

Correlated Q -Learning

Amy Greenwald

*Department of Computer Science
Brown University
Providence, RI 02912*

AMY@BROWN.EDU

Keith Hall

*Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218*

KEITH_HALL@JHU.EDU

Martin Zinkevich

*Computing Science Department
University of Alberta
Edmonton, AB T5J3P1 Canada*

MAZ@CS.BROWN.EDU

Editor: Leslie Pack Kaelbling

Abstract

There have been several attempts to design multiagent Q -learning algorithms capable of learning 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

There have been several attempts to design multiagent Q -learning algorithms capable of learning 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 multi-agent 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 primary 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 soccer, a two-player, zero-sum Markov game for which multiagent Q -learning is guaranteed to converge, the behavior of multiple traditional Q -learning agents fails to converge “readily” (i.e., after millions of iterations).
2. In three stylized general-sum Markov games, the aforementioned variants of correlated- Q perform on par with related variants of Nash- Q . The former outperforms the latter only in one 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 aforementioned variants of both correlated and Nash Q -learning often fail to converge. Hence, we introduce another selection mechanism—“closest match (*cm*)”—that emphasizes stability and successfully converges more often than the others. Further, we find that *cm*CE- Q converges to stationary equilibrium policies more often than its Nash counterpart.

This paper is organized as follows. In Section 2, we review the definitions of correlated equilibrium in one-shot games and correlated equilibrium *policies* in Markov games. In Section 3, we define two generic 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 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 (and Nash) Q -learning are guaranteed to converge to stationary equilibrium policies. In the remainder of the paper, we describe simulation experiments that compare variants of correlated Q -learning with traditional Q -learning, FF- Q , and 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 $s \in S$, $a \in A(s)$, $P[s' | s, a] \geq 0$ and $\sum_{s' \in S} P[s' | s, a] = 1$; we interpret $P[s' | s, a]$ as the probability that the next state is s' given that the current state is s and the current action profile is a
- $R : \mathcal{A} \rightarrow [\alpha, \beta]^n$, 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 $t = 1, 2, \dots$, the players and the referee observe the current game state $s^t \in S$; following its policy π , the referee selects the distribution π_{s^t} , based on which it recommends an action, say a_i^t , to each player i ; given its recommendation, each player selects an action a_i^t , and the pure action profile $a^t = (a_1^t, \dots, a_n^t)$ 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 $P[s^{t+1} | s^t, a^t]$; 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 $R : A \rightarrow \mathbb{R}^n$, 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 $\pi(a_{-i} | a_i) = \frac{\pi(a_{-i}, a_i)}{\pi(a_i)}$ whenever $\pi(a_i) > 0$.

Definition 2 *Given a one-shot game Γ , 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 $a'_i \in A_i$,*

$$\sum_{a_{-i} \in A_{-i}} \pi(a_{-i} | a_i) R_i(a_{-i}, a_i) \geq \sum_{a_{-i} \in A_{-i}} \pi(a_{-i} | a_i) R_i(a_{-i}, a'_i) \quad (1)$$

If the referee chooses a according to correlated equilibrium policy π , then the players are motivated to follow his advice, because the expression $\sum_{a_{-i} \in A_{-i}} \pi(a_{-i} | a_i) R(a_{-i}, a'_i)$ 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 $a_i, a'_i \in A_i$,

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

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 Γ , $R(a_{-i}, a_i)$ 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 $a_{-i} \in A_{-i}$, $\pi(a_{-i} | a_i) = \pi(a_{-i} | a'_i)$, 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 Γ_γ , a referee's policy π 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 Γ_γ , and a referee's policy π , consider the transition matrix T^π such that $T_{ss'}^\pi$ is the probability of transitioning to state s' from state s , given that the referee selects an action profile according to the distribution π_s that the agents indeed follow:

$$T_{ss'}^\pi = \sum_{a \in A(s)} \pi_s(a) P[s' | s, a] \quad (3)$$

Exponentiating this matrix, the probability of transitioning to state s' from state s after t time steps is given by $(T_{ss'}^\pi)^t$. 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 π :

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

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 π is followed thereafter:

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

The normalization constant $1 - \gamma$ ensures that the ranges of V_i^π and Q_i^π 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 π is stationary by assumption.)

Theorem 5 *Given a Markov game Γ_γ , for any $V : S \rightarrow [\alpha, \beta]^n$, for any $Q : A \rightarrow [\alpha, \beta]^n$, and for any stationary policy π , $V = V^\pi$ and $Q = Q^\pi$ if and only if for all $i \in N$,*

$$V_i(s) = \sum_{a \in A(s)} \pi_s(a) Q_i(s, a) \quad (6)$$

$$Q_i(s, a) = (1 - \gamma) R_i(s, a) + \gamma \sum_{s' \in S} P[s' | s, a] V_i(s') \quad (7)$$

Hereafter, in place of Equations 4 and 5, we define V_i^π and Q_i^π 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 $\pi_s(a_{-i} | a_i) = \frac{\pi_s(a_{-i}, a_i)}{\pi_s(a_i)}$ whenever $\pi_s(a_i) > 0$.

Remark 6 *Given a Markov game Γ_γ , a stationary policy π 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.:*

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} | a_i) Q_i^\pi(s, (a_{-i}, a_i)) < \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} | a_i) Q_i^\pi(s, (a_{-i}, a'_i)) \quad (8)$$

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 π to be a stationary correlated equilibrium policy in a Markov game.

Theorem 7 *Given a Markov game Γ_γ , a stationary policy π 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 $a'_i \in A_i(s)$,*

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} | a_i) Q_i^\pi(s, (a_{-i}, a_i)) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i} | a_i) Q_i^\pi(s, (a_{-i}, a'_i)) \quad (9)$$

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 π , 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 Γ_γ , a stationary policy π is a correlated equilibrium if for all $i \in N$, for all $s \in S$, and for all $a_i, a'_i \in A_i(s)$,*

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

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 π , 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 Γ_γ , 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 once again making explicit its dependence on the Q -values as well as the policy π . 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 $s \in S$,

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

Similarly, let us also express the Q -value function defined in Equation 7 once again 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 $a \in A(s)$,

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

Here, we have highlighted the fact that the dependence of Q on π arises through V . Finally, we define the correspondence $\pi_{\mathcal{Q}}^* : \mathcal{Q} \rightarrow \Pi$ 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 $a_i, a'_i \in A_i(s)$:

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

Theorem 9 *Given a Markov game Γ_γ , 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 π is a stationary correlated equilibrium policy.*

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

$$V_i(s) = \sum_{a \in A(s)} \pi_s(a) Q_i(s, a) \quad (14)$$

$$Q_i(s, a) = (1 - \gamma)R_i(s, a) + \gamma \sum_{s' \in S} P[s'|s, a]V_i(s') \quad (15)$$

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 π is a stationary correlated equilibrium policy. \blacksquare

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

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

$$(Q, \pi) \in \rho^*(Q, \pi) \tag{16}$$

$$= \{(Q_{\mathcal{V}}(V_{Q \times \Pi}(Q, \pi)), \pi') \mid \pi' \in \pi_{\mathcal{Q}}^*(Q_{\mathcal{V}}(V_{Q \times \Pi}(Q, \pi)))\} \tag{17}$$

$$= \{(Q_{\mathcal{V}}(V), \pi') \mid \pi' \in \pi_{\mathcal{Q}}^*(Q_{\mathcal{V}}(V))\} \tag{18}$$

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

Theorem 11 *The correspondence $\rho^* = (I \times \pi_{\mathcal{Q}}^*) \circ Q_{\mathcal{V}} \circ V_{Q \times \Pi}$ 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 $\rho^* = (I \times \pi_{\mathcal{Q}}^*) \circ Q_{\mathcal{V}} \circ V_{Q \times \Pi}$ 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). Although the existence of such policies is obvious, because Nash equilibria which are known to exist Fink (1964) are correlated equilibria, a general fixed-point computation falls out of our derivation (specifically, Equations 11, 12, and 13), based on which we propose a class of iterative algorithms in the next section. Then, in later sections, we investigate the question of whether or not certain variants of the algorithms in our class successfully 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 $t + 1$,

$$V^{t+1}(s) := \max_{a \in A(s)} Q^t(s, a) \tag{19}$$

$$Q^{t+1}(s, a) := (1 - \gamma)R(s, a) + \gamma \sum_{s' \in S} P[s'|s, a]V^{t+1}(s') \tag{20}$$

This procedure converges to a unique fixed point V^* , a unique fixed point Q^* , and a globally optimal policy π^* , 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 π^t ; and given a **selection mechanism** f , that is, a mapping from one-shot games into joint distributions; at time $t + 1$,

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

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

$$\pi_s^{t+1} \in f(N, A(s'), Q^{t+1}(s)) \quad (23)$$

We now proceed to investigate the question of whether or not an asynchronous variant of 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 “ Q -learning,” in which state values and Q -values at state-action pairs are updated asynchronously (see Tables 1), rather than “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 (see Table 1). 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 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 α decaying according to some schedule (step 7).

One important application-specific issue arises in multiagent Q -learning: 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? The primary difference between multiagent Q -learning as it applies in centralized and decentralized environments arises in step 3, the policy selection step. In a centralized setting, the central coordinator selects a joint distribution on which its updates in step 4 rely. In a decentralized setting, each individual agent selects a joint distribution based on which it performs the requisite updates in step 4. In either case, steps 3 and 4 require knowledge of *all* agents’ values and Q -values. By observing the agents’ actions and rewards in steps 1 and 2, sufficient information is assumed to be available to the individual agents in a decentralized setting, and to a central coordinator in a centralized setting, to carry out steps 3 and 4.

Next, we instantiate our generic multiagent Q -learning algorithm—parameterized by f —with specific equilibrium selection mechanisms, giving rise to (centralized and decentralized versions of) correlated- Q , Nash- Q , friend- Q , and foe- Q (i.e., minimax- Q) learning.

MULTIAGENTQ(Γ, f, g, α)	
Inputs	Markov game Γ , selection mechanism f , decay schedule g , learning rate α
Output	values V , Q -values Q , joint policy π^*
Initialize	values V , 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 $\pi_{s'}^* \in f(N, A(s'), Q(s'), V(s'))$
	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 α via g
FOREVER	

Table 1: Multiagent 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 an element in the set of joint distributions that satisfies Equation 1 (equivalently, Equation 2).

We consider five variants of correlated Q -learning. Each is defined by one of the following five objective functions, which we append to the system of linear inequalities (i.e., the linear program) expressed by Equation 2 to restrict the equilibrium selection process:

1. *utilitarian*: maximize the *sum* of all agents rewards:

$$u\text{CE}(N, A, R) = \operatorname{argmax}_{\pi \in \text{CE}(N, A, R)} \sum_{j \in N} \sum_{a \in A} \pi(a) R_j(a) \quad (24)$$

2. *egalitarian*: maximize the *minimum* of all agents rewards:

$$e\text{CE}(N, A, R) = \operatorname{argmax}_{\pi \in \text{CE}(N, A, R)} \min_{j \in N} \sum_{a \in A} \pi(a) R_j(a) \quad (25)$$

3. *plutocratic*: maximize the *maximum* of all agents rewards:

$$p\text{CE}(N, A, R) = \operatorname{argmax}_{\pi \in \text{CE}(N, A, R)} \max_{j \in N} \sum_{a \in A} \pi(a) R_j(a) \quad (26)$$

4. *dictatorial*: maximize agent i 's rewards:

$$d\text{CE}_i(N, A, R) = \operatorname{argmax}_{\pi \in \text{CE}(N, A, R)} \sum_{a \in A} \pi(a) R_i(a) \quad (27)$$

5. *closest match*: choose the correlated equilibrium in which all agents expected rewards are as close to their previous values (denoted here by the vector V) as possible:

$$cmCE(N, A, R, V) = \underset{\pi \in CE(N, A, R)}{\operatorname{argmin}} \max_{i \in N} \left| V_i - \sum_{a \in A} \pi(a) R_i(a) \right| \quad (28)$$

Note that only the closest match selection mechanism exploits the full generality of the policy update step in our multiagent Q -learning algorithm (step 3). In particular, it updates based on past *values*; all other techniques ignore such information. One could also imagine a dictatorial, and hence decentralized, version of closest match (which would also update based on past values). This variant of multiagent Q -learning is not studied here.

In the discussions that follow, we abbreviate these variants of correlated- Q (CE- Q) learning as uCE - Q , eCE - Q , pCE - Q , dCE - Q , and $cmCE$ - Q , respectively. The uCE - Q , eCE - Q , pCE - Q , and $cmCE$ - Q variants are naturally implemented as centralized algorithms, whereas the dictatorial variants are naturally implemented in a decentralized fashion, each agent choosing an equilibrium that maximizes its own utility.

Like the above variants of correlated Q -learning, variants of Nash- Q learning can also be seen as instances of generic multiagent Q -learning algorithms. Define $NE(N, A, R)$ to be the set of Nash equilibria in the one-shot game $\langle N, A, R \rangle$. By optimizing over this set instead of the set of correlated equilibria, we arrive at analogous algorithms based on analogous selection mechanisms, namely uNE - Q , eNE - Q , pNE - Q , dNE - Q , and $cmCE$ - Q .

Following Hu and Wellman (2003), we also consider coordinated Nash- Q (cNE - Q) and best Nash- Q (bNE - Q). In the former, a centralized algorithm, a central coordinator finds a Nash equilibrium using the Lemke-Howson algorithm (1964), which it broadcasts to all agents. In the latter, a decentralized algorithm, each agent independently selects a Nash equilibrium (again, using Lemke-Howson) 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 the dictatorial objective function, maximizing their own rewards, without enforcing the correlated equilibrium constraints expressed in Equation 2. Foe- Q learning works as follows in two-player games: player 1 optimizes the following objective function:

$$\operatorname{argmax}_{\sigma_1 \in \Delta(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} \sigma_1(a_1) R_1(a_1, a_2) \quad (29)$$

Player 2 optimizes analogously, and the joint distribution π 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 selection mechanisms).

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

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

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 $(\sigma_1^*, \dots, \sigma_n^*)$ is called a **Nash equilibrium** if σ_i^* is a **best-response** for player i to its opponents' mixed strategies, for all $i \in N$: i.e.,

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

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 $R_i : A \rightarrow \mathbb{R}$ 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:

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

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

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

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

We say that the **zero-sum property** holds of the Q -values of a Markov game Γ_γ 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 Γ , 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' ,

$$\pi_{s'}^{t+1} \in f(Q^t(s')) \quad (34)$$

$$V_1^{t+1}(s') := \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_1^t(s', a') \quad (35)$$

$$V_2^{t+1}(s') := \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_2^t(s', a') \quad (36)$$

$$Q_1^{t+1}(s, a) := (1 - \alpha) Q_1^t(s, a) + \alpha((1 - \gamma) R_1(s, a) + \gamma V_1^{t+1}(s')) \quad (37)$$

$$Q_2^{t+1}(s, a) := (1 - \alpha) Q_2^t(s, a) + \alpha((1 - \gamma) R_2(s, a) + \gamma V_2^{t+1}(s')) \quad (38)$$

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:

$$Q_1^{t+1}(s, a) := (1 - \alpha) Q_1^t(s, a) + \alpha((1 - \gamma) R_1(s, a) + \gamma V) \quad (39)$$

$$Q_2^{t+1}(s, a) := (1 - \alpha) Q_2^t(s, a) + \alpha((1 - \gamma) R_2(s, a) - \gamma V) \quad (40)$$

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 $t + 1$. \blacksquare

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 $i \in \{1, 2\}$,

$$Q_i^{t+1}(s, a) := (1 - \alpha) Q_i^t(s, a) + \alpha((1 - \gamma) R_i(s, a) + \gamma \text{CE}_i(Q^t(s'))) \quad (41)$$

$$:= (1 - \alpha) Q_i^t(s, a) + \alpha((1 - \gamma) R_i(s, a) + \gamma \text{NE}_i(Q^t(s'))) \quad (42)$$

$$= (1 - \alpha) Q_i^t(s, a) + \alpha((1 - \gamma) R_i(s, a) + \gamma \text{MM}_i(Q^t(s'))) \quad (43)$$

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. \blacksquare

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 Γ , 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 Γ_γ , 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 Γ , 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' ,

$$\pi_{s'}^{t+1} \in f(Q^t(s')) \tag{44}$$

$$V_i^{t+1}(s') := \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_i^t(s', a') \tag{45}$$

$$Q_i^{t+1}(s, a) := (1 - \alpha) Q_i^t(s, a) + \alpha((1 - \gamma) R_i(s, a) + \gamma V_i^{t+1}(s')) \tag{46}$$

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:

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

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 $t + 1$. \blacksquare

Given a one-shot game, an equilibrium π^* is called **Pareto-optimal** if there does not exist another equilibrium π 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 Γ , 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.*

R	b	s
B	2, 1	0, 0
S	0, 0	1, 2

Q_1^*	b	s
B	2, 1	$1, \frac{1}{2}$
S	$1, \frac{1}{2}$	$\frac{3}{2}, \frac{3}{2}$

Q_2^*	b	s
B	$\frac{3}{2}, \frac{3}{2}$	$\frac{1}{2}, 1$
S	$\frac{1}{2}, 1$	1, 2

Figure 1: Sample one-shot game: Bach vs. Stravinsky. Utilitarian correlated equilibrium Q -values for player 1 and player 2, respectively $\gamma = \frac{1}{2}$.

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 $i \in N$,

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

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 aforementioned multiagent Q -learning algorithms based on utilitarian, egalitarian, plutocratic, and dictatorial selection mechanisms, and best Nash, all select Pareto-optimal equilibria in common-interest Markov games. Thus, by Corollary 20, all these algorithms converge to Pareto-optimal equilibrium Q -values in this class of games, provided Q -values are initialized such that the common-interest property holds.

Common-interest Markov games can be viewed as glorified Markov decision processes. Indeed, in the single agent case, Equation 48 reduces to the classic Q -learning update rule.

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.² To guarantee that agents *play* equilibrium policies, not

2. 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,

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 can converge to either set of Q -values shown in Figure 1. Two agents playing this game, making independent decisions, can fail to coordinate their behavior, if, say, player 1 learns Q_1^* , and then selects and plays her part of the equilibrium (B, b) , while player 2 learns Q_2^* and then selects and plays his part of the equilibrium (S, s) , so that the action profile the agents play is (B, s) . Worse still, even if both agents learn Q_1^* , this behavior can arise because (B, b) and (S, s) are both utilitarian correlated equilibria in Q_1^* .

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 π_i according to that solution concept, but plays his marginal distribution, call it π_{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 in two-player, zero-sum games are “exchangeable” in the following sense (see, for example, Osborne and Rubinstein (1994)): If Γ is a two-player, zero-sum one-shot game with Nash equilibria (σ_1, σ_2) and (σ'_1, σ'_2) , then (σ_1, σ'_2) is also 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.

5. Experimental Overview

In the Sections 6 and 7, we describe experiments with multiagent Q -learning algorithms on a test bed of stylized Markov games, consisting of three grid games and soccer.³ Specifically, we compare the performance of (centralized) utilitarian, egalitarian, plutocratic, and (decentralized) dictatorial correlated Q -learning with other well-known multiagent Q -learning algorithms described in the literature, including: minimax (or foe) Q -learning, friend Q -learning, coordinated Nash Q -learning, best Nash Q -learning, and traditional Q -learning. We investigate the question of whether or not these Q -learning algorithms converge to equilibrium policies in these Markov games.

In an environment of multiple learners, off-policy traditional Q -learners would be 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, we investigated on-policy Q -learning (Sutton and

correlated value iteration converges to foe Q -values; in common-interest Markov games, Pareto-optimal correlated value iteration converges to friend Q -values.

3. A brief description of our experiments with these four games appeared in Greenwald and Hall (2003). Here, we present a far more thorough analysis of the behaviors of the various algorithms in these games. In particular, we explore empirical *non*convergence.

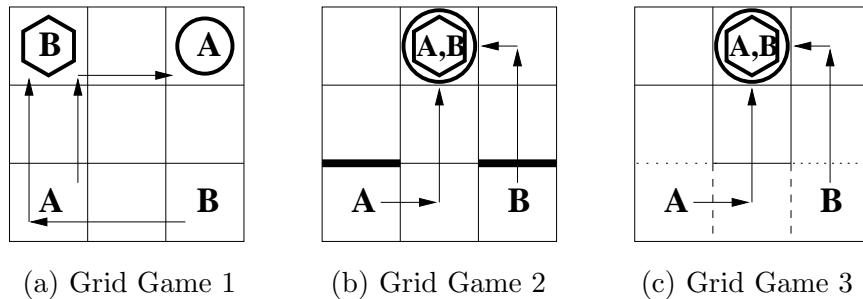


Figure 2: Grid games: Initial States and Sample Equilibria. Shapes indicate goals.

Barto, 1998). In our implementation, if ever the optimal action is not unique, the Q -learner randomizes uniformly among all its optimal actions. Otherwise, Q -learners can perform arbitrarily badly in games with multiple coordination equilibria, all of equivalent value, by repeatedly (and deterministically) failing to coordinate their behavior.

Almost without exception, Q -values tend to converge in these stylized Markov games. In Section 8, we extend our investigations, experimenting with centralized (e.g., utilitarian, egalitarian, and plutocratic) correlated and Nash Q -learning on a test set of 100 randomly generated Markov games. We conclude from these experiments that convergence to a stationary equilibrium policy of either correlated or Nash Q -learning with any of these selection functions is unlikely in general. In addition, we observe that the closest match selection operator, although it too does not always converge, outperforms the others. We also observe that correlated closest match outperforms Nash closest match by a substantial margin.

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 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 game with Chicken-like rewards.⁴ Indeed, only GG2 is stochastic. In the next section, we describe experiments with a simple version of soccer, a two-player, zero-sum Markov game, that is inherently 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 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,

4. Chicken is a game played by two people driving cars in front of an audience they wish to impress. Each driver can either drive straight ahead, and risk his life, or swerve out of the way, and risk embarrassment.

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.

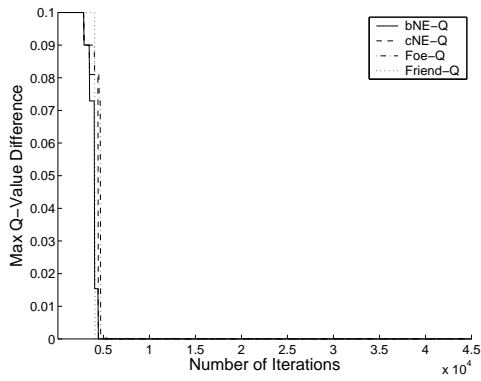
Since there are multiple pure strategy stationary equilibrium policies in these grid games, it is possible to construct additional stationary equilibrium policies as convex combinations of the pure policies. In GG2, there exists a continuum of symmetric correlated equilibrium policies: i.e., for all $p \in [0, 1]$, 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

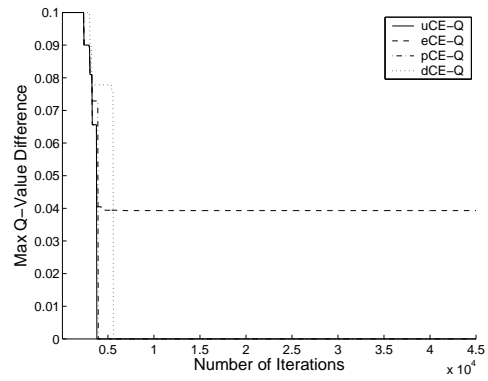
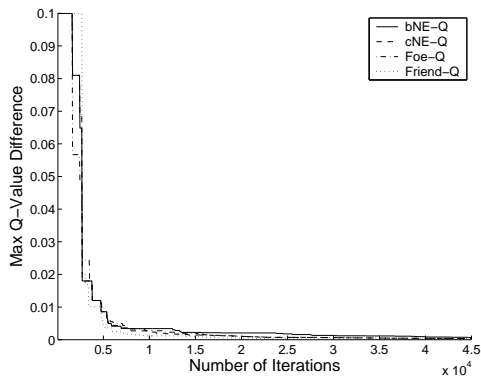
Our experiments reveal that nearly all of the Q -learning algorithms in our test suite are converging in the three grid games. (Two exceptions are noted below.) Littman (2001) proves that FF- Q converges in all general-sum Markov games. Hu and Wellman (2003) show empirically that coordinated and best Nash- Q both converge in both GG1 and GG2. Figure 3 shows that these variants of NE- Q are also converging in GG3. Figure 3 also shows that three variants of CE- Q are converging in all three grid games; however, e CE- Q does not converge at four (of 57) states in GG1. Finally, on-policy Q -learning is converging in GG1 and GG2, but not necessarily in GG3 (see Figure 5).

The values plotted in Figures 3, 4 and 5 are computed as follows. Define an error term 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., $ERR_i^t = |Q_i^t(s^t, a^t) - Q_i^{t-1}(s^t, a^t)|$. 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., $\max_{t=x, \dots, T} ERR_i^t$, for $i = 1$. The values on the x -axis, representing time, range from 1 to T' , for some $T' < T$.⁵

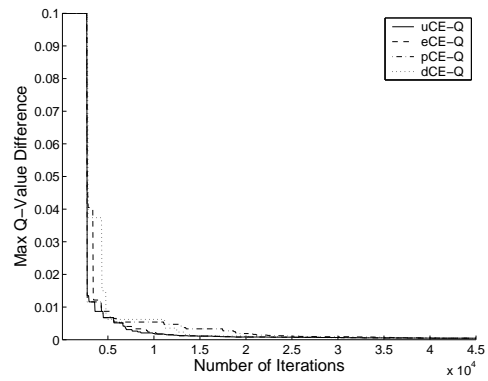
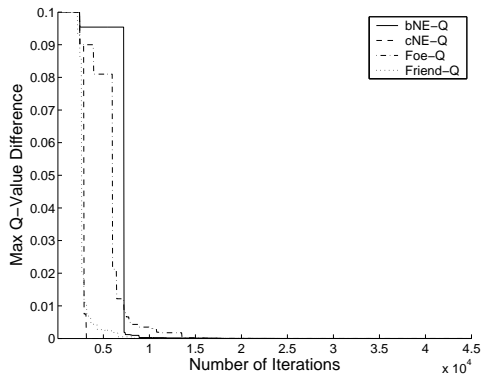
5. Setting $T' = T$ is sometimes misleading: It could appear that non-convergent algorithms are converging, because our metric measures the maximum error between the current time and the end of the simulation, but it could be that the change in Q -values is negligible for all states visited at the end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



(a) Grid Game 1

(b) Grid Game 1: CE- Q 

(c) Grid Game 2

(d) Grid Game 2: CE- Q 

(e) Grid Game 3

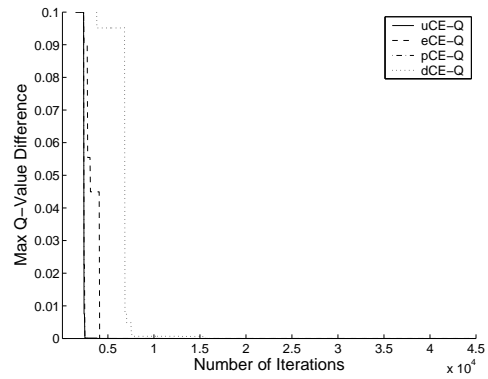
(f) Grid Game 3: CE- Q

Figure 3: Changing Q -values for multiagent Q -learning in the grid games. These graphs depict empirical convergence of all algorithms, but $eCE-Q$, after 5×10^4 iterations.

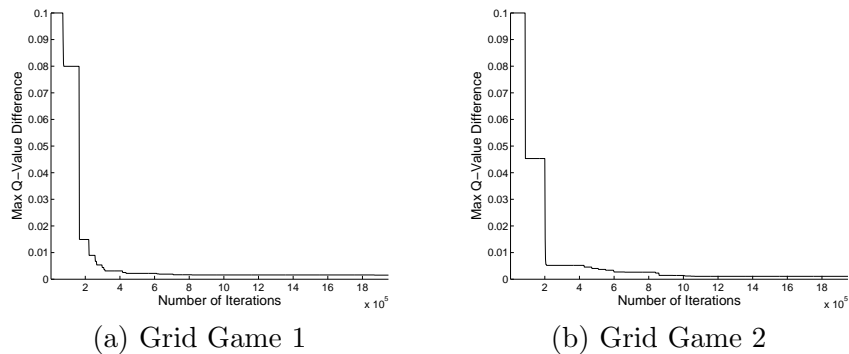


Figure 4: Changing Q -values for traditional Q -learning in GG1 and GG2 with $\epsilon = 0.01$. Traditional Q -learning is converging empirically in these games.

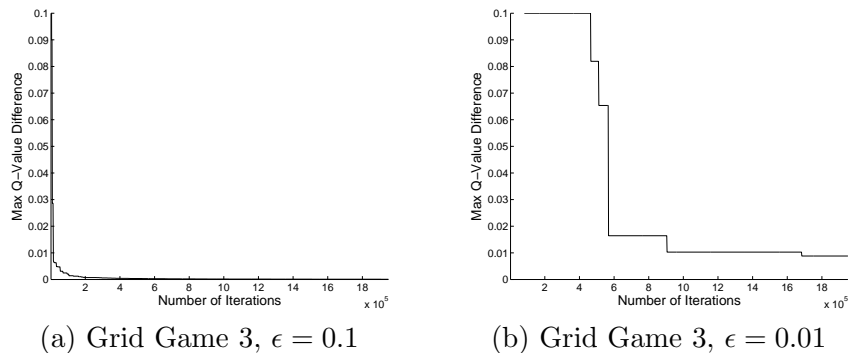


Figure 5: Changing Q -values for traditional Q -learning in GG3. Traditional Q -learning is converging empirically when $\epsilon = 0.1$, but not when $\epsilon = 0.01$.

In our experiments, we set $T' = 4.5 \times 10^4$ and $T = 5 \times 10^4$ for all algorithms except traditional Q -learning, for which $T' = 1.95 \times 10^6$ and $T = 2 \times 10^6$.

In our experiments, the algorithmic parameters are set as follows. Our implementation of traditional Q -learning is on-policy and ϵ -greedy, with $\alpha = 1/n(s, a)$, where $n(s, a)$ is the number of visits to state-action pair (s, a) , and $\epsilon = 0.01$, typically. (In GG3, for reasons that will become apparent, we also set $\epsilon = 0.1$.) All other algorithms are off-policy (equivalently, on-policy and ϵ -greedy with $\epsilon = 1$). For these off-policy learning algorithms, in GG1 and GG3, where there is no stochasticity, $\alpha = 1$; in GG2, however, as in traditional Q -learning, $\alpha = 1/n(s, a)$. Finally, $\gamma = 0.9$ in all cases.

Observe that the maximum change in Q -values is converging to zero for all algorithms in all games, except $eCE-Q$ in GG1 and traditional Q -learning with $\epsilon = 0.01$ in GG3. It may appear as if traditional Q -learning with $\epsilon = 0.01$ is slowly converging in GG3; however, the maximum difference in Q -values is gradually decreasing only because of the decaying

α parameter.⁶ In contrast, when $\epsilon = 0.1$, traditional Q -learning readily converges. Clearly, this approach is sensitive to the implementer’s choice of exploration strategy. Next, we detail the perhaps surprising behavior of eCE - Q in GG1.

6.2.1 eCE - Q LEARNING

In GG1, while the Q -values are converging at all state-action pairs for uCE - Q , pCE - Q , and dCE - Q , this is not the case for eCE - Q . The Q -values for this algorithm are converging at all states except the four states (of 57) depicted in Figure 6. Here, we investigate the root cause of non-convergence, and find that the leftmost state in Figure 6, where each player is in the other player’s goal, is the culprit. The Q -values do not converge at the other three states only because the agents can transition from these states to the problematic state.

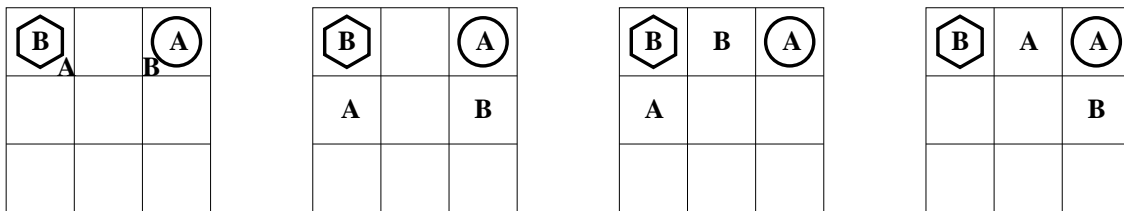


Figure 6: Non-convergent states for eCE - Q in GG1. The leftmost state, where each player is in the other player’s goal, is the culprit. All other states, Q -values do not converge only because the agents can transition from these states to the problematic state.

The Q -values at the problematic state, at an arbitrary point in time, are depicted below:

	LEFT	DOWN
RIGHT	0.13,0.13	9,0
DOWN	0,9	7.29,7.29

The unique equilibrium of the one-shot game induced by these Q -values is (RIGHT, LEFT). But whenever the players play these actions, they collide—earning -1 —and return to this state. Thus, by playing these actions the associated Q -values decrease, until eventually they become negative. At that point, there are two pure strategy equilibria, (RIGHT, DOWN) and (DOWN, LEFT), and a continuum of correlated equilibria, as in the Bach or Stravinsky game.

When the Q -values of (RIGHT, LEFT) are negative, the egalitarian correlated equilibrium operator chooses to play each of the pure strategy equilibria with probability $1/2$. At this point, the value at this state becomes $(\frac{9}{2}, \frac{9}{2})$. But now, upon updating the Q -values of (RIGHT, LEFT), the entries in this cell become positive, so that (RIGHT, LEFT) is again the unique pure strategy equilibrium at this state, and the cycle repeats.

In contrast, when the Q -values of (RIGHT, LEFT) are negative, all the other correlated equilibrium selection operators (i.e., utilitarian, plutocratic, and dictatorial) deterministi-

6. In Section 7, in our experiments with soccer, we again observe nonconvergent behavior for traditional Q -learning. There, we provide a detailed analysis of a state at which the Q -values do not converge.

cally choose to play one of the pure strategy equilibria, say (RIGHT, DOWN).⁷ In this case, upon updating the Q -values of (RIGHT, LEFT), the second entry remains negative, never to be positive again (see the Q -values depicted below). RIGHT becomes a dominant strategy for player 1, and the Q -values converge.

	LEFT	DOWN
RIGHT	8,-0.1	9,0
DOWN	0,9	7.29,7.29

A snapshot of e CE- Q 's oscillating Q -values in GG1 is depicted in Figure 7. These cycles were generated by an implementation of value iteration, rather than Q -learning. Although non-convergence is apparent with Q -learning, regular cycles are not, because of the asynchronous nature of the updating. In fact, the policies corresponding to these cyclical Q -values comprise a “cyclic” equilibrium in GG1, as defined in Zinkevich et al. (2006).

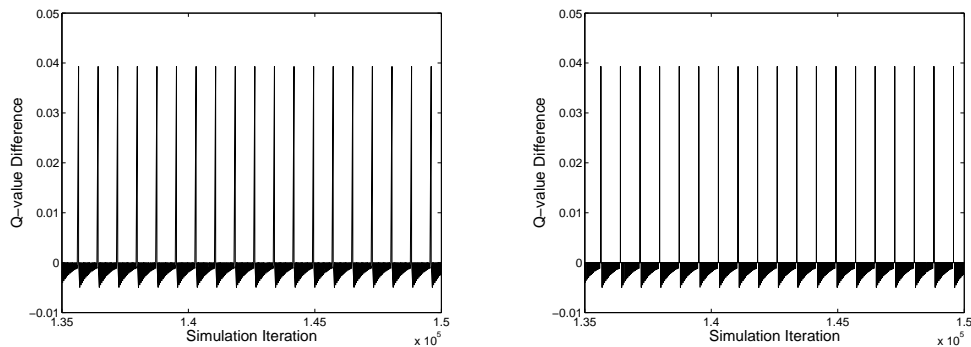


Figure 7: Non-convergence of value iteration at the left- and rightmost states in Figure 6.

6.3 Equilibrium Policies

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

- traditional Q -learning learns equilibrium policies when it converges, but does not always converge;
- 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 (although Q -values do not converge at all states for CE- Q).

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

7. A deterministic choice is the only choice for the plutocratic and the dictatorial operators. The utilitarian operator, however, is indifferent between selecting one of the pure strategy equilibria and the egalitarian equilibrium: Its deterministic choice is an artifact of our implementation.

games repeatedly according to the policies they learned. Our learning phase is randomized: not only are the state transitions stochastic, on-policy traditional 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 2.

Q -Learning In GG1 and GG2, we find that traditional Q -learners with $\epsilon = 0.01$ not only converge, they learn to play equilibrium policies. In GG3, such learners neither converge nor play equilibrium policies when $\epsilon = 0.01$; however, when $\epsilon = 0.1$, so that traditional Q -learners converge in GG3, they, too, learn to play equilibrium policies.

It is conceivable that under certain (perhaps restrictive) conditions, on-policy traditional Q -learning can be guaranteed to converge to an equilibrium policy. However, it appears to be quite brittle, and not sufficiently robust for general multi-agent learning environments. Indeed, this supposition was one of the underlying motivations for this study.

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. Furthermore, by choosing these policies, occasionally both agents fail to bypass the barrier (perhaps repeatedly); thus, foe Q -learning completes fewer games than are completed in equilibrium play.

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 policy⁸ 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.

8. 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.

GG1	Avg. Score	Games	Convergence?	Eqm. Values?	Eqm. Play?
Q	100,100	2500	Yes	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	51.43,100	3333	Yes	Yes	Yes
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?
$Q(\epsilon = 0.01)$	120,120	3333	No	No	No
$Q(\epsilon = 0.1)$	100,125	3333	Yes	Yes	Yes
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 2: 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.

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

NE- Q and CE- Q 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 on the equilibrium path of play:⁹ 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 GG1. 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 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:¹⁰ 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

9. Although e CE- Q 's values do not converge at all states, the non-convergent states are not reachable via the equilibrium policies learned at earlier states. Thus, these non-convergent states were irrelevant during the testing phase.

10. 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 exists some other strategy which yields higher rewards than said strategy in all circumstances.

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 $uCE-Q$ and $eCE-Q$ 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 $cNE-Q$, the $pCE-Q$ learning algorithm converges to a pure strategy equilibrium policy that is among those which maximize the maximum of all agents’ rewards. Finally, each $dCE-Q$ and $bNE-Q$ 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, $uCE-Q$ and $eCE-Q$ earn higher rewards than $cNE-Q$, and $pCE-Q$ 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 classic Q -learning. While it is generally understood that the dynamics 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. We find that even in this relatively simple game, for which theoretical guarantees about multiagent Q -learning can be obtained (see, Section 4), multiple Q -learners do not converge after millions of iterations. We conclude that traditional Q -learning is not of practical use in Markov games.

The soccer field is a grid (see Figure 8). 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.¹¹ 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 8 (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.

11. This form of the game is due to Littman (1994).

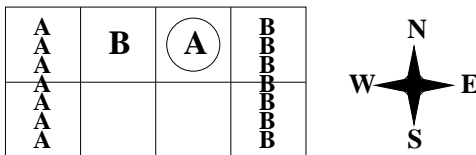


Figure 8: Soccer Game. The circle represents the ball. If player A moves W , he loses the ball to player B ; but if player B moves E , attempting to steal the ball, he cannot.

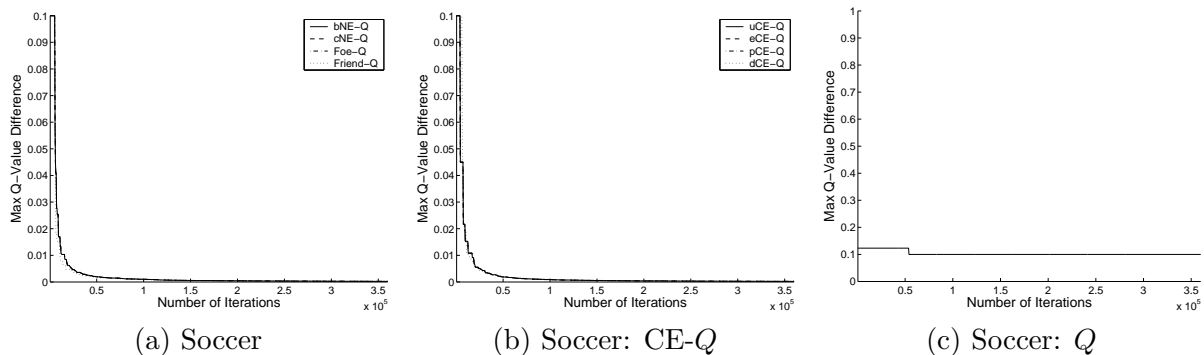


Figure 9: Changing Q -values in soccer: all algorithms except traditional Q -learning are converging. The discount factor $\gamma = 0.9$ and the parameter $\alpha = 1/n(s, a)$, where $n(s, a)$ is the number of visits to state-action pair (s, a) . Our traditional Q -learning implementation is on-policy and ϵ -greedy with $\epsilon = 0.01$.

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 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 9 shows that while all the multiagent- Q learning algorithms implemented converge, *traditional Q -learning does not converge*. Our implementation of basic Q -learning is on-policy and ϵ -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., $\max_{t=x, \dots, T} \text{ERR}_i^t = \max_{t=x, \dots, T} |Q_i^t(s^t, a^t) - Q_i^{t-1}(s^t, a^t)|$, for $i = A$. The values on the x -axis, representing time, range from 1 to T' , for some $T' < T$. We set $T = 3.6 \times 10^5$ and $T = 4 \times 10^5$ for all algorithms.

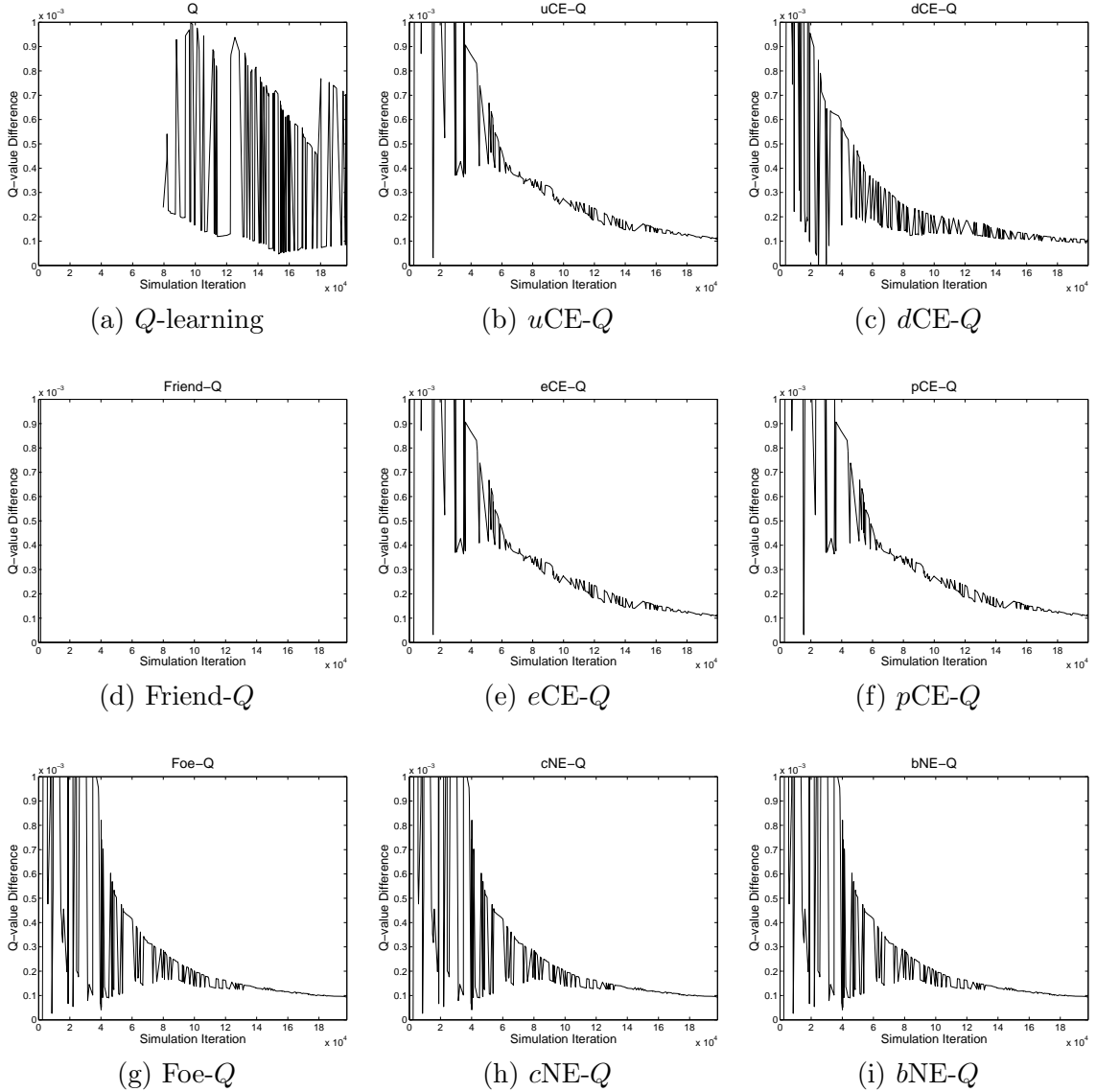


Figure 10: Changing Q -values at select state \hat{s} . All algorithms are converging, except basic Q -learning. Here, we depict only ten thousand iterations, but classic Q -learning failed to converge after six million iterations.

Figure 10 presents an example of a state-action pair at which traditional Q -learning does not converge. The values on the x -axis represent time, and the corresponding y -values are the error values $\text{ERR}_A^t = |Q_i^t(\hat{s}, S, E) - Q_i^{t-1}(\hat{s}, S, E)|$. In Figure 10(a), although the Q -value differences are decreasing at times, they are not converging. They are decreasing only

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 3: Testing phase: 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 2.

because the learning rate α 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 10(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 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} ,¹² although in principle, correlated- Q can learn correlated equilibrium policies, even in two-player, zero-sum Markov games.

7.2 Equilibrium Policies

In Table 3, we present the results of a testing phase for this soccer game. All players, except for traditional 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 soccer. All CE- Q and NE- Q variants behave in a manner that is similar to one another and similar to foe- Q .¹³

12. We did not check for independent policies at all states.

13. Any differences in scores among these algorithms is due to randomness in the simulations.

In summary, in soccer, a two-player, zero-sum Markov game, traditional Q -learning does not converge. Intuitively, the rationale for this outcome is clear: classic Q -learners seek deterministic optimal policies, but in this game no such policies exist.¹⁴ 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

In the last few sections, we showed how multiagent Q -learning endowed with centralized equilibrium selection mechanisms can outperform more exotic techniques, such as friend- Q , foe- Q , traditional Q -learning, and multiagent Q -learning with decentralized equilibrium selection mechanisms. These toy examples might lead one to falsely conclude that these techniques are adequate for multiagent learning in arbitrary Markov games. This section suggests they do not always converge to an equilibrium.¹⁵

In this section, we report the performance of centralized correlated and centralized Nash Q -learning algorithms applied to a dataset of randomly generated games. The results are largely disappointing: even after thousands of iterations, the utilitarian, egalitarian, plutocratic, and dictatorial selection mechanisms leave a substantial fraction of the games unsolved. For example, after many iterations on random games with five states, none of the corresponding CE algorithms finds a stationary CE on more than 83 of 100 games and none of these NE algorithms finds a stationary NE on more than 69 of 100 games.

The primary reason for this outcome is that most selection techniques do not put any emphasis on stability. In particular, small changes in Q -values can drastically affect state values. Hence, we propose the closest match equilibrium selection mechanism (see Section 3 for the formal definition). During each iteration of multiagent Q -learning, closest match chooses an equilibrium whose values are closest to the values the state was estimated as having during the last iteration. We find that $cmCE$ - Q and $cmNE$ - Q are more likely to converge to equilibria than the other selection mechanisms tested.

In Section 8.1, we outline the manner in which we randomly generate test games. In Section 8.2, we present our methodology for evaluating performance in these experiments, which differs from that of the grid game experiments because the equilibria of these games are not known *a priori*. Finally, in Section 8.3, we summarize our findings.

8.1 Setup

Following Zinkevich et al. (2006), our game generator takes as input a set of players N , a set of states S , and for each player $i \in N$ and state $s \in S$, a set of actions $A_i(s)$. To construct a game, for each state-action pair $(s, a) \in \mathcal{A}$, the value of $P(s, a)$ is set to be a distribution with all its mass on one state, chosen uniformly at random; and, for each agent

14. Recall that in our implementation of traditional 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, classic Q -learning cannot converge.

15. We cannot declare definitively from an experiment that an algorithm does not converge on some game, because it could be the case that continuing to run the algorithm would lead to convergence.

$i \in N$, the value of $R_i(s, a)$ is set to be an integer between 0 and 99, also chosen uniformly at random. The discount factor was fixed at $\gamma = 0.75$ in all games.

Zinkevich et al. (2006) demonstrate that multiagent value iteration techniques fail to converge to stationary equilibria in general-sum Markov games. In this work, we investigate the convergence properties of multiagent Q -learning algorithms in similar games.

8.2 Methodology

As in our experiments with the grid games, in these experiments with randomly generated games, we test for convergence to stationary equilibrium policies. But unlike in the grid games, the equilibria of these games are not known in advance. (Indeed, multiagent Q -learning can be viewed as a means of computing equilibrium policies in Markov games!)

To test for convergence in this setting, we test whether the final policy π^* is an ϵ -stationary equilibrium policy. In particular, for small $\epsilon > 0$, we check this condition:

$$\epsilon + \sum_{a_{-i} \in A_{-i}(s)} \pi_s^*(a_{-i}, a_i) Q_i^{\pi^*}(s, (a_{-i}, a_i)) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s^*(a_{-i}, a_i) Q_i^{\pi^*}(s, (a_{-i}, a'_i)) \quad (49)$$

In this way, we compute a score for each algorithm, namely “in how many games does the algorithm find a policy near a stationary equilibrium?”

To compute these scores requires that we first compute Q^{π^*} . This can be done using a straightforward generalization of policy evaluation from Markov decision processes to Markov games (see Table 4). As in the single-agent case, the update function is a contraction mapping, and hence policy evaluation in Markov games converges to the true value and Q -value functions. In our experiments, we ran policy evaluation for 100 iterations, which yields an approximation of Q^{π^*} that is within 10^{-9} of the actual Q^{π^*} function.

MULTIAGENTPE(Γ, π)	
Inputs	Markov game Γ , policy π
Output	values V , Q -values Q
Initialize	values V
REPEAT	
1.	For all agents i , states s , actions a
(a)	$Q_i(s, a) = (1 - \gamma)R_i(s, a) + \gamma \sum_{s' \in S} P[s' s, a]V_i(s')$
2.	For all agents i , states s , actions a
(a)	$V_i(s) = \sum_{a \in A(s)} \pi_s(a)Q_i(s, a)$
FOREVER	

Table 4: Multiagent policy evaluation.

8.3 Results

For testing purposes, we generated five sets of 100 games, each set having a fixed number of players (2) and actions (3) and between two and six states. We ran the centralized variants of correlated and Nash Q -learning. Learning was off-policy, and every 100 iterations the system was restarted: i.e., the state and the action profile were reinitialized. The total number of iterations was 1000 times the number of states times the number of action

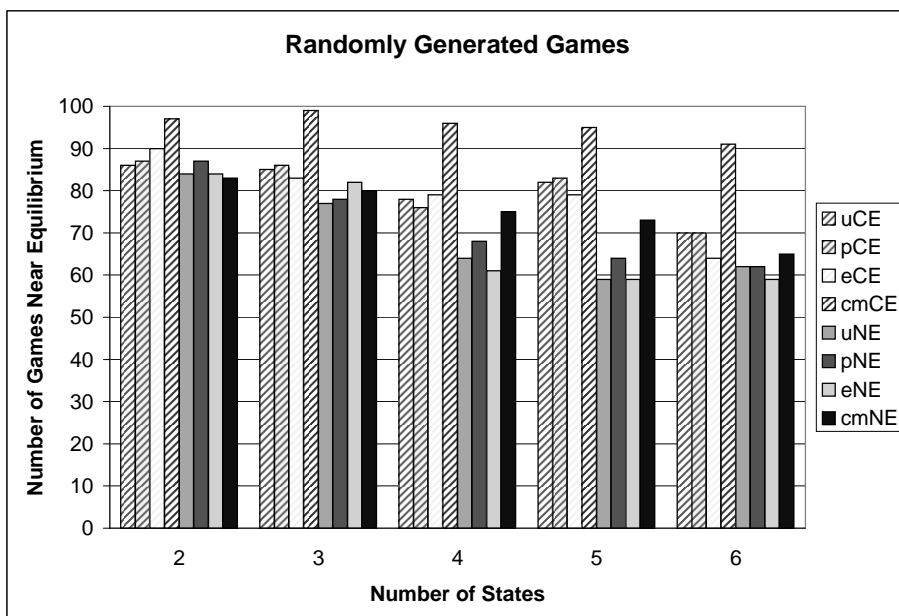


Figure 11: A comparative analysis of various centralized multiagent Q -learning algorithms. Note that $cmCE$ - Q learning outperforms all others, doing so by a wider and wider margin as the number of states increases. In contrast, $cmNE$ - Q outperforms all other Nash equilibrium selection mechanisms, but tends not to perform as well as the correlated equilibrium selection mechanisms.

profiles, so that on average each state-action pair was visited 1000 times. Since the games have deterministic transitions, we set $\alpha = 1$. We report experiments with $\epsilon = 0.0001$, although we would not expect to see qualitatively different results for similar values of ϵ (e.g., $\epsilon = 0.001$).

Our results are presented in Figure 11. In summary,

1. none of the proposed correlated or Nash selection mechanisms give rise to multiagent Q -learning algorithms that converge in all games;
2. $cmCE$ - Q outperforms the other CE - Q learning algorithms, and the degree of this outperformance grows as the game size (i.e., number of states) increases;
3. $cmNE$ - Q outperforms the other NE - Q learning algorithms for four, five, and six state games, but not for two or three state games (i.e., no clear trend emerges); and
4. CE - Q learning tends to outperform NE - Q learning (one insignificant exception is $cmNE$ - Q learning vs. eCE - Q learning in six state games).

In conclusion, while the closest match selection mechanisms dominate the others in our test suite, no known multiagent Q -learning algorithm suffices to learn stationary equilibrium policies in arbitrary general-sum Markov games.

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 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 reinforcement 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, two-player, zero-sum and common-interest. Empirically, we established that neither correlated nor Nash Q -learning converge in general. Still, 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. In fact, we contend that correlated- Q is preferred to Nash- Q for the following 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.
3. On a test set consisting of randomly generated games, correlated Q -learning converged more often than Nash- Q learning. (It remains to verify this claim on larger test sets.)

In related work, we have also 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 multi-agent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, June 1998.
- A.M. Fink. Equilibrium in a stochastic n -person game. *Journal of Science in Hiroshima University*, 28:89–93, 1964.
- 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.
- C.E. Lemke and J.T. Howson Jr. Equilibrium points of bimatrix games. *SIAM Journal of Applied Mathematics*, 12:413–423, 1964.
- 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.
- M. Zinkevich, A. Greenwald, and M. Littman. Cyclic equilibria in Markov games. In *Advances in Neural Information Processing Systems 18*, page To Appear. MIT Press, 2006.