in the current state. Second (step 2), the rewards at that state-action profile pair are observed, as is the next state. Third (step 3), a policy at that next state is selected. That policy is used to update each agent's value at the next state (step 4(a)), which in turn is used to update each agent's $Q$-value at the current state-action pair (step 4(b)). A new action profile is then selected, either on- or off-policy (step 5). An on-policy action profile is one that is sampled from the current policy (perhaps with some random exploration); an off-policy action profile (e.g., totally random exploration) need not be consistent with the current policy. This process repeats as necessary (e.g. until convergence), with the learning rate $\alpha$ decaying according to some schedule (step 7).

As already mentioned, a template for *centralized* multiagent $Q$-learning, appears in Table 1. Notably, in step 3, the central coordinator, who has knowledge of all agents' $Q$-values, selects a joint distribution on which the updates in step 4 rely.

Rather than rely on a central coordinator, Hu and Wellman (2003) assume that each agent can observe all the other agents' actions and rewards. With this assumption, one can *decentralize* the implementation of multiagent $Q$-learning. In step 3 of Table 2, each agent selects its own joint distribution on which to base its updates in step 4. Doing so requires knowledge of all agents' $Q$-values. By observing the actions and rewards of the other agents in step 2, sufficient information is available to perform this updating exactly as the central coordinator does in the centralized version of multiagent $Q$-learning.

```
CENTRALIZEDQ($\Gamma, f, g, \alpha$)
 Inputs      Markov game $\Gamma$, selection mechanism $f$, decay schedule $g$, learning rate $\alpha$
 Output      values $V$, $Q$-values $Q$, joint policy $\pi^*$
 Initialize  $Q$-values $Q$, state $s$, action profile $a$

REPEAT
    1.  simulate action profile $a$ in state $s$
    2.  observe rewards $R(s, a)$ and next state $s'$
    3.  select $\pi^*_{s'} \in f(Q(s'))$
    4.  for all agents $j$
        (a)  $V_j(s') = \sum_{a \in A_{s'}} \pi^*_{s'}(a) Q_j(s', a)$
        (b)  $Q_j(s, a) = (1 - \alpha) Q_j(s, a) + \alpha[(1 - \gamma) R_j(s, a) + \gamma V_j(s')]$
    5.  choose action profile $a'$ (on- or off-policy)
    6.  update $s = s'$, $a = a'$
    7.  decay $\alpha$ via $g$
FOREVER
```

Table 1: Multiagent $Q$-Learning: Centralized.

```
DECENTRALIZEDQ($\Gamma, f, g, \alpha, i$)
 Inputs      game $\Gamma$, selection mechanism $f$, decay schedule $g$, learning rate $\alpha$, agent $i$
 Output      values $V$, $Q$-values $Q$, joint policy $\pi^{i*}$
 Initialize  $Q$-values $Q$, state $s$, action profile $a$

REPEAT
    1.  simulate action $a_i$ in state $s$
    2.  observe action profile $a_{-i}$, rewards $R(s, a)$, and next state $s'$
    3.  select $\pi^{i*}_{s'} \in f(Q(s'))$
    4.  for all agents $j$
        (a)  $V_j(s') = \sum_{a \in A_{s'}} \pi^{i*}_{s'}(a) Q_j(s', a)$
        (b)  $Q_j(s, a) = (1 - \alpha) Q_j(s, a) + \alpha[(1 - \gamma) R_j(s, a) + \gamma V_j(s')]$
    5.  choose action $a'_i$ (on- or off-policy)
    6.  update $s = s'$, $a = a'_i$
    7.  decay $\alpha$ via $g$
FOREVER
```

Table 2: Multiagent $Q$-Learning: Decentralized.

In the remainder of this section, we show how these generic multiagent $Q$-learning algorithms can be instantiated with specific selection mechanisms to give rise to (centralized and decentralized versions of) correlated-$Q$, Nash-$Q$, friend-$Q$, and foe-$Q$ learning.

### 3.2 Specific Multiagent $Q$-Learning Algorithms

Recall that a selection mechanism $f$ is a mapping from one-shot games into (sets of) joint distributions. In particular, an equilibrium selection mechanism selects an equilibrium. For example, a correlated equilibrium selection mechanism, given a one-shot game, returns a (set of) joint distributions that satisfies Equation 1 (or, equivalently, Equation 2).

Correlated $Q$-learning is an instantiation of multiagent $Q$-learning, with the equilibrium selection mechanism $f$ defined as follows: at state $s$, given the one-shot game $Q(s)$, select an equilibrium $\pi_s$ that satisfies the following constraints: for all $i \in N$ and for all $a_i, a_i' \in A_i$,

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a) Q_i(s,a) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a) Q_i(s,(a_{-i},a_i')) \tag{24}$$

Like Equation 2, this system of inequalities is a linear program. We study four variants of correlated $Q$-learning, based on the following four objective functions, which we append to this linear program to further restrict the equilibrium selection process:

1. *utilitarian:* maximize the *sum* of all agents' rewards: at state $s$,

$$\max_{\pi_s \in \Delta(A(s))} \sum_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{25}$$

2. *egalitarian:* maximize the *minimum* of all agents' rewards: at state $s$,

$$\max_{\pi_s \in \Delta(A(s))} \min_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{26}$$

3. *plutocratic:* maximize the *maximum* of all agents' rewards: at state $s$,

$$\max_{\pi_s \in \Delta(A(s))} \max_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{27}$$

4. *dictatorial:* maximize the *maximum* of any individual agent's rewards:
   for agent $i$ and at state $s$,

$$\max_{\pi_s \in \Delta(A(s))} \sum_{a \in A(s)} \pi_s(a) Q_i(s,a) \tag{28}$$

In the discussions that follow, we abbreviate these variants of correlated-$Q$ (CE-$Q$) learning as $u$CE-$Q$, $e$CE-$Q$, $p$CE-$Q$, and $d$CE-$Q$, respectively. Note that $u$CE-$Q$, $e$CE-$Q$, and $p$CE-$Q$ are naturally implemented as centralized algorithms, whereas $d$CE-$Q$ is naturally implemented in a decentralized fashion.

Like the above variants of correlated $Q$-learning, variants of Nash-$Q$ learning can also be seen as instances of our generic multiagent $Q$-learning algorithms. For the most part,[2] we focus our study on two versions of Nash-$Q$ learning: one centralized; the other decentralized. We refer to these two algorithms as coordinated Nash-$Q$ ($c$NE-$Q$) and best Nash-$Q$ ($b$NE-$Q$), respectively. In the former, a central coordinator selects and broadcasts a Nash equilibrium to all agents; in the latter, each agent independently selects a Nash equilibrium that maximizes its own utility.

Friend-or-foe $Q$-learning also fits into our generic multiagent $Q$-learning framework. Both variants are best understood as decentralized algorithms. Friend-$Q$ agents optimize

[2] Sections 4.1 and 4.3 contain exceptions.

the dictatorial objective function, maximizing their own rewards, without enforcing the correlated equilibrium constraints expressed in Equation 24. Foe-$Q$ learning works as follows in two-player games: players 1 optimizes the following objective function:

$$\max_{\sigma_1 \in \Delta(A_1(s))} \min_{a_2 \in A_2(s)} \sum_{a_1 \in A_1(s)} \sigma_1(a_1) Q_1(s, a_1, a_2) \tag{29}$$

Player 2 optimizes analogously, and the joint distribution $\pi_s$ is the product of the marginals.

This concludes our presentation of multiagent $Q$-learning algorithms. We have described a generic framework sufficiently rich to represent correlated $Q$-learning as well as other popular multiagent $Q$-learning algorithms. In the remainder of this paper, we analyze the convergence properties of instances of multiagent $Q$-learning both theoretically (in special cases) and experimentally (on both stylized and random games). We are interested in the question of whether or not multiagent $Q$-learners learn to play stationary equilibrium policies in Markov games. It is necessary but not sufficient for $Q$-values to converge to, say, $Q^*$. Players must also play an equilibrium supported by $Q^*(s)$ at each state $s$.

## 4. Convergence of Multiagent-$Q$ Learning in Two Special Cases

In this section, we discuss two special classes of Markov games: two-player, zero-sum Markov games and common-interest Markov games. We prove that, like Nash $Q$-learning, correlated $Q$-learning behaves like foe $Q$-learning in the former class of Markov games, and like friend $Q$-learning in the latter (assuming certain objective functions).

Let $\Gamma = \langle N, A, R \rangle$ denote a *one-shot game*. A **mixed strategy profile** $(\sigma_1, \ldots, \sigma_n) \in \Delta(A_1) \times \ldots \times \Delta(A_n)$ is a profile of randomized actions, one per player. Overloading our notation, we extend $R$ to be defined over mixed strategies:

$$R_i(\sigma_1, \ldots, \sigma_n) = \sum_{a_1 \in A_1} \ldots \sum_{a_n \in A_n} \sigma_1(a_1) \ldots \sigma_n(a_n) R_i(a_1, \ldots, a_n) \tag{30}$$

and, in addition, over correlated policies $\pi \in \Delta(A)$: $R_i(\pi) = \sum_{a \in A} \pi(a) R_i(a)$. The mixed strategy profile $(\sigma_1^*, \ldots, \sigma_n^*)$ is called a **Nash equilibrium** if $\sigma_i^*$ is a **best-response** for player $i$ to its opponents' mixed strategies, for all $i \in N$: i.e.,

$$R_i(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*) = \max_{\sigma_i} R_i(\sigma_1^*, \ldots, \sigma_i, \ldots \sigma_n^*) \tag{31}$$

### 4.1 Multiagent-$Q$ Learning in Two-Player, Zero-Sum Markov Games

Our first result concerns correlated $Q$-learning in two-player, zero-sum Markov games. We prove that correlated-$Q$ learns minimax equilibrium $Q$-values in such games.

Let $\Gamma = \langle N, A, R \rangle$ denote a two-player, zero-sum one-shot game. In particular, $N = \{1, 2\}$, $A = A_1 \times A_2$, and $R_i : A \to \mathbb{R}$ *s.t.* for all $a \in A$, $R_1(a) = -R_2(a)$. A **mixed strategy profile** $(\sigma_1^*, \sigma_2^*) \in \Delta(A_1) \times \Delta(A_2)$ is a **minimax equilibrium** if:

$$R_1(\sigma_1^*, \sigma_2^*) = \max_{\sigma_1} R_1(\sigma_1, \sigma_2^*) \tag{32}$$

$$R_2(\sigma_1^*, \sigma_2^*) = \max_{\sigma_2} R_2(\sigma_1^*, \sigma_2) \tag{33}$$

Observe that Nash equilibria and minimax equilibria coincide on zero-sum games.

Define $\mathrm{MM}_i(R)$ to be the minimax (equivalently, the Nash) equilibrium value of the $i$th player in a two-player, zero-sum one-shot game $\Gamma$. Similarly, define $\mathrm{CE}_i(R)$ to be the (unique) correlated equilibrium value of the $i$th player in a two-player, zero-sum one-shot game $\Gamma$. It is well-known (e.g., Forges (1990)) that $\mathrm{CE}_i(R) = \mathrm{NE}_i(R) = \mathrm{MM}_i(R)$.

We say that the **zero-sum property** holds of the $Q$-values of a Markov game $\Gamma_\gamma$ at time $t$ if $Q_1^t(s,a) = -Q_2^t(s,a)$, for all $s \in S$ and for all $a \in A(s)$. In what follows, we show that multiagent $Q$-learning preserves the zero-sum property in zero-sum Markov games, provided $Q$-values are initialized such that this property holds.

**Observation 13** *Given a two-player, zero-sum one-shot game $\Gamma$, any selection $\pi \in \Delta(A)$ yields negated values: i.e., $R_1(\pi) = -R_2(\pi)$.*

**Lemma 14** *Multiagent $Q$-learning preserves the zero-sum property in two-player, zero-sum Markov games, provided $Q$-values are initialized such that this property holds.*

**Proof** The proof is by induction on $t$. By assumption, the zero-sum property holds at time $t = 0$.

Assume the zero-sum property holds at time $t$; we show that the property is preserved at time $t+1$. In two-player games, multiagent $Q$-learning updates $Q$-values as follows: assuming action profile $a$ is played at state $s$ and the game transitions to state $s'$,

$$\begin{aligned}
\pi_{s'}^{t+1} &\in f(Q^t(s')) && (34)\\
V_1^{t+1}(s') &:= \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_1^t(s',a') && (35)\\
V_2^{t+1}(s') &:= \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_2^t(s',a') && (36)\\
Q_1^{t+1}(s,a) &:= (1-\alpha)Q_1^t(s,a) + \alpha((1-\gamma)R_1(s,a) + \gamma V_1^{t+1}(s')) && (37)\\
Q_2^{t+1}(s,a) &:= (1-\alpha)Q_2^t(s,a) + \alpha((1-\gamma)R_2(s,a) + \gamma V_2^{t+1}(s')) && (38)
\end{aligned}$$

where $f$ is any selection mechanism. By the induction hypothesis, $Q^t(s')$ is a zero-sum one-shot game. Hence, by Observation 13, $V \equiv V_1^{t+1}(s') = -V_2^{t+1}(s') \equiv -V$, so that the multiagent-$Q$ learning update procedure simplifies as follows:

$$\begin{aligned}
Q_1^{t+1}(s,a) &:= (1-\alpha)Q_1^t(s,a) + \alpha((1-\gamma)R_1(s,a) + \gamma V) && (39)\\
Q_2^{t+1}(s,a) &:= (1-\alpha)Q_2^t(s,a) + \alpha((1-\gamma)R_2(s,a) - \gamma V) && (40)
\end{aligned}$$

Now (i) by the induction hypothesis, $Q_1^t(s,a) = -Q_2^t(s,a)$; (ii) the Markov game is zero-sum: i.e., $R_1(s,a) = -R_2(s,a)$. Therefore, $Q_1^{t+1}(s,a) = -Q_2^{t+1}(s,a)$: i.e., the zero-sum property is preserved at time $t+1$. ■

**Theorem 15** *If all $Q$-values are initialized such that the zero-sum property holds, then correlated and Nash $Q$-learning both converge to foe (i.e., minimax equilibrium) $Q$-values in two-player, zero-sum Markov games.*

**Proof** By Lemma 14, correlated and Nash $Q$-learning both preserve the zero-sum property: in particular, at time $t$, $Q_1^t(s,a) = -Q_2^t(s,a)$, for all $s \in S$ and for all $a \in A(s)$. Thus, they simplify as follows: assuming action profile $a$ is played at state $s$ and the game transitions to state $s'$, for all $i \in \{1,2\}$,

$$\begin{aligned} Q_i^{t+1}(s,a) &:= (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \mathrm{CE}_i(Q^t(s'))) && (41) \\ &:= (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \mathrm{NE}_i(Q^t(s'))) && (42) \\ &= (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \mathrm{MM}_i(Q^t(s'))) && (43) \end{aligned}$$

Indeed, the correlated, Nash, and foe $Q$-learning update procedures coincide, so that all three converge to foe (i.e., minimax) $Q$-values in two-player, zero-sum Markov games, if all $Q$-values are initialized such that the zero-sum property holds. ■

In summary, correlated and Nash $Q$-learning both converge in two-player, zero-sum Markov games. In particular, they converge to precisely the minimax equilibrium $Q$-values.

### 4.2 Multiagent-$Q$ Learning in Common-Interest Markov Games

In a common-interest one-shot game $\Gamma$, for all $i,j \in N$ and for all $a \in A$, it is the case that $R_i(a) = R_j(a)$. More generally, in a common-interest Markov game $\Gamma_\gamma$, the one-shot game defined at each state is common-interest: i.e., for all $i,j \in N$, for all $s \in S$, and for all $a \in A(s)$, it is the case that $R_i(s,a) = R_j(s,a)$.

We say that the **common-interest property** holds of the $Q$-values of a Markov game at time $t$ if $Q_i^t(s,a) = Q_j^t(s,a)$, for all $i,j \in N$, for all $s \in S$, and for all $a \in A(s)$. In what follows, we show that multiagent $Q$-learning preserves the common-interest property in common-interest Markov games, if $Q$-values are initialized such that this property holds.

**Observation 16** *Given a common-interest one-shot game $\Gamma$, any selection $\pi \in \Delta(A)$ yields common values: i.e., $R_i(\pi) = R_j(\pi)$, for all $i,j \in N$.*

**Lemma 17** *Multiagent $Q$-learning preserves the common-interest property in common-interest Markov games, provided $Q$-values are initialized such that this property holds.*

**Proof** The proof is by induction on $t$. By assumption, the common-interest property holds at time $t = 0$.

Assume the common-interest property holds at time $t$; we show that this property is preserved at time $t+1$. Multiagent $Q$-learning updates $Q$-values as follows: assuming action profile $a$ is played at state $s$ and the game transitions to state $s'$,

$$\begin{aligned} \pi_{s'}^{t+1} &\in f(Q^t(s')) && (44) \\ V_i^{t+1}(s') &:= \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_i^t(s',a') && (45) \\ Q_i^{t+1}(s,a) &:= (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma V_i^{t+1}(s')) && (46) \end{aligned}$$

where $f$ is any selection mechanism. By the induction hypothesis, $Q^t(s')$ is a common-interest one-shot game. Hence, by Observation 16, $V_i^{t+1}(s') = V_j^{t+1}(s') \equiv V$, for all $i,j \in N$, so that the multiagent-$Q$ learning update procedure simplifies as follows:

$$Q_i^{t+1}(s,a) \quad := \quad (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma V) \tag{47}$$

Now, for all $i, j \in N$, (i) by the induction hypothesis, $Q_i^t(s,a) = Q_j^t(s,a)$; (ii) the Markov game is common-interest: i.e., $R_i(s,a) = R_j(s,a)$. Therefore, $Q_i^{t+1}(s,a) = Q_j^{t+1}(s,a)$, for all $i, j \in N$: i.e., the common-interest property is preserved at time $t+1$. ■

Given a one-shot game, an equilibrium $\pi^*$ is called **Pareto-optimal** if there does not exist another equilibrium $\pi$ such that (i) for all $i \in N$, $R_i(\pi) \geq R_i(\pi^*)$, and (ii) there exists $i \in N$ such that $R_i(\pi) > R_i(\pi^*)$.

**Observation 18** *In a common-interest one-shot game $\Gamma$, for any equilibrium $\pi \in \Delta(A)$ that is Pareto-optimal, $R_i(\pi) = \max_{a \in A} R_i(a)$, for all $i \in N$.*

**Theorem 19** *If all $Q$-values are initialized such that the common-interest property holds, then any multiagent $Q$-learning algorithm that selects Pareto-optimal equilibria converges to friend $Q$-values in common-interest Markov games.*

**Proof** By Lemma 17, the common-interest property is preserved by correlated $Q$-learning: in particular, at time $t$, $Q_i^t(s,a) = Q_j^t(s,a)$, for all $i, j \in N$, for all $s \in S$, and for all $a \in A(s)$. By Observation 18, any multiagent-$Q$ learning update procedure with a Pareto-optimal selection mechanism simplifies as follows: for all $i \in N$,

$$Q_i^{t+1}(s,a) \quad := \quad (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \max_{a' \in A(s')} Q_i^t(s',a')) \tag{48}$$

assuming action profile $a$ is played at state $s$ and the game transitions to state $s'$. Indeed, the correlated and friend $Q$-learning update procedures coincide, so that correlated $Q$-learning converges to friend $Q$-values in common-interest Markov games, if all $Q$-values are initialized such that the common-interest property holds. ■

The following corollary follows immediately, since friend $Q$-learning converges to Pareto-optimal equilibrium $Q$-values in common-interest Markov games (Littman, 2001).

**Corollary 20** *If all $Q$-values are initialized such that the common-interest property holds, then any multiagent $Q$-learning algorithm that selects Pareto-optimal equilibria converges to Pareto-optimal equilibrium $Q$-values in common-interest Markov games.*

The multiagent $Q$-learning algorithms $u$CE-$Q$, $e$CE-$Q$, $p$CE-$Q$, $d$CE-$Q$, and $b$NE-$Q$ all rely on Pareto-optimal equilibrium selection operators, and thus converge to Pareto-optimal equilibrium $Q$-values in common-interest Markov games, provided $Q$-values are initialized such that the common-interest property holds.

| $R$ | $b$ | $s$ |
|---|---|---|
| $B$ | $2,1$ | $0,0$ |
| $S$ | $0,0$ | $1,2$ |

| $Q^*$ | $b$ | $s$ |
|---|---|---|
| $B$ | $2,1$ | $1,\frac{1}{2}$ |
| $S$ | $1,\frac{1}{2}$ | $\frac{3}{2},\frac{3}{2}$ |

| $\pi^*$ | $b$ | $s$ |
|---|---|---|
| $B$ | $1$ | $0$ |
| $S$ | $0$ | $0$ |

Figure 1: Sample one-shot game: Bach vs. Stravinsky. Utilitarian correlated equilibrium $Q$-values and policies ($\gamma = \frac{1}{2}$).

### 4.3 Exchangeability vs. Miscoordination

To guarantee that agents play equilibrium policies in a general-sum Markov game, it is necessary but not sufficient for agents to learn equilibrium $Q$-values. Further, the agents must play an equilibrium at every state they encounter: i.e., in each of the one-shot games $Q^*(s)$, where $Q^*(s)$ is the set of $Q$-values the agents learn at state $s \in S$.

Our theoretical convergence guarantees apply to both centralized and decentralized versions of multiagent $Q$-learning.[3] To guarantee that agents *play* equilibrium policies, not just *learn* equilibrium $Q$-values, however, may require that play be centralized (i.e., referee-guided) to avoid miscoordination in the presence of multiple equilibria.

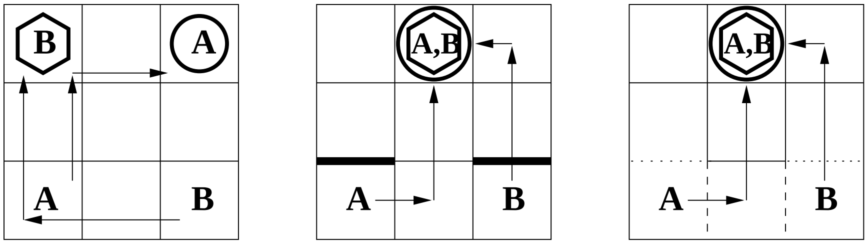
For example, in the repeated Bach or Stravinsky game with $\gamma = \frac{1}{2}$, utilitarian correlated $Q$-learning converges to the $Q$-values shown in Figure 1. Two agents playing this game, making independent decisions, can fail to coordinate their behavior, if, say, player 1 selects and plays her part of the equilibrium $(B, b)$ and player 2 selects and plays his part of the equilibrium $(S, s)$, so that the action profile the agents play is $(B, s)$.

In the case of two-player, zero-sum Markov games, however, miscoordination is ruled out by what we call the "Nash marginals" property together with the fact that Nash equilibria are "exchangeable" in this special class of games.

- The "Nash marginals" property holds of a solution concept in a one-shot game if, assuming each player $i$ selects a joint distribution $\pi_i$ according to that solution concept, but plays his marginal distribution, call it $\pi_{A_i}$, the mixed strategy profile $(\pi_{A_i})_{i \in I}$ is a Nash equilibrium. Forges (1990) establishes that this property holds of correlated equilibria in two-player, zero-sum one-shot games.
- It is well known that Nash equilibria are "exchangeable" in the following sense (see, for example, Osborne and Rubinstein (1994)): If $\Gamma$ is a two-player, zero-sum one-shot game with Nash equilibria $(\sigma_1, \sigma_2)$ and $(\sigma_1', \sigma_2')$, then $(\sigma_1, \sigma_2')$ is a Nash equilibrium.

It follows that decentralized correlated and Nash $Q$-learning suffice to lead to equilibrium *play* in two-player, zero-sum Markov games, although miscoordination is certainly possible among decentralized learners in more general settings.

[3] Note that our results also hold for "correlated value iteration," that is, synchronous updating of $Q$-values based on a correlated equilibrium selection mechanism: in two-player, zero-sum Markov games, correlated value iteration converges to foe $Q$-values; in common-interest Markov games, Pareto-optimal correlated value iteration converges to friend $Q$-values.



(a) Grid Game 1 (b) Grid Game 2 (c) Grid Game 3

Figure 2: Grid games: Initial States and Sample Equilibria. Shapes indicate goals.

## 5. Implementation Details

In the next several sections, we describe experiments with various multiagent $Q$-learning algorithms on a standard test bed of Markov games, which includes three grid games and grid soccer,[4] and on a set of randomly generated games. In doing so, we compare the performance of the aforementioned variants of correlated $Q$-learning with other well-known multiagent $Q$-learning algorithms described in the literature: specifically, minimax (or foe) $Q$-learning, friend $Q$-learning, and the two aforementioned variants of Nash $Q$-learning. We investigate the question of whether or not these multiagent $Q$-learning algorithms converge to equilibrium policies in Markov games.

Three out of four of our correlated $Q$-learning implementations are centralized: utilitarian, egalitarian, and plutocratic. Only the dictatorial variant is decentralized: each agent learns as if it is the dictator. Recall that the coordinated Nash-$Q$ algorithm computes one Nash equilibrium for all agents. Accordingly, our implementation of coordinated Nash-$Q$ is centralized. In contrast, best Nash-$Q$ is decentralized: each agent independently selects one of its preferred Nash equilibria. Lastly, we implemented decentralized FF-$Q$ learning. Convergence of $Q$-values is guaranteed for FF-$Q$ even in the decentralized setting; convergence of play, however, can only be guaranteed for friend-$Q$ in the centralized setting.

In an environment of multiple learners, off-policy (single agent) $Q$-learners are unlikely to converge to an equilibrium policy. Each agent would learn a best-response to the random behavior of the other agents, rather than a best-response to intelligent behavior on the part of the other agents. Hence, as a first point of comparison, we implemented on-policy $Q$-learning (Sutton and Barto, 1998). In our implementation of if ever the optimal action is not unique, an agent randomizes uniformly among all its optimal actions. Otherwise, $Q$-learning can easily perform arbitrarily badly in games with multiple coordination equilibria, all of equivalent value, by failing to coordinate their behavior.

## 6. Grid Games

The first set of detailed experimental results on which we report pertain to grid games. We describe three grid games, all of which are two-player, general-sum Markov games: grid game 1 (GG1) (Hu and Wellman, 2003), a multi-state coordination game; grid game 2 (GG2) (Hu

[4] A summary of our experiments with these four grid games appeared in Greenwald and Hall (2003).

and Wellman, 2003), a stochastic game that is reminiscent of Bach or Stravinsky; and grid game 3 (GG3) (Greenwald and Hall, 2003), a multi-state version of Chicken.[5] In fact, only GG2 is inherently stochastic. In the next section, we describe experiments with a simple version of soccer, a two-player, zero-sum Markov game, that is highly stochastic.

Figure 2 depicts the initial states of GG1, GG2, and GG3. All three games involve two agents and two (possibly overlapping) goals. If ever an agent reaches its goal, it scores some points, and the game ends. The agents' action sets include one step in any of the four compass directions. Actions are executed simultaneously, which implies that both agents can score in the same game instance. If both agents attempt to move into the same cell *and this cell is not an overlapping goal*, their moves fail (that is, the agents positions do not change), and they both lose 1 point in GG1 and GG2 and 50 points in GG3.

In GG1, there are two distinct goals, each worth 100 points. In GG2, there is one goal worth 100 points and two barriers: if an agent attempts to move through one of the barriers, then with probability 1/2 this move fails. In GG3, like GG2, there is one goal worth 100 points, but there are no stochastic transitions and the reward structure differs: At the start, if both agents avoid the center state by moving up the sides, they are each rewarded with 20 points; in addition, any agent that chooses the center state is rewarded with 25 points (NB: if both agents choose the center state, they collide, each earning $-25 = 25 - 50$).

## 6.1 Grid Game Equilibria

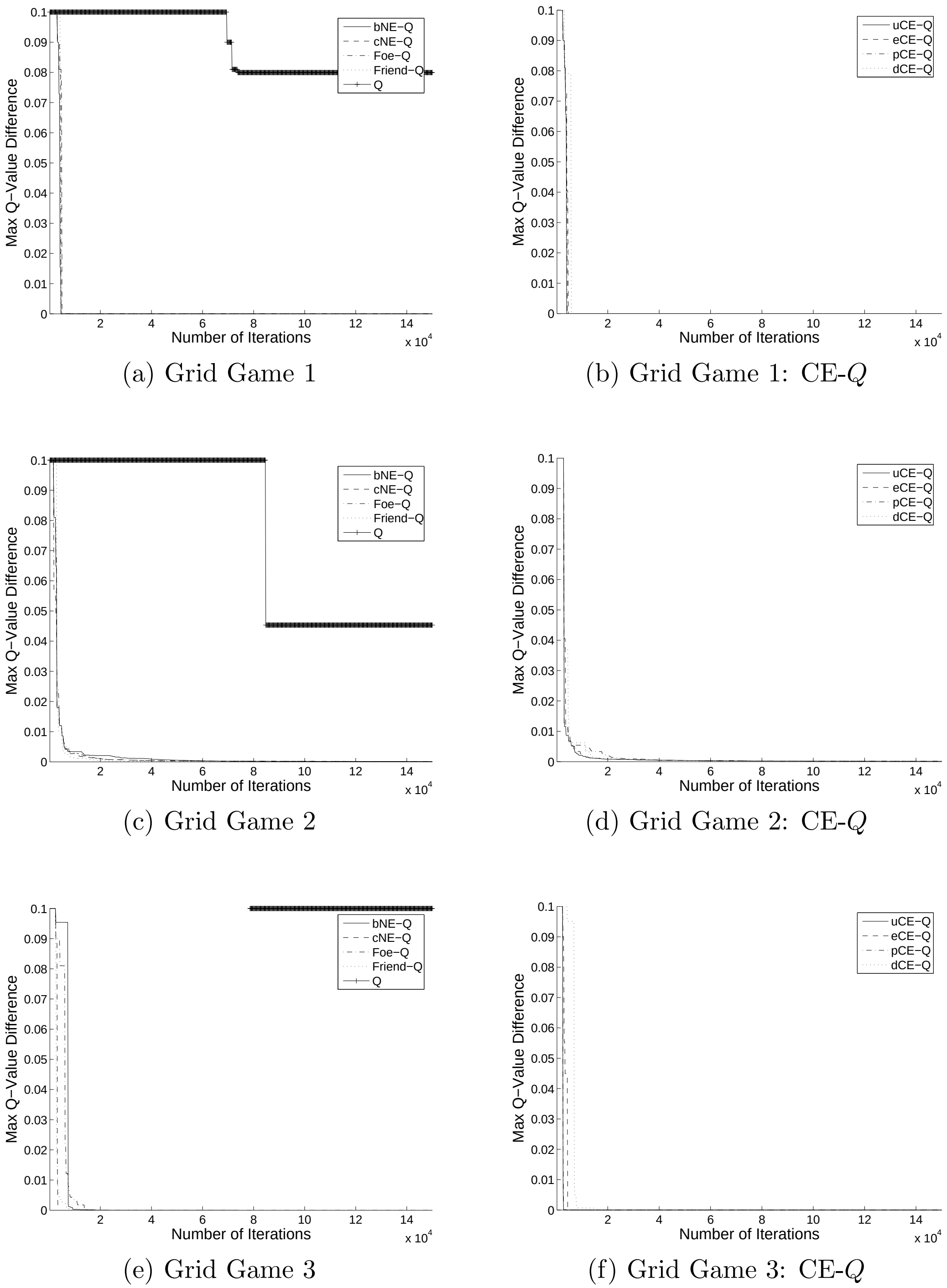
In all three grid games, there exist pure strategy stationary correlated, and hence Nash, equilibrium policies for both agents. In GG1, there are several pairs of pure strategy equilibrium policies in which the agents coordinate their behavior (see Hu and Wellman (2003) for graphical depictions). In GG2 and GG3, there are exactly two pure strategy equilibrium policies: one agent moves up the center and the other moves up the side, and the same again with the agents' roles reversed. These equilibria are asymmetric: in GG2, the agent that moves up the center scores 100, but the agent that moves up the sides scores only 50 on average (due to the 50% chance of crossing the barrier); in GG3, the agent that moves up the center scores 125, but the agent that moves up the sides scores only 100.

Since there are multiple pure strategy stationary equilibrium policies in these grid games, it is possible to construct additional stationary equilibrium policies as convex combinations of the pure policies. In GG2, there exists a continuum of symmetric correlated equilibrium policies: i.e., for all $p \in [0, 1]$, with probability $p$ one agent moves up the center and the other attempts to pass through the barrier, and with probability $1 - p$ the agents' roles are reversed. In GG3, there exists a symmetric correlated equilibrium policy in which both agents move up the sides with high probability and each of the pure strategy equilibrium policies is played with equally low probability. Do multiagent $Q$-learners learn to play these stationary equilibrium policies? We investigate this question presently.

## 6.2 Empirical Convergence

Our experiments reveal that all of the multiagent $Q$-learning algorithms in our test suite are converging in the three grid games. However, $Q$-learning itself does not converge in any

[5] Chicken is a game played by two people driving cars. Each driver can either drive straight ahead, and risk his life, or swerve out of the way, and risk embarrassment.



(a) Grid Game 1

(b) Grid Game 1: CE-$Q$

(c) Grid Game 2

(d) Grid Game 2: CE-$Q$

(e) Grid Game 3

(f) Grid Game 3: CE-$Q$

Figure 3: Changing $Q$-values in the grid games: all algorithms except $Q$-learning in all games and $e$CE-$Q$ in GG1 are converging. For all algorithms except $Q$-learning: in GG1 and GG3, where there is no stochasticity, $\alpha = 1$; in GG2, $\alpha = 1/n(s,a)$, where $n(s,a)$ is the number of visits to state-action pair $(s,a)$. Our $Q$-learning implementation is on-policy and $\epsilon$-greedy with $\epsilon = 0.01$ and $\alpha = 1/n(s,a)$.

of the grid games. Littman (2001) proves that FF-$Q$ converges in all general-sum Markov games. Hu and Wellman (2003) show empirically that both variants of NE-$Q$ converge in both GG1 and GG2. Figure 3 shows that both variants of NE-$Q$ are also converging in GG3. Similarly, all four variants of CE-$Q$ converge in all three grid games.

The values plotted in Figure 3 are computed as follows. Define an error term $\text{ERR}_i^t$ at time $t$ for agent $i$ as the difference between $Q(s^t, a^t)$ at time $t$ and $Q(s^t, a^t)$ at time $t-1$: i.e., $\text{ERR}_i^t = \left|Q_i^t(s^t, a^t) - Q_i^{t-1}(s^t, a^t)\right|$. The error values on the $y$-axis depict the maximum error from the current time $x$ to the end of the simulation $T$: i.e., $\max_{t=x,\ldots,T} \text{ERR}_i^t$, for $i = 1$. The values on the $x$-axis, representing time, range from 1 to $T'$, for some $T' < T$.[6] In our experiments, we set $T' = 1.5 \times 10^5$ and $T = 2 \times 10^5$. The maximum change in $Q$-values is converging to 0 for all algorithms except $Q$-learning in all games.

In our experiments, the parameters are set as follows. Our implementation of $Q$-learning is on-policy and $\epsilon$-greedy, with $\epsilon = 0.01$ and $\alpha = 1/n(s,a)$, where $n(s,a)$ is the number of visits to state-action pair $(s,a)$. All other algorithms are off-policy (equivalently, on-policy and $\epsilon$-greedy with $\epsilon = 1$). For these off-policy learning algorithms, in GG1 and GG3, where there is no stochasticity, $\alpha = 1$; in GG2, however, like $Q$-learning, $\alpha = 1/n(s,a)$. Finally, $\gamma = 0.9$ in all cases. Next, we investigate the policies learned by the algorithms.

### 6.3 Equilibrium Policies

We now address the question: what is it that the $Q$-learning algorithms learn? In summary,

- $Q$-learning does not converge, and it does not learn equilibrium policies;
- friend and foe $Q$-learning converge, but need not learn equilibrium policies;
- and both variants of NE-$Q$ and all four variants of CE-$Q$ learn equilibrium policies.

To address this question, we analyzed the agents' policies at the end of each simulation by appending to the learning phase an auxiliary testing phase in which the agents play the games repeatedly according to the policies they learned. Our learning phase is randomized: not only are the state transitions stochastic, on-policy $Q$-learners and off-policy multiagent $Q$-learners can all make probabilistic decisions. Thus, if there exist multiple equilibrium policies in a game, agents can learn different equilibrium policies across different runs. Moreover, since agents can learn stochastic policies, scores can vary across different test runs. Nonetheless, we presented only one run of the learning phase (see Section 6.2) and here we present only one test run, each of which is representative of their respective sets of possible outcomes. The results of our testing phase are depicted in Table 3.

**$Q$-Learning** Although our implementation of on-policy $Q$-learning does not converge in the grid games, the policies appeared stable at the end of the learning phase. (By decaying $\alpha$, we disallow large changes in the agents' $Q$-values, which makes changes in their policies less and less frequent.) However, we still contend that $Q$-learning is not successful in the grid games: it happens to learn equilibrium policies in GG1, but in GG2, it learns a foe-$Q$-like

[6] Setting $T' = T$ is sometimes misleading: It could appear that non-convergent algorithms are converging, because our metric measures the maximum error between the current time and the end of the simulation, but it could be that the change in $Q$-values is negligible for all states visited at the end of the simulation.

| GG1 | Avg. Score | Games | Convergence? | Eqm. Values? | Eqm. Play? |
|---|---|---|---|---|---|
| $Q$ | 100,100 | 2500 | No | Yes | Yes |
| Foe-$Q$ | 0,0 | 0 | Yes | No | No |
| Friend-$Q$ | $-3239, -3239$ | 3 | Yes | Yes | No |
| $u$CE-$Q$ | 100,100 | 2500 | Yes | Yes | Yes |
| $e$CE-$Q$ | 100,100 | 2500 | Yes | Yes | Yes |
| $p$CE-$Q$ | 100,100 | 2500 | Yes | Yes | Yes |
| $c$NE-$Q$ | 100,100 | 2500 | Yes | Yes | Yes |
| $d$CE-$Q$ | $-10^4, -10^4$ | 0 | Yes | Yes | No |
| $b$NE-$Q$ | $-10^4, -10^4$ | 0 | Yes | Yes | No |

| GG2 | Avg. Score | Games | Convergence? | Eqm. Values? | Eqm. Play? |
|---|---|---|---|---|---|
| $Q$ | 67.3,66.2 | 3008 | No | No | No |
| Foe-$Q$ | 65.9,67.4 | 3011 | Yes | No | No |
| Friend-$Q$ | $-10^4, -10^4$ | 0 | Yes | No | No |
| $u$CE-$Q$ | 50.4,100 | 3333 | Yes | Yes | Yes |
| $e$CE-$Q$ | 49.5,100 | 3333 | Yes | Yes | Yes |
| $p$CE-$Q$ | 50.3,100 | 3333 | Yes | Yes | Yes |
| $c$NE-$Q$ | 100,50.2 | 3333 | Yes | Yes | Yes |
| $d$CE-$Q$ | 49.9,100 | 3333 | Yes | Yes | Yes |
| $b$NE-$Q$ | 100,49.7 | 3333 | Yes | Yes | Yes |

| GG3 | Avg. Score | Games | Convergence? | Eqm. Values? | Eqm. Play? |
|---|---|---|---|---|---|
| $Q$ | 62.6,95.4 | 3314 | No | No | No |
| Foe-$Q$ | 120,120 | 3333 | Yes | No | No |
| Friend-$Q$ | $-25 \times 10^4, -25 \times 10^4$ | 0 | Yes | No | No |
| $u$CE-$Q$ | 117,117 | 3333 | Yes | Yes | Yes |
| $e$CE-$Q$ | 117,117 | 3333 | Yes | Yes | Yes |
| $p$CE-$Q$ | 100,125 | 3333 | Yes | Yes | Yes |
| $c$NE-$Q$ | 125,100 | 3333 | Yes | Yes | Yes |
| $d$CE-$Q$ | $-25 \times 10^4, -25 \times 10^4$ | 0 | Yes | Yes | No |
| $b$NE-$Q$ | $-25 \times 10^4, -25 \times 10^4$ | 0 | Yes | Yes | No |

Table 3: Testing phase: Grid games played repeatedly. Average scores across $10^4$ moves are shown. The number of games played varied with the agents' policies: sometimes agents moved directly to the goal; other times they digressed. For each learning algorithm, the Convergence? column states whether or not the $Q$-values converge; the Equilibrium Values? column states whether or not any convergent $Q$-values correspond to an equilibrium policy; the Equilibrium Play? column states whether or not the trajectories of play during testing correspond to an equilibrium policy.

(non-equilibrium) policy. As expected, we have found that the $Q$-learning algorithm does not converge in general; moreover, the more or less stable policies to which it "converges" need not be equilibrium policies. Indeed, this observation is the underlying motivation for multiagent $Q$-learning research.

**Foe-$Q$** Foe $Q$-learners perform poorly in GG1. Rather than progress toward the goal, they cower in the corners, avoiding collisions, and consequently avoiding the goal. Sometimes one agent simply moves out of the way of the other, allowing its opponent to reach its goal rather than risk collision. In GG2 and GG3, the principle of avoiding collisions leads both foe $Q$-learners straight up the sides of the grid. Although these policies yield reasonable scores in GG2, and Pareto optimal scores in GG3, these are not equilibrium policies. On the contrary, foe $Q$-learning yields policies that are not rational—both agents have an incentive to deviate to the center, since the reward for using the center passage exceeds that of moving up the sides, given that one's opponent is moving up the side.

**Friend-$Q$** In GG1, friend $Q$-learning can perform even worse than foe $Q$-learning. This result may appear surprising at first glance, since GG1 satisfies the conditions under which friend $Q$-learning is guaranteed to learn equilibrium $Q$-values (Littman, 2001). Indeed, friend-$Q$ does learn $Q$-values that support equilibrium policies, but in our decentralized implementation of friend $Q$-learning, friends lack the ability to coordinate their play. Whenever these so-called "friends" choose policies that collide, both agents obtain negative scores for the remainder of the simulation: e.g., if the agents' policies lead them to one another's goals, both agents move towards the center ever after. In our experiments, friend-$Q$ learned a stochastic policy[7] at the start state that allowed it to complete a few games successfully before arriving at a state where the friendly assumption led the players to collide indefinitely. In GG2 and GG3, friend-$Q$'s performance is always poor: both agents learn equilibrium policies that use the center passage, which leads to repeated collisions.

**NE-$Q$ and CE-$Q$ Learning** In GG1, $u$CE-$Q$, $e$-CE-$Q$, $p$CE-$Q$, and $c$NE-$Q$ all learn $Q$-values that coincide exactly with those of friend-$Q$: i.e., $Q$-values that support stationary equilibrium policies. But unlike friend-$Q$, these variants of CE-$Q$ and NE-$Q$ always obtain positive scores. In our implementation of CE-$Q$, a centralized mechanism broadcasts an equilibrium policy, even during testing. Thus, play is always coordinated, and $u$CE-$Q$, $e$CE-$Q$ and $p$CE-$Q$ learners do not collide while playing the grid games. In our implementation of NE-$Q$, however, the agents are more robust during testing: they make independent decisions according to their individual policies. Still, since the $c$NE-$Q$ agents learn coordinated equilibrium policies, they manage to coordinate their play perfectly.

The dictatorial operator is one way to eliminate CE-$Q$'s dependence on a centralized mechanism; similarly, the best Nash operator eliminates NE-$Q$'s dependence on a centralized mechanism. In $d$CE-$Q$ and $b$NE-$Q$, each agent solves an independent optimization problem during learning; thus, learning is not necessarily coordinated. Like the other variants of CE-$Q$ and NE-$Q$, the $Q$-values of $d$CE-$Q$ and $b$NE-Q coincide exactly with those of friend-$Q$ in GG1. But like friend-$Q$, these agents are unable to coordinate their play. Indeed, during our testing phase, for both pairs of learners, agent $A$ played R, thinking agent $B$ would play

[7] Like $Q$-learning, in our implementation of friend $Q$-learning, if ever the optimal action is not unique, an agent randomizes uniformly among all its optimal actions.

**Grid Game 2: Start State**

| | SIDE | CENTER |
|---|---|---|
| SIDE | 4.96, 5.92 | 3.97, 7.99 |
| CENTER | 8.04, 4.02 | 3.62, 6.84 |

U, but at the same time agent $B$ played L, thinking agent $A$ would play U. Returning to the start state (again and again), the agents employed the same policy (again and again).

In GG2, all variants of CE-$Q$ and NE-$Q$ learning studied here converge to stationary equilibrium policies. Interestingly, the asynchronous updating that characterizes $Q$-learning converts this symmetric game into a dominance-solvable game:[8] The agent that scores first by playing CENTER learns that this action can yield high rewards, reinforcing its instinct to play CENTER, and leaving the other agent has no choice but to play SIDE, its best-response to CENTER. The $Q$-table below depicts the $Q$-values at the start state that were learned by $u$CE-$Q$. (The other algorithms learned similar, although possibly transposed, values.) The column player eliminates SIDE, since it is dominated, after which the row player eliminates CENTER. Thus, the equilibrium outcome is (SIDE, CENTER), as the scores indicate.

By learning similar $Q$-values, the $d$CE-$Q$ and $b$NE-$Q$ agents effectively coordinate their behavior: since the game is dominance-solvable, there is a unique pure strategy correlated, and hence Nash, equilibrium in the one-shot game specified by the $Q$-values.

In both GG1 and GG2, all variants of CE $Q$-learning are indifferent between all stationary correlated equilibrium policies, pure and mixed, since they all yield equivalent rewards to all players. In GG3, however, both $u$CE-$Q$ and $e$CE-$Q$ learn the particular correlated equilibrium policy that yields symmetric scores, because both the sum and the minimum of the agents' rewards at this equilibrium exceed those of any other equilibrium policies. Indeed, the sum of the scores of $u$CE-$Q$ and $e$CE-$Q$ exceed that of any Nash equilibrium. This sum does not exceed the sum of the foe $Q$-learners' scores, however, but foe $Q$-learners do not behave rationally. Coincident with $c$NE-$Q$, the $p$CE-$Q$ learning algorithm converges to a pure strategy equilibrium policy that is among those which maximize the maximum of all agents' rewards. Finally, each $d$CE-$Q$ and $b$NE-$Q$ agent attempts to play the equilibrium policy that maximizes its own rewards, yielding repeated collisions and negative scores.

In summary, the behavior of the centralized and decentralized variants of CE-$Q$ and NE-$Q$ coincide in GG1 and GG2. In GG3, however, while the convergence properties again coincide, in the centralized camp, $u$CE-$Q$ and $e$CE-$Q$ earn higher rewards than $c$NE-$Q$, and $p$CE-$Q$ earns comparable rewards. In these grid games at least, correlated $Q$-learning is at least as powerful as Nash $Q$-learning; moreover, its computation is far easier.

## 7. Soccer Game

In this section, we describe experiments with a simplified version of the soccer game that is described in Littman (1994). The main point of this discussion is to further demonstrate the shortcomings of basic $Q$-learning. While it is generally understood that the dynamics

[8] A one-shot game is said to be *dominance solvable* if, after iteratively deleting all dominated strategies, a unique strategy profile remains. A strategy is *dominated* if there is exists some other strategy which yields higher rewards that said strategy in all circumstances.

of multiple $Q$-learners agents need not converge, there are few examples of non-convergence in the literature (one notable exception appears in Tesauro and Kephart (1999)). Here, we study a two-player, zero-sum Markov game. Theoretical guarantees on multiagent $Q$-learning can be obtained for such games (see, Section 4). Still, we find that even in this relatively simple game, multiple $Q$-learners do not converge. It appears instead that the behavior is cycling, but this conjecture merits further study.

The soccer field is a grid (see Figure 4). There are two players, whose possible actions are N, S, E, W, and stick. Players choose their actions simultaneously. Actions are executed in random order. If the sequence of actions causes the players to collide, then only the first player moves, and only if the cell into which he is moving is unoccupied. If the player with the ball attempts to move into the player without the ball, then the ball changes possession; however, the player without the ball cannot steal the ball by attempting to move into the player with the ball.[9] Finally, if the player with the ball moves into a goal, then he scores +100 if it is in fact his own goal and the other player scores −100, or he scores −100 if it is the other player's goal and the other player scores +100. In either case, the game ends.

There are no explicit stochastic state transitions in this game's specification. However, there are "implicit" stochastic state transitions, resulting from the fact that the players actions are executed in random order. From each state, there are transitions to (at most) two subsequent states, each with probability 1/2. These subsequent states are: the state that arises when player $A$ ($B$) moves first and player $B$ ($A$) moves second.

Unlike in the grid games, in this simple soccer game, there do not exist pure stationary equilibrium policies, since at certain states there do not exist pure strategy equilibria. For example, at the state depicted in Figure 4 (hereafter, state $\hat{s}$), any pure policy for player $A$ is subject to indefinite blocking by player $B$; but if player $A$ employs a mixed policy, then player $A$ can hope to pass player $B$ on his next move.

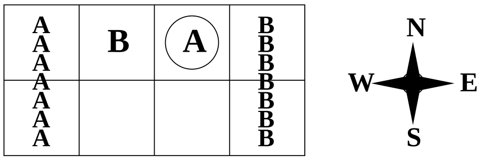
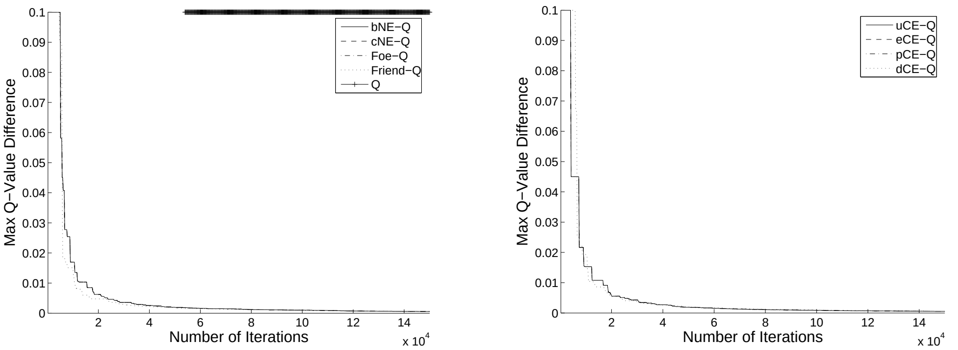


Figure 4: Soccer Game. The circle represents the ball. If player $A$ moves W, he loses the ball to player $B$; but if player $B$ moves E, attempting to steal the ball, he cannot.

### 7.1 Empirical Convergence

We experimented with the same set of $Q$-learning algorithms in this soccer game as in the grid games. Consistent with the theory of two-player, zero-sum Markov games, friend-$Q$ and foe-$Q$ converge at all state-action pairs. Moreover, both variants of Nash-$Q$ converge everywhere, as do all four variants of correlated-$Q$—in this game, all equilibria at all states

[9] This form of the game is due to Littman (1994).



(a) Grid Soccer

(b) Grid Soccer: CE-$Q$

Figure 5: Changing $Q$-values in the soccer game: all algorithms are converging, except $Q$-learning. For all algorithms, the discount factor $\gamma = 0.9$ and the parameter $\alpha = 1/n(s,a)$, where $n(s,a)$ is the number of visits to state-action pair $(s,a)$. Our $Q$-learning implementation is on-policy and $\epsilon$-greedy with $\epsilon = 0.01$.

have equivalent values; thus, all selection mechanisms yield identical outcomes. Moreover, Nash-$Q$ and correlated-$Q$ learn $Q$-values that coincide exactly with those of foe-$Q$.

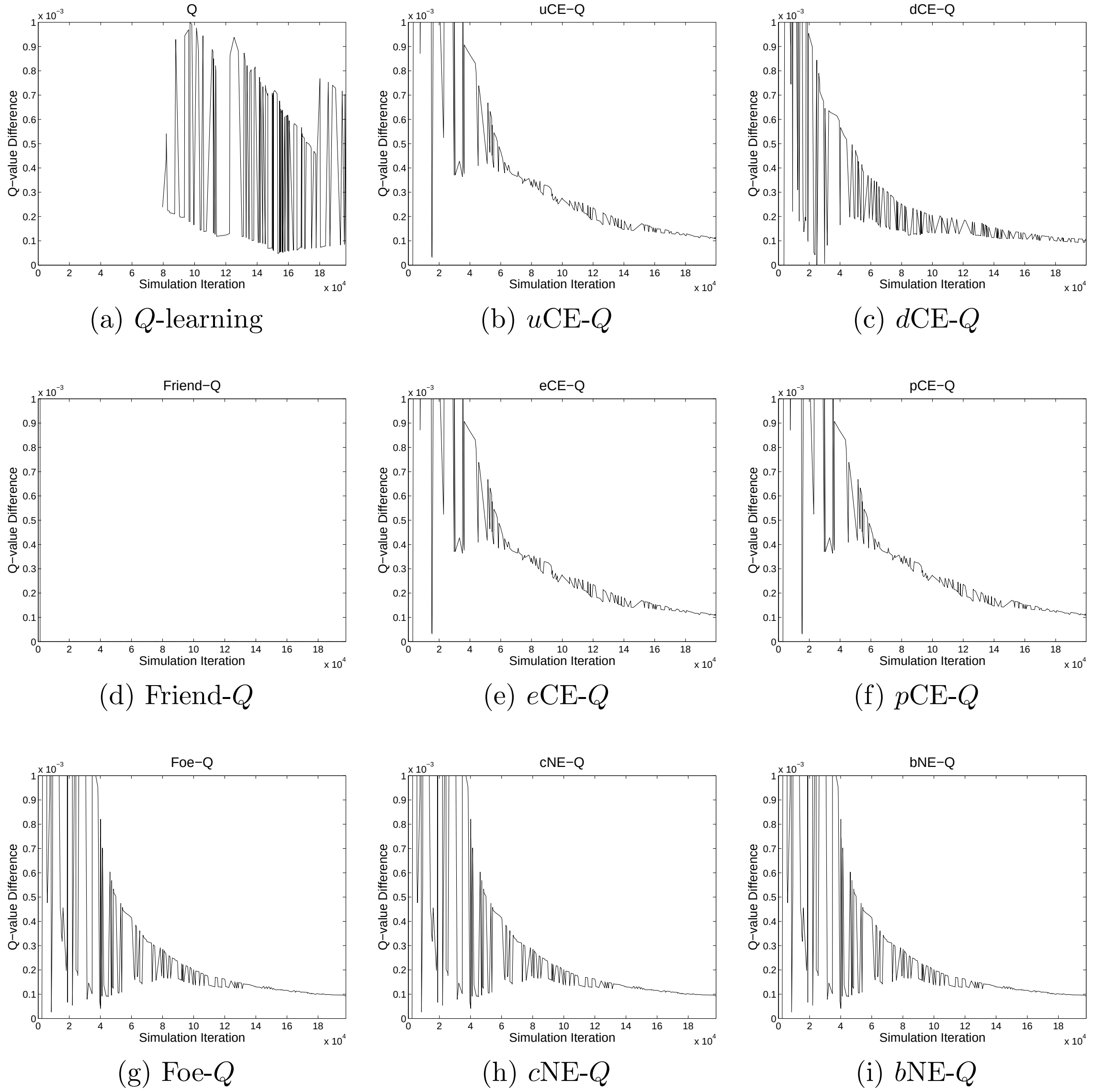
Figure 5 shows that while all the multiagent-$Q$ learning algorithms implemented converge, *$Q$-learning itself does not converge.* Our implementation of $Q$-learning is on-policy and $\epsilon$-greedy, with $\epsilon = 0.01$. The parameter $\alpha = 1/n(s,a)$, where $n(s,a)$ is the number of visits to state-action pair $(s,a)$. The discount factor $\gamma = 0.9$.

As in Figure 3, the $y$-values depict the maximum error from the current time $x$ to the end of the simulation $T$: i.e., $\max_{t=x,\ldots,T} \text{ERR}_i^t = \max_{t=x,\ldots,T} \left|Q_i^t(s^t, a^t) - Q_i^{t-1}(s^t, a^t)\right|$, for $i = A$. The values on the $x$-axis, representing time, range from 1 to $T'$, for some $T' < T$. As in our experiments with the grid games, we set $T = 1.5 \times 10^5$ and $T = 2 \times 10^5$.

Figure 6 presents an example of a state-action pair at which classic $Q$-learning does not converge. The values on the $x$-axis represent time, and the corresponding $y$-values are the error values $\text{ERR}_A^t = \left|Q_i^t(\hat{s}, \text{S,E}) - Q_i^{t-1}(\hat{s}, \text{S,E})\right|$. In Figure 6(a), although the $Q$-value differences are decreasing at times, they are not converging. They are decreasing only because the learning rate $\alpha$ is decreasing. Often times, the amplitude of the oscillations in error values is as great as the envelope of the learning rate.

Friend-$Q$, however, converges to a pure policy for player $A$ at state $\hat{s}$, namely W. Learning according to friend-$Q$, player $A$ fallaciously anticipates the following sequence of events: player $B$ sticks at state $\hat{s}$, and player $A$ takes action W. By taking action W, player $A$ passes the ball to player $B$, with the intent that player $B$ score for him. Player $B$ is indifferent among her actions, since she, again fallaciously, assumes player $A$ plans to score a goal for her immediately.

In two-player, zero-sum games, the values of all Nash equilibria, including those which are best for individual players, are equivalent. Hence, the behaviors of foe-$Q$, $c$NE-$Q$, and $b$NE-$Q$ are indistinguishable in such games. Indeed, Figures 6(g), (h), and (i) show that at state $\hat{s}$, foe-$Q$ and both variants of Nash-$Q$ converge along the same path. Moreover, foe-$Q$



(a) $Q$-learning (b) $u$CE-$Q$ (c) $d$CE-$Q$

(d) Friend-$Q$ (e) $e$CE-$Q$ (f) $p$CE-$Q$

(g) Foe-$Q$ (h) $c$NE-$Q$ (i) $b$NE-$Q$

Figure 6: Changing $Q$-values at state $\hat{s}$. All algorithms are converging, except $Q$-learning.

and both variants of Nash-$Q$ all converge to the same mixed policies for both players, with each player randomizing between sticking and heading south.

Finally, all four variants of CE-$Q$ converge. Perhaps surprisingly, these variants converge to *independent* minimax equilibrium policies at state $\hat{s}$,[10] although in principle, correlated-$Q$ can learn correlated equilibrium policies, even in two-player, zero-sum Markov games.

[10] We did not check for independent policies at all states.

| Soccer | Avg. Score | Games | Convergence? | Eqm. Values? | Eqm. Play? |
|---|---|---|---|---|---|
| $Q$ | 0, 0 | $< 1$ | No | No | No |
| Foe-$Q$ | $-1.06$, 1.06 | 4170 | Yes | Yes | Yes |
| Friend-$Q$ | 0.11, $-0.11$ | 6115 | Yes | No | No |
| $u$CE-$Q$ | 2.30, $-2.30$ | 4051 | Yes | Yes | Yes |
| $e$CE-$Q$ | 1.18, $-1.18$ | 4167 | Yes | Yes | Yes |
| $p$CE-$Q$ | 1.12, $-1.12$ | 4104 | Yes | Yes | Yes |
| $c$NE-$Q$ | 1.86, $-1.86$ | 4194 | Yes | Yes | Yes |
| $d$CE-$Q$ | $-0.24$, 0.24 | 4130 | Yes | Yes | Yes |
| $b$NE-$Q$ | 0.84, $-0.84$ | 4304 | Yes | Yes | Yes |

Table 4: Testing phase: Grid soccer played repeatedly, with random start states. Average scores across $10^4$ moves are shown. The number of games played varied with the agents' policies: sometimes agents moved directly to the goal; other times they digressed. The final three columns are analogous to those in Table 3.

### 7.2 Equilibrium Policies

In Table 4, we present the results of a testing phase for this soccer game. All players, except for $Q$-learners play a "good" game, meaning that each player wins approximately the same number of games; hence, scores are close to $0, 0$. Friend-$Q$ tends to let the other player win quickly (observe the large number of games played), and plays a "good" game only because of the symmetric nature of grid soccer. All CE-$Q$ and NE-$Q$ variants behave in a manner that is similar to one another and similar to foe-$Q$.[11]

In summary, in grid soccer, a two-player, zero-sum Markov game, $Q$-learning does not converge. Intuitively, the rationale for this outcome is clear: $Q$-learning seeks deterministic optimal policies, but in this game no such policies exist.[12] Friend-$Q$ converges but its policies are not rational. Correlated $Q$-learning, like Nash $Q$-learning, learns the same $Q$-values as foe-$Q$ learning. However, correlated-$Q$ learns possibly correlated equilibrium policies, while foe-$Q$ and Nash-$Q$ learn minimax equilibrium policies.

## 8. Random Games

Work in progress.

## 9. Related Work

While Markov games have been the subject of extensive research since the latter part of the twentieth century, multiagent reinforcement learning in Markov games has only recently received attention. In 1994, the field was launched with Littman's (1994) seminal paper on minimax $Q$-learning. The proof of convergence of this algorithm to a minimax equilibrium

[11] Any differences in scores among these algorithms is due to randomness in the simulations.

[12] Recall that in our implementation of $Q$-learning, players randomize if the action that yields the maximum $Q$-value is not unique. At any state in which playing a uniform distribution across such actions is not an equilibrium policy, $Q$-learning does not converge.

policy appeared subsequently in Littman and Szepesvári (1996). This algorithm computes the value of a state as the value to one player of the zero-sum game induced by the $Q$-values at that state. Later, $Q$-learning techniques were extended to general-sum games by Hu and Wellman (2003). Here, each state's value is computed based an arbitrary Nash equilibrium of the matrix game induced by the $Q$-values at that state. This algorithm has weak convergence guarantees (e.g., Bowling (2000)). Moreover, the computation of a Nash equilibrium, even for a bimatrix game, is PPAD-Complete Chen and Deng (2005). Finally, algorithms have also been designed for the special case of coordination games. For example, Littman's (2001) friend-$Q$ algorithm converges to equilibrium policies in this class of games. In addition, Claus and Boutilier (1998), generalize the classic game-theoretic learning method known as fictitious play (e.g., Robinson (1951)) to multiagent reinforcement learning, and apply their algorithm to this class of games. In addition to $Q$-learning algorithms, there are also model-based techniques like R-max (Brafman and Tennenholtz, 2001), which has been proven to learn near-minimax equilibrium policies in finite, average-reward, zero-sum Markov games; and policy-search techniques like WoLF (Bowling and Veloso, 2002), which has been shown empirically to converge very quickly in zero-sum Markov games.

To summarize, multiagent reinforcment learning in general-sum Markov games is an open problem: no algorithms exist to date that are guaranteed to learn an equilibrium policy of any type in arbitrary general-sum Markov games.

## 10. Conclusion

This research originated with a fixed point proof of the existence of stationary correlated equilibrium policies in general-sum Markov games (Greenwald and Zinkevich, 2005), which motivated the design of correlated $Q$-learning, an algorithm that generalizes other commonly studied multiagent $Q$-learning algorithms. Theoretically, we established that correlated $Q$-learning converges to stationary correlated equilibrium policies in two special classes of Markov games, namely zero-sum and common-interest. Empirically, we established that correlated $Q$-learning converges to stationary correlated equilibrium policies on a standard test bed of Markov games, although it does not converge in general. Our empirical findings suggest that like Nash $Q$-learning, correlated $Q$-learning can serve as an effective heuristic for the computation of equilibrium policies in general-sum Markov games.

However, we contend that correlated-$Q$ is preferred to Nash-$Q$ for two reasons:

1. Correlated equilibria in one-shot games can be computed in polynomial time; the computation of Nash equilibria in one-shot games is PPAD-complete.

2. Correlated equilibrium rewards can fall outside the convex hull of Nash equilibrium rewards; hence, all players may fare better at the former than at the latter.

In past work, we have studied adaptive algorithms for learning game-theoretic equilibria in repeated games (Greenwald and Jafari, 2003). In ongoing work, we are combining these adaptive algorithms with multiagent $Q$-learning. Specifically, our goal is to replace the linear programming call in correlated $Q$-learning with an adaptive procedure that converges to the set of correlated equilibria (e.g., Foster and Vohra (1997)). Similarly, we are studying an adaptive version of minimax-$Q$ by replacing its linear programming call with an adaptive procedure that converges to minimax equilibrium (e.g., Freund and Schapire (1996)). These adaptive approaches could simultaneously achieve an objective of artificial intelligence researchers—to learn $Q$-values—and an objective of game theory researchers—to learn game-theoretic equilibria.

Practically speaking, one of the goals of this line of research is to improve the design and implementation of multiagent systems. At one extreme, multiagent system designers act as central planners, equipping all agents in the system with specified behaviors; however, such systems are rarely compatible with agents' incentives. At the other extreme, multiagent system designers allow the agents to specify their own behavior; however, these systems are susceptible to miscoordination. A multiagent system design based on the correlated equilibrium solution concept would perhaps rely on a central planner (i.e., the referee), but nonetheless, would specify rational agent behavior. Such a design would not only facilitate multiagent coordination, but could generate greater rewards to the agents than any multiagent system design based on the Nash equilibrium solution concept.

## Acknowledgments

The authors are grateful to Michael Littman for inspiring discussions. We also thank Dinah Rosenberg and Roberto Serrano for comments on an earlier draft of this paper. This research was supported by NSF Career Grant #IIS-0133689.

## References

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

M. Bowling. Convergence problems of general-sum multiagent reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 89–94. Morgan Kaufman, 2000.

Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

Ronen I. Brafman and Moshe Tennenholtz. R-MAX — a general polynomial time algorithm for near-optimal reinforcement learning. In *IJCAI*, pages 953–958, 2001.

Xi Chen and Xiaotie Deng. Settling the complexity of 2-player nash equilibrium. Technical Report 140, Electronic Colloquium on Computational Complexity, 2005.

C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, June 1998.

F. Forges. Correlated equilibrium in two-person zero-sum games. *Econometrica*, 58(2):515, 1990.

D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 21:40–55, 1997.

Y. Freund and R. Schapire. Game theory, on-line prediction, and boosting. In *Proceedings of the 9th Annual Conference on Computational Learning Theory*, pages 325–332. ACM Press, May 1996.

A. Greenwald and K. Hall. Correlated $Q$-learning. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 242–249, 2003.

A. Greenwald and A. Jafari. A general class of no-regret algorithms and game-theoretic equilibria. In *Proceedings of the 2003 Computational Learning Theory Conference*, pages 1–11, August 2003.

A. Greenwald and M. Zinkevich. A direct proof of the existence of correlated equilibrium policies in general-sum markov games. Technical Report CS-05-07, Brown University, Department of Computer Science, June 2005.

J. Hu and M. Wellman. Nash $Q$-learning for general-sum stochastic games. *Machine Learning Research*, 4:1039–1069, 2003.

M. Littman. Friend or foe $Q$-learning in general-sum Markov games. In *Proceedings of Eighteenth International Conference on Machine Learning*, pages 322–328, June 2001.

M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163, July 1994.

M. Littman and C. Szepesvári. A generalized reinforcement learning model: Convergence and applications. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 310–318, 1996.

J. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

M. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, Cambridge, 1994.

M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley-Interscience, 1994.

J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:298–301, 1951.

L.S. Shapley. A value for $n$-person games. In H. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games*, volume II, pages 307–317. Princeton University Press, 1953.

R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Massachusetts, 1998.

G.J. Tesauro and J.O. Kephart. Pricing in agent economies using multi-agent $Q$-learning. In *Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 71–86, July 1999.