Correlated Q-Learning

Amy Greenwald, Keith Hall and Martin Zinkevich

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-05-08
July 2005

Brown University Technical Report (2005) CS-05-08

Correlated Q-Learning

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

Keith Hall KEITH_HALL@JHU.EDU
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21201

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

Abstract

Recently, there have been several attempts to design multiagent learning algorithms that learn equilibrium policies in general-sum Markov games, just as Q-learning learns optimal policies in Markov decision processes. This paper introduces *correlated-Q* learning, one such algorithm. The contributions of this paper are twofold: (i) We show empirically that correlated-Q learns correlated equilibrium policies on a standard test bed of Markov games. (ii) We prove that certain variants of correlated-Q learning are guaranteed to converge to stationary correlated equilibrium policies in two special classes of Markov games, namely zero-sum and common-interest.

Keywords: Multiagent Learning, Reinforcement Learning, Markov Games

1. Introduction

Recently, there have been several attempts to design multiagent learning algorithms that learn equilibrium policies in general-sum Markov games, just as Q-learning learns optimal policies in Markov decision processes. Hu and Wellman (2003) propose an algorithm called Nash-Q that converges to Nash equilibrium policies in general-sum games under restrictive conditions. Littman's (2001) friend-or-foe-Q (FF-Q) algorithm always converges, but only learns equilibrium policies in restricted classes of games. For example, Littman's (1994) minimax-Q algorithm (equivalently, foe-Q) converges to minimax equilibrium policies in two-player, zero-sum games. This paper introduces correlated-Q (CE-Q) learning, a multiagent Q-learning algorithm based on the correlated equilibrium solution concept (Aumann, 1974). CE-Q generalizes Nash-Q in general-sum games, since the set of correlated equilibria contains the set of Nash equilibria; CE-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), CE can be computed efficiently via linear programming. Also unlike NE, to which no general class of learning algorithms is known to converge, no-internal-regret algorithms (e.g., Foster and Vohra (1997)) converge to the set of CE in repeated games. In addition, CE that are not NE can achieve higher rewards than NE, by avoiding positive probability mass on less desirable outcomes (e.g., 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 equilibrium selection problem by introducing four variants of CE-Q, based on four equilibrium selection mechanisms. We define utilitarian, egalitarian, plutocratic, and dictatorial CE-Q learning, and we demonstrate empirical convergence to correlated equilibrium policies for all four CE-Q variants on a standard test bed of Markov games.

Overview This paper is organized as follows. First, we review the definition of correlated equilibrium in one-shot games, and we define correlated equilibrium policies in Markov games. In Section 3, we define two versions of multiagent Q-learning, one centralized and one decentralized, and we show how CE-Q, Nash-Q, and FF-Q all arise as special cases of these generic algorithms. In Section 4, we compare utilitarian, egalitarian, plutocratic, and dictatorial CE-Q learning with Q-learning, FF-Q, and Nash-Q in grid games. Next, we describe experiments with the same set of algorithms in a simple soccer game. Overall, we demonstrate that CE-Q learns correlated equilibrium policies on this standard test bed of general-sum Markov games. Finally, we include a theoretical discussion of zero-sum and common-interest Markov games, in which we prove that certain variants of CE-Q learning are guaranteed to converge to stationary correlated equilibrium policies.

2. Correlated Equilibrium Policies in Markov Games

In this section, we review the definition of correlated equilibrium in one-shot games, and we define correlated equilibrium policies in Markov games. In a companion paper (Greenwald and Zinkevich, 2005), we provide a direct proof of the existence of correlated equilibrium policies in Markov games.

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

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

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

Let us imagine that in addition to the players, there is also a *referee*,[1] who can be considered to be a physical machine (i.e., the referee itself has no beliefs, desires, or intentions). At each time step, the referee sends to each player a private signal consisting of a recommended action for that player.[2] We assume the referee selects these actions according to a *stationary* policy $\pi \in \prod_{s\in S} \Delta(A(s))$: i.e., a policy that depends only on state, not on time.

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

1. Note that the referee is not part of the definition of a Markov game. While a referee can be of assistance in the implementation of a correlated equilibrium, the concept can be defined without reference to this third party. In this section, we introduce the referee as a pedagogical device. In our experimental work, we sometimes rely on the referee to facilitate the implementation of correlated equilibria.
2. Generalizing sunspot equilibria (Shell, 1989), which rely on public randomization devices, to define or implement a correlated equilibria, the referee sends a *private*, rather than a public, signal to each player. It suffices for the referee to send to each player as this private signal precisely its recommended action. Any joint distribution of the players' actions that could arise by the referee sending more general signals can also be achieved by the referee sending each player its recommended action (Aumann, 1974).

2.1 Correlated Equilibrium in One-Shot Games: A Review

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

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

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

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

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

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

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

Equation 2 is Equation 1 multiplied by $\pi(a_i)$. Equation 2 holds trivially whenever $\pi(a_i) = 0$, because in such cases both sides equal zero. Given a one-shot game 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 $a_{-i} \in A_{-i}$, $\pi(a_{-i} \mid a_i) = \pi(a_{-i} \mid a'_i)$, whenever $\pi(a_i), \pi(a'_i) > 0$), then a correlated equilibrium is also a Nash equilibrium. In fact, any Nash equilibrium can be represented as a correlated equilibrium: the players can simply generate their own advice (independently). Existence of both types of equilibria is ensured by Nash's theorem (Nash, 1951). Therefore, the set of correlated equilibria is nonempty, as well. We have established the following (well-known) result.

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

Finally, we note that 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.

Game 1: Bach or Stravinsky

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

2.1.1 Examples

Here, we show by example some of the benefits of correlated equilibria over Nash equilibria. Game 1 represents a situation where two agents, the first a Bach lover and the second a Stravinsky lover, are deciding whether to go to a Bach concert or a Stravinsky concert. The first agent selects a row, and the second agent selects a column. The utility profile of each outcome is written in each cell. The first agent prefers to go to the Bach concert, but given a choice between going with the second agent to the Stravinsky concert or going to the Bach concert alone, she prefers the former. Similarly, the second agent prefers to go to the Stravinsky concert, but given a choice between going with the first agent to the Bach concert or going to the Stravinsky concert alone, he too prefers the former.

In Game 1, there are two pure strategy Nash equilibria: one where the agents play (b, B) (go to a Bach concert together) and another where the agents play (s, S) (go to a Stravinsky concert together). Either of these behaviors could be considered "unfair:" the outcome (s, S) is unfair to the first agent, since she would rather go to the Bach concert; similarly, the outcome (b, B) is unfair to the second agent, since he would rather go to the Stravinsky concert. There is one mixed strategy Nash equilibrium in this game, where the first agent plays b with probability $2/3$ and the second agent plays s with probability $2/3$. Here, each agent obtains a utility of $2/3$, which is "fair," but is also less than either agent would obtain were either of the pure strategy Nash equilibria to be played.

If two people were seeking a fair solution to this game, they might decide to flip a (fair) coin, agreeing in advance that if the coin comes up heads, they both go to the Bach concert, whereas if the coin comes up tails, they both go to the Stravinsky concert. This solution is an example of a correlated equilibrium, where $\pi(b, B) = 1/2$ and $\pi(s, S) = 1/2$. Not only is this solution fair, it is also Pareto-optimal, that is, no agent can be made better off without making some other agent worse off.[3]

3. In Markov games, such a balance can be achieved another way. For instance, in a Markov game with Game 1 as the only state (i.e., the infinitely repeated game of Bach or Stravinsky), the agents can alternate between going to the Bach concert and going to the Stravinsky concert. However, it is not always the case that time can be used as a correlation device, as the following example shows.

   Imagine three agents choosing a number between 1 and 100. If the first two agents choose the same number, but the third agent chooses a different number, then the third agent must pay each of the first two agents one dollar. Otherwise, each of the first two agents pay the third agent one dollar. A game in this spirit is played by a quarterback (the first agent), a wide receiver (the second agent), and an opposing cornerback (the third agent): the quarterback and the wide receiver must agree upon a passing pattern which the opposing cornerback must guess.

   Now, observe that it is a Nash equilibrium for each agent to choose a number uniformly at random between 1 and some $m \in \{1, \ldots, 100\}$. Among these equilibria, the one that is best for the first two agents is $m = 2$, in which case they win a quarter of the time. However, if they correlate their actions, both of them choosing the same number k uniformly at random between 1 to 100, they win 99/100 of the time. In this game, the agents cannot use time as a correlation device because the third agent could anticipate their choices.

Game 2: Shapley's Game

	R	P	S
r	0,0	0,1	1,0
p	1,0	0,0	0,1
s	0,1	1,0	0,0

In Shapley's Game (Game 2), both agents earn a higher utility by playing a correlated equilibrium instead of a Nash equilibrium. (Shapley's game differs from Rock-Paper-Scissors only in that in the latter the diagonal entries are $(\frac{1}{2}, \frac{1}{2})$.) At the unique Nash equilibrium, each agent chooses an action uniformly at random and each agent's expected utility is 1/3. However, if a referee selects an action profile uniformly at random from the set

$$\{(r, P), (r, S), (p, R), (p, S), (s, R), (s, P)\}$$

and if the two agents follow the referee's advice, then each agent's expected utility is 1/2. Initially, one might think that the referee could select uniformly at random from, say $\{(r, P), (p, R)\}$. But then, if the first agent were advised to play r, she could infer that the second agent was advised to play P, which would motivate her to play s. If one agent were to cooperate with the referee, the other agent would be motivated to deviate.

By playing a correlated equilibrium in the Bach or Stravinsky game, the agents achieve a fair and Pareto-optimal solution. In Shapley's game, by playing a correlated rather than a Nash equilibrium, all of the agents fare better. In addition, a correlated equilibrium in a one-shot game can be computed in polynomial time. The corresponding complexity question for Nash equilibrium is open, although it is known that it is NP-hard to compute certain classes of Nash equilibria (Gilboa and Zemel, 1989; Conitzer and Sandholm, 2003).

2.2 Correlated Equilibrium Policies in Markov Games

We are now ready to address the question of whether or not agents are willing to follow the advice of a referee in a Markov game. To do so, 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 $T^\pi_{ss'}$ is the probability of transitioning to state s' from state s, given that the referee selects an action profile according to the distribution π_s that the agents indeed follow:

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

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

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

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

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

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, is an analog of Bellman's Theorem (Bellman, 1957) that follows directly from Equations 4 and 5 via the Markov property. (Note also that the referee's policy π is stationary by assumption.)

Theorem 4 *Given a Markov game Γ_γ, for any $V : S \to [\alpha,\beta]^n$, for any $Q : \mathcal{A} \to [\alpha,\beta]^n$, and for any stationary policy π, $V = V^\pi$ and $Q = Q^\pi$ if and only if:*

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

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

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.

Definition 5 *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.*

In this section, by assuming that the referee's policy is stationary, we restrict our attention to stationary correlated equilibrium policies.

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

Remark 6 *Given a Markov game 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.*:*

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

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

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

Here, in state s, when it is recommended that agent i play a_i, it is happy 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 policy π, then from the point of view of agent i, its environment is an MDP. Hence, the one-shot deviation principle for MDPs establishes Theorem 7 (see, for example, (Greenwald and Zinkevich, 2005)).

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 $a_i, a_i' \in A_i(s)$,*

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

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

Unlike in one-shot games where only the $\pi(a_{-i}, a_i)$'s are unknown (see Equation 2), here the $\pi_s(a_{-i}, a_i)$'s, and hence the $Q_i^\pi(s, (a_{-i}, a_i))$'s, are unknown. In particular, Equation 10 is not a system of linear inequalities, but rather a system of nonlinear inequalities. Next, we propose a class of iterative algorithms, based on the correlated equilibrium solution concept, and we investigate the question of whether or not any algorithms in this class converge to correlated equilibrium policies in Markov games, effectively solving this nonlinear system.

3. Multiagent Q-Learning

In a companion paper (Greenwald and Zinkevich, 2005), we rely on Kakutani's fixed point theorem to establish the existence of stationary correlated equilibrium policies in Markov games. Specifically, we define a correspondence, the fixed points of which are the stationary correlated equilibrium policies of a Markov game, and we show that this correspondence satisfies Kakutani's sufficient conditions, ensuring that the set of such fixed points is nonempty.

The definition of this correspondence suggests an algorithm for computing one of its fixed points, that is, a *global* equilibrium policy, 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 the remainder of this paper, we investigate the question of whether or not instances of this procedure converge to correlated equilibrium policies in Markov games.

In MDPs, the special case of Markov games with only a single agent, the corresponding local update procedure, known as value iteration, is well-known: 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 $t + 1$,

$$V^{t+1}(s) := \max_{a \in A(s)} Q^t(s, a) \tag{11}$$

$$Q^{t+1}(s, a) := (1 - \gamma) R(s, a) + \gamma \sum_{s' \in S} P[s' | s, a] V^{t+1}(s') \tag{12}$$

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 (sets of) joint distributions; at time $t+1$,

$$V_i^{t+1}(s) := \sum_{a \in A(s)} \pi_s^t(a) Q_i^t(s,a) \tag{13}$$

$$Q_i^{t+1}(s,a) := (1-\gamma) R_i(s,a) + \gamma \sum_{s' \in S} P[s'|s,a] V_i^{t+1}(s') \tag{14}$$

$$\pi_s^{t+1} \in f(Q^{t+1}(s)) \tag{15}$$

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

Following the literature on this subject (e.g., Littman (1994, 2001); Hu and Wellman (2003)), we experiment with "correlated-Q learning," in which values and Q-values are updated asynchronously (see Tables 1 and 2), rather than "correlated value iteration," in which these values are updated synchronously, as suggested by Equations 13, 14, and 15.

Finally, one important application-specific issue arises: can we assume the existence of a trusted third party who can act as a centralized coordinator? Or need we decentralize the implementation of multiagent value iteration and Q-learning? We present two generic formulations of multiagent Q-learning: the first is centralized; the second is decentralized.

3.1 Centralized Multiagent Learning

The generalization of dynamic programming and reinforcement learning from MDPs to Markov games is straightforward, if one assumes the existence of a trusted third party who serves as a central coordinator. A template for *centralized* multiagent Q-learning, is shown in Table 1. Notably, in Step 3, the central coordinator, who has knowledge of all agents' Q-tables, selects a joint distribution on which the agents updates in Step 4 rely.

3.2 Decentralized Multiagent Learning

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

3.3 Correlated-Q Learning

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

CENTRALIZEDQ(Γ, f, g, α)	
Inputs	game Γ, selection mechanism f, decay schedule g, learning rate α
Output	values V, Q-values Q, joint policy π^*
Initialize	Q-values Q, state s, action profile a

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

FOREVER

Table 1: Multiagent Q-Learning: Centralized.

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

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

FOREVER

Table 2: Multiagent Q-Learning: Decentralized.

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

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a) Q_i(s,a) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a) Q_i(s,(a_{-i}, a'_i)) \tag{16}$$

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

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

$$\max_{\pi_s \in \Delta(A(s))} \sum_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{17}$$

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

$$\max_{\pi_s \in \Delta(A(s))} \min_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{18}$$

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

$$\max_{\pi_s \in \Delta(A(s))} \max_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{19}$$

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

$$\max_{\pi_s \in \Delta(A(s))} \sum_{a \in A(s)} \pi_s(a) Q_i(s,a) \tag{20}$$

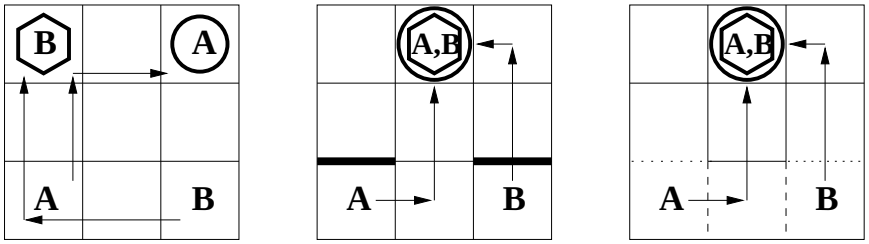
In our experimental discussion, we abbreviate these variants of correlated-Q (CE-Q) learning as uCE-Q, eCE-Q, pCE-Q, and dCE-Q, respectively. Three out of four of our implementations of CE-Q learning are centralized: uCE-Q, eCE-Q, and pCE-Q learning; however, dCE-Q learning is decentralized: each agent i learns as if it is the dictator.

3.4 Experimental Setup

In the next several sections, we describe experiments with various multiagent Q-learning algorithms on a standard test bed of Markov games, including the "grid games" as well as grid soccer. In doing so, we compare the performance of correlated-Q learning with other well-known multiagent Q-learning algorithms described in the literature: specifically, Q-learning (Watkins, 1989), minimax-Q (or foe-Q) learning (Littman, 1994), friend-Q learning (Littman, 2001), and two variants of Nash-Q learning (Hu and Wellman, 2003). We investigate the question of whether or not these multiagent Q-learning algorithms converge to equilibrium policies in general-sum Markov games.

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

In addition, we implemented centralized and decentralized versions of Nash-Q learning. We refer to these two algorithms as coordinated Nash-Q (cNE-Q) and best Nash-Q (bNE-Q), respectively. In the former, a central coordinator selects and broadcasts a Nash equilibrium



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

Figure 1: Grid games: Initial States and Sample Equilibria. Shapes indicate goals.

to all agents; in the latter, each agent independently selects that Nash equilibrium which maximizes its utility. Lastly, we implemented decentralized versions of foe-Q and friend-Q. In our implementations of foe-Q, friend-Q, and best Nash-Q, we allow the agents to observe their opponents Q-values. In our implementation of coordinated Nash-Q, it suffices for the central coordinator to observe all agents Q-values.

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

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

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

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

4.1 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.

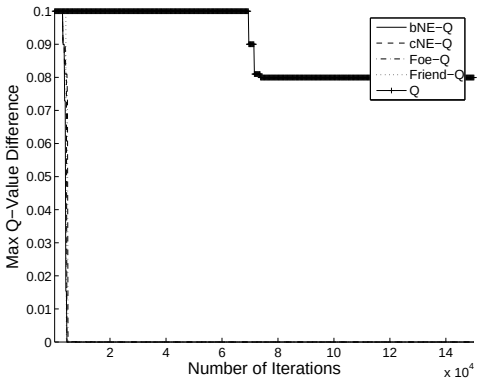
4.2 Empirical Convergence

Our experiments reveal that all of the multiagent Q-learning algorithms are converging in the three grid games. However, Q-learning does not converge in the grid games. Littman (2001) proves that FF-Q converges in general-sum Markov games. Hu and Wellman (2003) show empirically that NE-Q converges in both GG1 and GG2. Figure 2 shows that NE-Q is also converging in GG3. Similarly, CE-Q converges in all three grid games. We cannot, however, make any claims about the convergence of CE-Q in general.

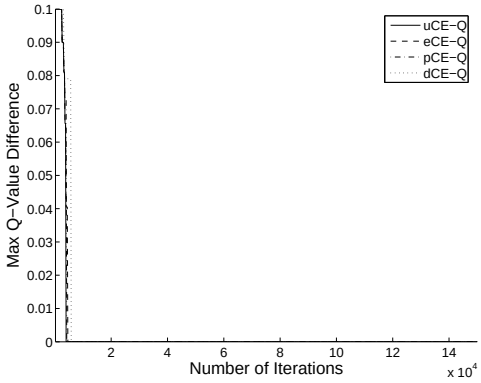
The values plotted in Figure 2 are computed as follows. Define an error term ERR_i^t at time t for agent i as the difference between $Q(s^t, a^t)$ at time t and $Q(s^t, a^t)$ at time $t - 1$: i.e., $\text{ERR}_i^t = \left|Q_i^t(s^t, a^t) - Q_i^{t-1}(s^t, a^t)\right|$. The error values on the y-axis depict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

In our experiments, the parameters are set as follows. Our implementation of Q-learning is on-policy and ϵ-greedy, with $\epsilon = 0.01$ and $\alpha = 1/n(s, a)$, where $n(s, a)$ is the number of visits to state-action pair (s, a). All other algorithms are off-policy (equivalently, on-policy and 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, like Q-learning, $\alpha = 1/n(s, a)$. Finally, $\gamma = 0.9$ in all cases. Next, we investigate the policies learned by the algorithms.

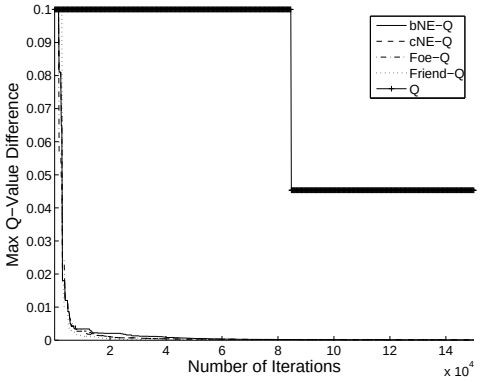
5. Setting $T' = T$ is sometimes misleading: It could appear that non-convergent algorithms are converging, because our metric measures the maximum error between the current time and the end of the simulation, but it could be that the change in Q-values is negligible for all states visited at the end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



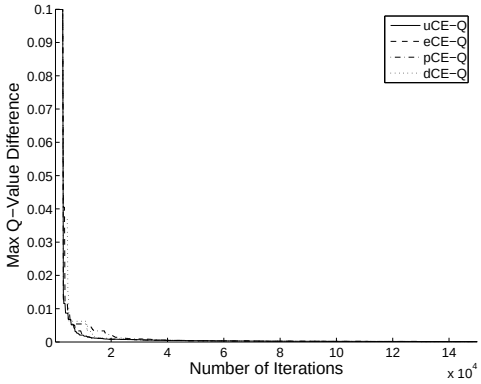
(a) Grid Game 1



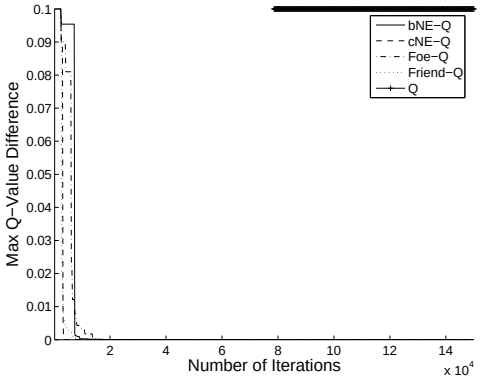
(b) Grid Game 1: CE-Q



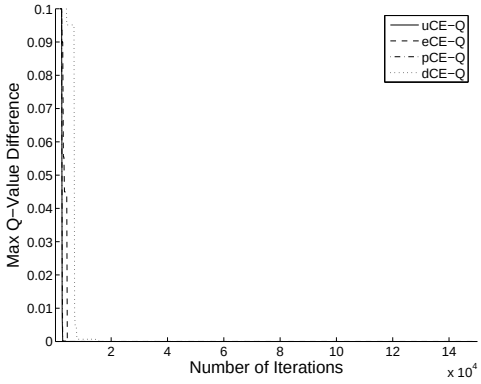
(c) Grid Game 2



(d) Grid Game 2: CE-Q



(e) Grid Game 3



(f) Grid Game 3: CE-Q

Figure 2: Changing Q-values in the grid games: all algorithms except Q-learning in all games and eCE-Q in GG1 are converging. For all algorithms except Q-learning: in GG1 and GG3, where there is no stochasticity, $\alpha = 1$; in GG2, $\alpha = 1/n(s,a)$, where $n(s,a)$ is the number of visits to state-action pair (s,a). Our Q-learning implementation is on-policy and ϵ-greedy with $\epsilon = 0.01$ and $\alpha = 1/n(s,a)$.

4.3 Equilibrium Policies

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

- Q-learning does not converge, and it does not learn equilibrium policies;
- friend-Q and foe-Q learning converge, but need not learn equilibrium policies;
- NE-Q and CE-Q learn equilibrium policies, whenever they converge.

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 according to the policies they learned. Our learning phase is randomized: not only are the state transitions stochastic, on-policy Q-learners and off-policy multiagent Q-learners can all make probabilistic decisions. Thus, if there exist multiple equilibrium policies in a game, agents can learn different equilibrium policies across different runs. Moreover, since agents can learn stochastic policies, scores can vary across different test runs. Nonetheless, we presented only one run of the learning phase (see Section 4.2) 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, during which the agents played the grid games repeatedly, are depicted in Table 3. 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.

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 converge to an equilibrium policy (Littman, 2001). Indeed, friend-Q learns Q-values that support equilibrium policies, but in our decentralized implementation of friend-Q learning, friends lack the ability to coordinate their play. Whenever these so-called "friends" choose policies that collide, both agents obtain negative scores for the remainder of the simulation: e.g., if the agents' policies lead them to one another's goals, both agents move towards the center ever after. In our experiments, friend-Q learned a stochastic policy[6] 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.

On-policy Q-learning is not successful in the grid games: it learns equilibrium policies in GG1, but in GG2, it learns a foe-Q-like (non-equilibrium) policy.[7] As expected, we have

6. 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.
7. Although Q-learning did not converge in the grid games, the policies appeared stable at the end of the learning phase. By decaying α, we disallow large changes in the agents' Q-values, which makes changes in their policies less and less frequent.

GG1	Avg. Score	Games	Convergence?	Eqm. Values?	Eqm. Play?
Q	100,100	2500	No	Yes	Yes
Foe-Q	0,0	0	Yes	No	No
Friend-Q	$-3239, -3239$	3	Yes	Yes	No
uCE-Q	100,100	2500	Yes	Yes	Yes
eCE-Q	100,100	2500	Yes	Yes	Yes
pCE-Q	100,100	2500	Yes	Yes	Yes
cNE-Q	100,100	2500	Yes	Yes	Yes
dCE-Q	$-10^4, -10^4$	0	Yes	Yes	No
bNE-Q	$-10^4, -10^4$	0	Yes	Yes	No

GG2	Avg. Score	Games	Convergence?	Eqm. Values?	Eqm. Play?
Q	67.3,66.2	3008	No	No	No
Foe-Q	65.9,67.4	3011	Yes	No	No
Friend-Q	$-10^4, -10^4$	0	Yes	No	No
uCE-Q	50.4,100	3333	Yes	Yes	Yes
eCE-Q	49.5,100	3333	Yes	Yes	Yes
pCE-Q	50.3,100	3333	Yes	Yes	Yes
cNE-Q	100,50.2	3333	Yes	Yes	Yes
dCE-Q	49.9,100	3333	Yes	Yes	Yes
bNE-Q	100,49.7	3333	Yes	Yes	Yes

GG3	Avg. Score	Games	Convergence?	Eqm. Values?	Eqm. Play?
Q	62.6,95.4	3314	No	No	No
Foe-Q	120,120	3333	Yes	No	No
Friend-Q	$-25 \times 10^4, -25 \times 10^4$	0	Yes	No	No
uCE-Q	117,117	3333	Yes	Yes	Yes
eCE-Q	117,117	3333	Yes	Yes	Yes
pCE-Q	100,125	3333	Yes	Yes	Yes
cNE-Q	125,100	3333	Yes	Yes	Yes
dCE-Q	$-25 \times 10^4, -25 \times 10^4$	0	Yes	Yes	No
bNE-Q	$-25 \times 10^4, -25 \times 10^4$	0	Yes	Yes	No

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

found that the Q-learning algorithm does not converge in general; moreover, the "stable" policies that we tested need not be equilibrium policies. Indeed, this observation is the underlying motivation for multiagent Q-learning research.

4.3.1 CE-Q and NE-Q Learning

In GG1, uCE-Q, e-CE-Q, pCE-Q, and cNE-Q all learn Q-values that coincide exactly with those of friend-Q: i.e., Q-values that support stationary equilibrium policies. But unlike friend-Q, these variants of CE-Q and NE-Q always obtain positive scores. In our implementation of CE-Q, a centralized mechanism broadcasts an equilibrium, even during testing. Thus, CE-Q play is always coordinated, and uCE-Q, eCE-Q, pCE-Q, and cNE-Q learners do not collide while playing the grid games. In our implementation of cNE-Q, however, the agents make independent decisions according to their coordinated equilibrium policies during testing. Nonetheless, since the cNE-Q agents learn coordinated equilibrium policies in GG1, they play the game 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 dCE-Q and bNE-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 dCE-Q and bNE-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 converge to stationary equilibrium policies. Interestingly, the asynchronous updating that is characteristic of Q-learning converts this symmetric game into a dominance-solvable game: 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 uCE-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 dCE-Q and bNE-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, CE-Q learning is indifferent between all stationary correlated equilibrium policies, pure and mixed, since they all yield equivalent rewards to all players. In GG3, however, both uCE-Q and eCE-Q learn the particular correlated equilibrium policy

that yields symmetric scores, because both the sum and the minimum of the agents' rewards at this equilibrium exceed those of any other equilibrium policies. Consequently, the sum of the scores of uCE-Q and eCE-Q exceed that of any Nash equilibrium. CE-Q's rewards do 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 policies that 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.

5. Soccer Game

The grid games are general-sum Markov games for which there exist pure strategy stationary equilibrium policies. In this section, we consider a two-player, zero-sum Markov game for which there do not exist pure strategy equilibrium policies. Our game is a simplified version of the soccer game that is described in Littman (1994).

The soccer field is a grid (see Figure 3). 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.[8] 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.

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 3 (hereafter, state $\hat{s}$), any pure policy for player A is subject to indefinite blocking by player B; but if player A employs a mixed policy, then player A can hope to pass player B on his next move.

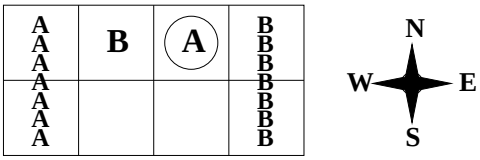
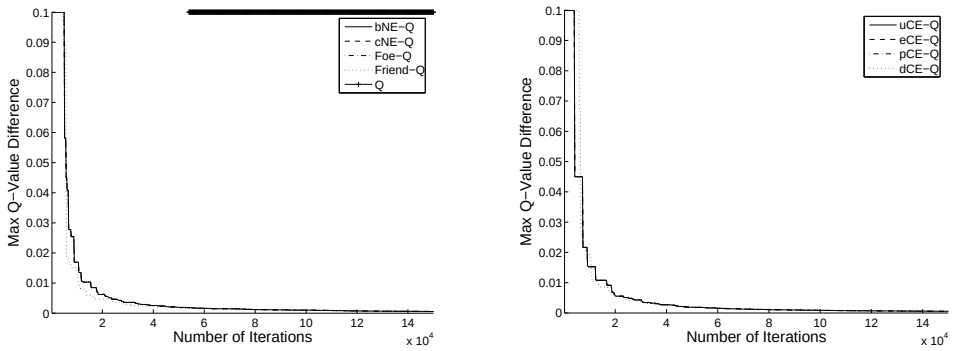


Figure 3: Soccer Game. The circle represents the ball. If player A moves W, he loses the ball to player B; but if player B moves E, attempting to steal the ball, he cannot.

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



(a) Grid Soccer (b) Grid Soccer: CE-Q

Figure 4: Changing Q-values in the soccer game: all algorithms are converging, except Q-learning. For all algorithms, the discount factor $\gamma = 0.9$ and the parameter $\alpha = 1/n(s,a)$, where $n(s,a)$ is the number of visits to state-action pair (s,a). Our Q-learning implementation is on-policy and ϵ-greedy with $\epsilon = 0.01$.

5.1 Empirical Convergence

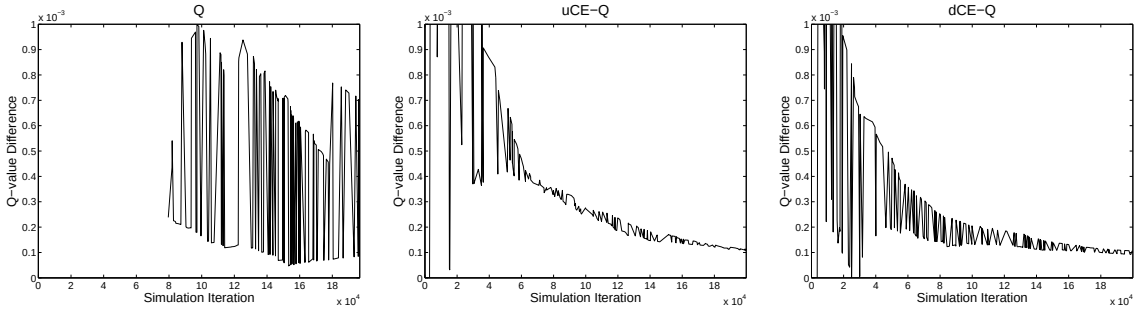
We experimented with the same set of algorithms in this soccer game as in the grid games. Consistent with the theory, friend-Q and foe-Q converge at all state-action pairs. Nash-Q also converges everywhere, as do all variants of correlated-Q—in this game, all equilibria at all states have equivalent values; thus, all correlated-Q operators yield identical outcomes. Moreover, correlated-Q, like Nash-Q, learns Q-values that coincide exactly with those of foe-Q. But Q-learning, as in the grid games, does not converge.

Figure 4 shows that while the multiagent-Q learning algorithms converge, Q-learning itself does not converge. Our implementation of 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 2, the y-values depict the maximum error from the current time x to the end of the simulation T: i.e., $\max_{t=x,\ldots,T} \text{ERR}_i^t = \max_{t=x,\ldots,T} \left|Q_i^t(s^t,a^t) - Q_i^{t-1}(s^t,a^t)\right|$, for $i = A$. The values on the x-axis, representing time, range from 1 to T', for some $T' < T$. As in our experiments with the grid games, we set $T = 1.5 \times 10^5$ and $T = 2 \times 10^5$.

Figure 5 presents an example of a state-action pair at which classic Q-learning does not converge. The values on the x-axis represent time, and the corresponding y-values are the error values $\text{ERR}_A^t = \left|Q_i^t(\hat{s},\text{S},\text{E}) - Q_i^{t-1}(\hat{s},\text{S},\text{E})\right|$. In Figure 5(a), although the Q-value differences are decreasing at times, they are not converging. They are decreasing only 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.

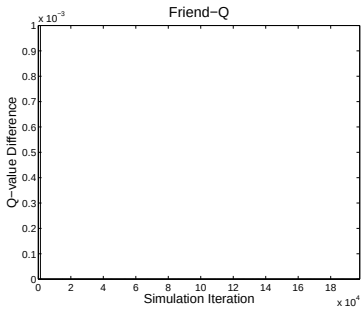
Friend-Q, however, converges to a pure policy for player A at state $\hat{s}$, namely W. Learning according to friend-Q, player A fallaciously anticipates the following sequence of events: player B sticks at state $\hat{s}$, and player A takes action W. By taking action W, player A passes the ball to 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.



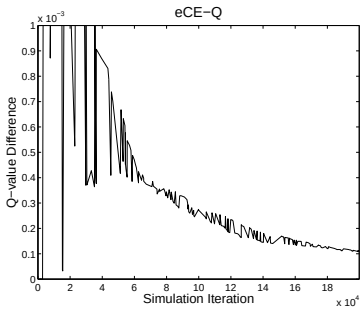
(a) Q-learning

(b) uCE-Q

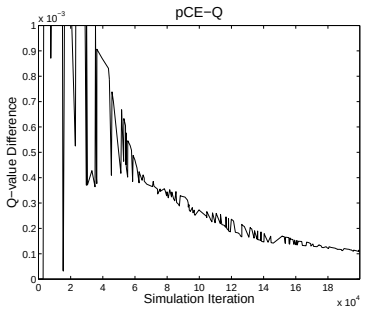
(c) dCE-Q



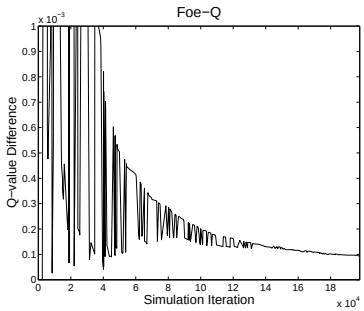
(d) Friend-Q



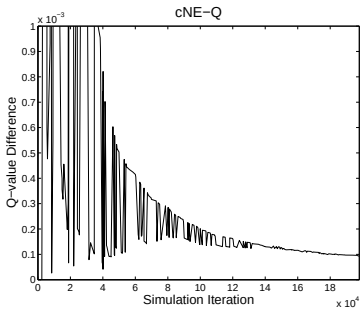
(e) eCE-Q



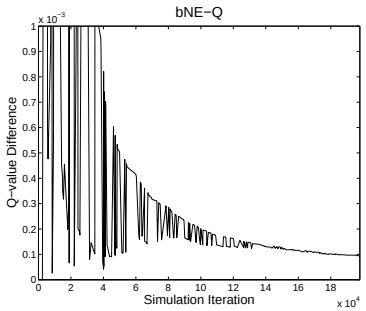
(f) pCE-Q



(g) Foe-Q



(h) cNE-Q



(i) bNE-Q

Figure 5: Changing Q-values at state $\hat{s}$. All algorithms are converging, except Q-learning.

In 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, cNE-Q, and bNE-Q are indistinguishable in such games. Indeed, Figures 5(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.

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
uCE-Q	2.30, -2.30	4051	Yes	Yes	Yes
eCE-Q	1.18, -1.18	4167	Yes	Yes	Yes
pCE-Q	1.12, -1.12	4104	Yes	Yes	Yes
cNE-Q	1.86, -1.86	4194	Yes	Yes	Yes
dCE-Q	-0.24, 0.24	4130	Yes	Yes	Yes
bNE-Q	0.84, -0.84	4304	Yes	Yes	Yes

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

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

5.2 Equilibrium Policies

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

In summary, in grid soccer, a two-player, zero-sum Markov game, Q-learning does not converge. Intuitively, the rationale for this outcome is clear: Q-learning seeks deterministic optimal policies, but in this game no such policies exist.[9] Friend-Q converges but its policies are not rational. Correlated-Q learning, however, like Nash-Q learning, learns the same Q-values as foe-Q learning. Nonetheless, correlated-Q learns possibly correlated equilibrium policies, while foe-Q and Nash-Q learn minimax equilibrium policies. In the next section, we offer some theoretical justification for our observations about correlated Q-learning.

6. Convergence of Correlated-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 CE-Q learning behaves like foe-Q learning in the former class of Markov games, and like friend-Q learning in the latter.

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

Let $\Gamma = \langle N, A, R \rangle$ denote a *one-shot game*. Recall from Section 2: N is a set of n players; $A = \prod_{i \in N} A_i$, where A_i is player i's set of pure actions; and $R : A \to \mathbb{R}^n$, where $R_i(a)$ is player i's reward at action profile $a \in A$. A **mixed strategy profile** $(\sigma_1, \ldots, \sigma_n) \in \Delta(A_1) \times \ldots \times \Delta(A_n)$ is a profile of randomized actions, one per player. Overloading our notation, we extend R to be defined over mixed strategies:

$$R_i(\sigma_1, \ldots, \sigma_n) = \sum_{a_1 \in A_1} \cdots \sum_{a_n \in A_n} \sigma_1(a_1) \ldots \sigma_n(a_n) R_i(a_1, \ldots, a_n) \tag{21}$$

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

$$R_i(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*) = \max_{\sigma_i} R_i(\sigma_1^*, \ldots, \sigma_i, \ldots \sigma_n^*) \tag{22}$$

6.1 Correlated-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. Hence, the empirical results obtained on the grid soccer game are not surprising.

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

$$R_1(\sigma_1^*, \sigma_2^*) = \max_{\sigma_1} R_1(\sigma_1, \sigma_2^*) \tag{23}$$

$$R_2(\sigma_1^*, \sigma_2^*) = \max_{\sigma_2} R_2(\sigma_1^*, \sigma_2) \tag{24}$$

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

6.1.1 Convergence

Lemma 9 *If Γ is a two-player, zero-sum one-shot game with Nash equilibrium (σ_1^*, σ_2^*), then*

$$R_1(\sigma_1^*, \sigma_2^*) \leq R_1(\sigma_1^*, \sigma_2), \quad \textit{for all } \sigma_2 \in \Delta(A_2) \tag{25}$$

$$R_2(\sigma_1^*, \sigma_2^*) \leq R_2(\sigma_1, \sigma_2^*), \quad \textit{for all } \sigma_1 \in \Delta(A_1) \tag{26}$$

Proof Because $R_1 = -R_2$, Equation 23 implies:

$$R_2(\sigma_1^*, \sigma_2^*) = -R_1(\sigma_1^*, \sigma_2^*) = -\max_{\sigma_1} R_1(\sigma_1, \sigma_2^*) = \min_{\sigma_1} -R_1(\sigma_1, \sigma_2^*) = \min_{\sigma_1} R_2(\sigma_1, \sigma_2^*) \tag{27}$$

so that $R_2(\sigma_1^*, \sigma_2^*) \leq R_2(\sigma_1, \sigma_2^*)$, for all $\sigma_1 \in \Delta(A_1)$. The proof for player 1 is analogous. ■

It is well-known that the value of all Nash equilibria in zero-sum games is unique (von Neumann and Morgenstern, 1944): i.e., for any Nash equilibria σ and σ', $R_i(\sigma) = R_i(\sigma')$, for all $i \in \{1, 2\}$. In the next theorem, we argue that any correlated equilibrium has the equivalent Nash (or minimax) equilibrium value.

Theorem 10 *If Γ is a two-player, zero-sum one-shot game with Nash equilibrium (σ_1^*, σ_2^*) and correlated equilibrium π, then $R_i(\pi) = R_i(\sigma_1^*, \sigma_2^*)$, for all $i \in \{1, 2\}$.*

Proof Consider player 1 and an action $a_1 \in A_1$ with $\pi(a_1) > 0$. By Lemma 9, $R_1(\sigma_1^*, \pi(\cdot|a_1)) \geq R_1(\sigma_1^*, \sigma_2^*)$. By the definition of correlated equilibrium, $R_1(a_1, \pi(\cdot|a_1)) \geq R_1(\sigma_1^*, \pi(\cdot|a_1))$. Hence, $R_1(a_1, \pi(\cdot|a_1)) \geq R_1(\sigma_1^*, \sigma_2^*)$. Because this condition holds for all $a_1 \in A_1$ with $\pi(a_1) > 0$, it follows that $R_1(\pi) \geq R_1(\sigma_1^*, \sigma_2^*)$. By an analogous argument, $R_2(\pi) \geq R_2(\sigma_1^*, \sigma_2^*)$. Because the game is zero-sum, in fact, $R_1(\pi) \leq R_1(\sigma_1^*, \sigma_2^*)$. Therefore, $R_1(\pi) = R_1(\sigma_1^*, \sigma_2^*)$. The argument is analogous for player 2. ■

Define $\mathrm{MM}_i(R)$ to be the minimax (equivalently, the Nash) equilibrium value of the ith player in a two-player, zero-sum one-shot game Γ. Similarly, define $\mathrm{CE}_i(R)$ to be the (unique) correlated equilibrium value of the ith player in a two-player, zero-sum one-shot game Γ. By Theorem 10, $\mathrm{CE}_i(R) = \mathrm{MM}_i(R)$.

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 11 *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 12 *Multiagent Q-learning preserves the zero-sum property in two-player, zero-sum Markov games, provided Q-values are initialized such that this property holds.*

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

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

$$\begin{aligned}
\pi_{s'}^{t+1} &\in f(Q^t(s')) && (28)\\
V_1^{t+1}(s') &:= \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_1^t(s', a') && (29)\\
V_2^{t+1}(s') &:= \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_2^t(s', a') && (30)\\
Q_1^{t+1}(s, a) &:= (1 - \alpha) Q_1^t(s, a) + \alpha((1 - \gamma) R_1(s, a) + \gamma V_1^{t+1}(s')) && (31)\\
Q_2^{t+1}(s, a) &:= (1 - \alpha) Q_2^t(s, a) + \alpha((1 - \gamma) R_2(s, a) + \gamma V_2^{t+1}(s')) && (32)
\end{aligned}$$

where f is any selection mechanism. By the induction hypothesis, $Q^t(s')$ is a zero-sum one-shot game. Hence, by Observation 11, $V \equiv V_1^{t+1}(s') = -V_2^{t+1}(s') \equiv -V$, so that the multiagent-Q learning update procedure simplifies as follows:

$$\begin{aligned}
Q_1^{t+1}(s, a) &:= (1 - \alpha) Q_1^t(s, a) + \alpha((1 - \gamma) R_1(s, a) + \gamma V) && (33)\\
Q_2^{t+1}(s, a) &:= (1 - \alpha) Q_2^t(s, a) + \alpha((1 - \gamma) R_2(s, a) - \gamma V) && (34)
\end{aligned}$$

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

Theorem 13 *If all Q-values are initialized such that the zero-sum property holds, then correlated-Q learning converges to the minimax equilibrium Q-values in two-player, zero-sum Markov games.*

Proof By Lemma 12, correlated-Q learning preserves 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, correlated-Q learning simplifies as follows: assuming action profile a is played at state s and the game transitions to state s', for all $i \in \{1,2\}$,

$$\begin{aligned} Q_i^{t+1}(s,a) &:= (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \mathrm{CE}_i(Q^t(s'))) && (35) \\ &= (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \mathrm{MM}_i(Q^t(s'))) && (36) \end{aligned}$$

Indeed, the correlated-Q and minimax-Q learning update procedures coincide, so that correlated-Q learning converges to minimax equilibrium Q-values in two-player, zero-sum Markov games, if all Q-values are initialized such that the zero-sum property holds. ■

Remark 14 *"Correlated value iteration," that is, synchronous updating of Q-values based on a correlated equilibrium selection mechanism, also converges to the minimax equilibrium Q-values in two-player, zero-sum Markov games, if all Q-values are initialized such that the zero-sum property holds.*

In summary, correlated-Q learning converges in two-player, zero-sum Markov games. In particular, correlated-Q learning converges to precisely the minimax equilibrium Q-values. This result applies to both centralized and decentralized versions of the learning algorithm.

6.1.2 Exchangeability

To guarantee that agents play equilibrium policies in a general-sum Markov game, it is not sufficient for agents to learn equilibrium Q-values. In addition, the agents must play an equilibrium at every state that is encountered while playing the game.

Specifically, to guarantee that agents play equilibrium policies in a general-sum Markov game, the agents must play an equilibrium in each of the one-shot games $Q^*(s)$, where $Q^*(s)$ is the set of Q-values the agents learn at state $s \in S$. In the repeated Bach or Stravinsky game (formulated as a deterministic, Markov game) with $\gamma = \frac{1}{2}$, egalitarian correlated-Q learning converges to the Q-values shown in Game 3. Two agents playing Game 3, and making independent decisions, can fail to coordinate their behavior, if, say, player 1 selects and plays her part of the equilibrium (b, B) and player 2 selects and plays his part of the equilibrium (s, S), so that the action profile the agents play is (b, S).

In the special case of two-player, zero-sum Markov games, however, it suffices for agents to learn minimax equilibrium Q-values, because miscoordination in two-player, zero-sum

Game 3: Repeated Bach or Stravinsky Q-values for $\gamma = \frac{1}{2}$

	B	S
b	$\frac{7}{4}, \frac{5}{4}$	$\frac{3}{4}, \frac{3}{4}$
s	$\frac{3}{4}, \frac{3}{4}$	$\frac{5}{4}, \frac{7}{4}$

one-shot games is ruled out by the exchangeability property. The exchangeability property holds in a one-shot game if, assuming each player i selects a correlated equilibrium π_i and plays his marginal distribution, call it π_{A_i}, the mixed strategy profile $(\pi_{A_i})_{i \in I}$ is a Nash equilibrium. Our proof that the exchangeability property holds (for the correlated equilibrium solution concept) in two-player, zero-sum one-shot games relies on the fact that exchangeability also holds of Nash equilibria in this class of games:

Lemma 15 *If Γ is a two-player, zero-sum one-shot game with Nash equilibria σ and σ', then (σ_1, σ_2') is a Nash equilibrium.*

Proof Since the minimax equilibrium value of a zero-sum game is unique, $R_1(\sigma) = R_1(\sigma')$. Observe that $R_1(\sigma_1, \sigma_2) \leq R_1(\sigma_1, \sigma_2')$, by Lemma 9 (Equation 25), and $R_1(\sigma_1, \sigma_2') \leq R_1(\sigma_1', \sigma_2')$, by the definition of minimax equilibrium (Equation 23). Hence, $R_1(\sigma_1, \sigma_2) = R_1(\sigma_1, \sigma_2') = R_1(\sigma_1', \sigma_2')$. Because $R_1 = -R_2$, it follows that $R_2(\sigma_1, \sigma_2) = R_2(\sigma_1, \sigma_2') = R_2(\sigma_1', \sigma_2')$. Now $R_2(\sigma_1, \sigma_2') = R_2(\sigma_1, \sigma_2) = \max_{\sigma_2''} R_2(\sigma_1, \sigma_2'')$ and $R_1(\sigma_1, \sigma_2') = R_1(\sigma_1, \sigma_2) = \max_{\sigma_1''} R_1(\sigma_1'', \sigma_2)$. Therefore, (σ_1, σ_2') is a Nash equilibrium. ■

Define π_{A_i} to be the marginal distribution of π over A_i: i.e.,

$$\pi_{A_i}(a_i) = \sum_{a_{-i} \in A_{-i}} \pi(a_{-i}, a_i) \tag{37}$$

Definition 16 *A one-shot game Γ satisfies the **exchangeability property** if for any set of n correlated equilibria π^1, π^2,...,π^n, it is the case that $(\pi^1_{A_1}, \ldots, \pi^n_{A_n})$ is a Nash equilibrium.*

Theorem 17 *The exchangeability property holds in two-player, zero-sum one-shot games.*

Proof By Lemma 15, it suffices to show that for any correlated equilibrium π, (π_{A_1}, π_{A_2}) is a Nash equilibrium. The proof of this statement relies on a series of lemmas.

Lemma 18 *In a two-player, zero-sum one-shot game Γ, if π is a correlated equilibrium, then $\max_{a_1'} R_1(a_1', \pi(\cdot \mid a_1)) = \mathrm{MM}_1(R)$, for all $a_1 \in A_1$ such that $\pi_{A_1}(a_1) > 0$.*

Proof By the definition of correlated equilibrium, for all $a_1 \in A_1$ such that $\pi_{A_1}(a_1) > 0$, a_1 is a best-response to $\pi(\cdot \mid a_1)$: i.e., $R_1(a_1, \pi(\cdot|a_1)) \geq R_1(a_1', \pi(\cdot|a_1))$, for all $a_1' \in A_1$. By Theorem 10, $R_1(\pi) = \mathrm{MM}_1(R)$. Therefore, $\max_{a_1' \in A_1} R_1(a_1', \pi(\cdot|a_1)) = R_1(a_1, \pi(\cdot|a_1)) = \mathrm{MM}_1(R)$, for all such $a_1 \in A_1$ such that $\pi_{A_1}(a_1) > 0$. ■

Corollary 19 *In a two-player, zero-sum one-shot game Γ, if π is a correlated equilibrium, then $\max_{\sigma_1} R_1(\sigma_1, \pi(\cdot \mid a_1)) = \mathrm{MM}_1(R)$, for all $a_1 \in A_1$ such that $\pi_{A_1}(a_1) > 0$.*

Lemma 20 *In a two-player, zero-sum one-shot game Γ, if (σ_1, σ_2) is a Nash equilibrium, then (σ_1, σ_2') is also a Nash equilibrium, for any σ_2' such that $\max_{\sigma_1'} R_1(\sigma_1', \sigma_2') = \mathrm{MM}_1(R)$.*

Proof By assumption, $R_1(\sigma_1, \sigma_2) = \mathrm{MM}_1(R)$. By Lemma 9 (Equation 25), $R_1(\sigma_1, \sigma_2') \geq \mathrm{MM}_1(R)$. But $R_1(\sigma_1', \sigma_2') \leq \mathrm{MM}_1(R)$, for all $\sigma_1' \in \Delta(A_1)$. Hence, $R_1(\sigma_1, \sigma_2') = \mathrm{MM}_1(R)$, which implies (i) σ_1 is a best-response to σ_2', and (ii) σ_2' is a best-response to σ_1, since $R_2(\sigma_1, \sigma_2') = -R_1(\sigma_1, \sigma_2') = -\mathrm{MM}_1(R) = \mathrm{MM}_2(R)$ and $R_2(\sigma_1, \sigma_2) = \mathrm{MM}_2(R)$. ■

Lemma 21 *In a two-player, zero-sum one-shot game Γ, for any $\lambda \in [0, 1]$ and for any two Nash equilibria σ and σ', if $\sigma_2'' = \lambda\sigma_2 + (1 - \lambda)\sigma_2'$, then (σ_1, σ_2'') is a Nash equilibrium.*

Proof By Lemma 15, σ_1 is a best response to σ_2'':

$$\begin{aligned} R_1(\sigma_1, \sigma_2'') &= \lambda R_1(\sigma_1, \sigma_2) + (1 - \lambda) R_1(\sigma_1, \sigma_2') && (38)\\ &= \lambda \left(\max_{\sigma_1''} R_1(\sigma_1'', \sigma_2) \right) + (1 - \lambda) \left(\max_{\sigma_1''} R_1(\sigma_1'', \sigma_2') \right) && (39)\\ &\geq \max_{\sigma_1''} R_1(\sigma_1'', \lambda\sigma_2 + (1 - \lambda)\sigma_2') && (40)\\ &= \max_{\sigma_1''} R_1(\sigma_1'', \sigma_2'') && (41) \end{aligned}$$

Equation 40 follow from the following fact: for any $f, g : X \to \mathbb{R}$ and for any $a, b \in \mathbb{R}$, $\max_{x \in X}(af(x) + bg(x)) \leq a(\max_{x \in X} f(x)) + b(\max_{y \in X} g(y))$.

By Lemma 15, σ_2'' is a best response to σ_1:

$$\begin{aligned} R_2(\sigma_1, \sigma_2'') &= \lambda R_2(\sigma_1, \sigma_2) + (1 - \lambda) R_2(\sigma_1, \sigma_2') && (42)\\ &= \lambda \left(\max_{\sigma_2'''} R_2(\sigma_1, \sigma_2''') \right) + (1 - \lambda) \left(\max_{\sigma_2'''} R_2(\sigma_1, \sigma_2''') \right) && (43)\\ &= \max_{\sigma_2'''} R_2(\sigma_1, \sigma_2''') && (44) \end{aligned}$$

Therefore, (σ_1, σ_2'') is a Nash equilibrium. ■

By Corollary 19, player 1 achieves his minimax payoff whenever he plays a best-response to $\pi(\cdot \mid a_1)$, for all $a_1 \in A_1$ such that $\pi_{A_1}(a_1) > 0$. By Lemma 20, $\pi(\cdot \mid a_1)$ is player 2's part of a Nash equilibrium, and by repeated application of Lemma 21, any convex combination of $\pi(\cdot \mid a_1)$'s, is player 2's part of a Nash equilibrium, for all such $a_1 \in A_1$.

In particular, π_{A_2} is player 2's part of a Nash equilibrium, since

$$\begin{aligned} \pi_{A_2}(a_2) &= \sum_{a_1 \in A_1} \pi(a_1, a_2) && (45)\\ &= \sum_{\{a_1 \in A_1 \mid \pi(a_1) > 0\}} \pi(a_2 \mid a_1)\pi(a_1) && (46) \end{aligned}$$

implying

$$\pi_{A_2} = \sum_{\{a_1 \in A_1 | \pi(a_1) > 0\}} \pi(\cdot \mid a_1)\pi(a_1) \tag{47}$$

Analogously, one can show that π_{A_1} is player 1's part of a Nash equilibrium. Therefore, by Lemma 15, (π_{A_1}, π_{A_2}) is a Nash equilibrium. ■

6.2 Correlated-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 22 *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 23 *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',

$$\pi_{s'}^{t+1} \in f(Q^t(s')) \tag{48}$$

$$V_i^{t+1}(s') := \sum_{a' \in A} \pi_{s'}^{t+1}(a') Q_i^t(s', a') \tag{49}$$

$$Q_i^{t+1}(s, a) := (1 - \alpha) Q_i^t(s, a) + \alpha((1 - \gamma) R_i(s, a) + \gamma V_i^{t+1}(s')) \tag{50}$$

where f is any selection mechanism. By the induction hypothesis, $Q^t(s')$ is a common-interest one-shot game. Hence, by Observation 22, $V_i^{t+1}(s') = V_j^{t+1}(s') \equiv V$, for all $i, j \in N$, so that the multiagent-Q learning update procedure simplifies as follows:

$$Q_i^{t+1}(s, a) := (1 - \alpha) Q_i^t(s, a) + \alpha((1 - \gamma) R_i(s, a) + \gamma V) \tag{51}$$

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

Given a one-shot game, an equilibrium 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 24 *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 25 *If all Q-values are initialized such that the common-interest property holds, then any multiagent Q-learning algorithm that selects Pareto-optimal equilibria converges to friend Q-values in common-interest Markov games.*

Proof By Lemma 23, 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 24, any multiagent-Q learning update procedure with a Pareto-optimal selection mechanism simplifies as follows: for all $i \in N$,

$$Q_i^{t+1}(s,a) \quad := \quad (1-\alpha)Q_i^t(s,a) + \alpha((1-\gamma)R_i(s,a) + \gamma \max_{a' \in A(s')} Q_i^t(s',a')) \qquad (52)$$

assuming action profile a is played at state s and the game transitions to state s'. Indeed, the correlated-Q 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 26 *If all Q-values are initialized such that the common-interest property holds, then any multiagent Q-learning algorithm that selects Pareto-optimal equilibria converges to Pareto-optimal equilibrium Q-values in common-interest Markov games.*

The multiagent Q-learning algorithms uCE-Q, eCE-Q, pCE-Q, dCE-Q, and bNE-Q all rely on Pareto-optimal equilibrium selection operators, and thus converge to Pareto-optimal equilibrium Q-values in common-interest Markov games, provided Q-values are initialized such that the common-interest property holds.

Observe that an arbitrary CE-Q learner need not learn Q-values that correspond to Pareto-optimal equilibrium policies in common-interest Markov games, because an arbitrary CE-Q operator need not select Pareto-optimal equilibria. Similarly, cNE-Q need not select such equilibria nor learn such Q-values.

7. 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 in and of itself a hard problem (e.g., Papadimitriou (2001)). 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 is based on a variable learning rate.

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

8. Conclusion

This research originates with a proof of the existence of stationary correlated equilibrium policies in finite, discounted, general-sum Markov games (Greenwald and Zinkevich, 2005), which motivates the design of a class of multiagent Q-learning algorithms we call correlated Q-learning. Theoretically, we established that correlated-Q learning converges to stationary correlated equilibrium policies in two special classes of Markov games, namely zero-sum and common-interest. Empirically, we established that correlated-Q learning converges to stationary correlated equilibrium policies on a standard test bed of Markov games. Our empirical findings suggest that like Nash-Q, correlated Q-learning can serve an effective heuristic for the computation of equilibrium policies in general-sum Markov games. However, correlated-Q learning is more efficient than Nash-Q learning.

In past work, we have studied adaptive algorithms for learning game-theoretic equilibria in repeated games (Greenwald and Jafari, 2003). In ongoing work, we are combining these adaptive algorithms with multiagent Q-learning. Specifically, we are replacing the linear programming call in CE-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)). (No adaptive algorithm is known to converge to Nash equilibrium.) This adaptive approach 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.

R. Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

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.

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

Vincent Conitzer and Tuomas Sandholm. Complexity results about Nash equilibria. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 765–771, August 2003.

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.

Itzhak Gilboa and Eitan Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

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 7, Brown University, Department of Computer Science, June 2005.

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

M. Littman. 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. 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 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.

C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, 2001.

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.

K. Shell. General equilibrium: The new palgrave. In J. Eatwell, M. Milgate, and P. Newman, editors, *Sunspot Equilibrium*, pages 274–280. Macmillan, New York, 1989.

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

J. von Neumann and O. Morgenstern. *The Theory of Games and Economic Behavior.* Princeton University Press, Princeton, 1944.

C. Watkins. *Learning from Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, 1989.