Correlated-Q Learning

Amy Greenwald AMY@CS.BROWN.EDU
Keith Hall KH@CS.BROWN.EDU
Department of Computer Science, Brown University, Box 1910, Providence, RI 02912

Abstract

This paper introduces Correlated-Q (CE-Q) learning, a multiagent Q-learning algorithm based on the correlated equilibrium (CE) solution concept. CE-Q generalizes both Nash-Q and Friend-and-Foe-Q: in general-sum games, the set of correlated equilibria contains the set of Nash equilibria; in constant-sum games, the set of correlated equilibria contains the set of minimax equilibria. This paper describes experiments with four variants of CE-Q, demonstrating empirical convergence to equilibrium policies on a testbed of general-sum Markov games.

1. Introduction

Recently, there have been several attempts to design a multiagent learning algorithm that learns equilibrium policies in general-sum Markov games, just as Q-learning converges to optimal policies in Markov decision processes. Hu and Wellman [8] propose an algorithm called Nash-Q that converges to Nash equilibrium policies under certain (restrictive) conditions. Littman's [11] friend-or-foe-Q (FF-Q) algorithm always converges, but it only learns equilibrium policies in restricted classes of games: *e.g.*, two-player, constant-sum Markov games, which exhibit minimax equilibria (foe-Q); *e.g.*, coordination games with uniquely-valued equilibria (friend-Q).

This paper introduces Correlated-Q (CE-Q) learning, a multiagent Q-learning algorithm based on the correlated equilibrium solution concept [1]. CE-Q generalizes both Nash-Q and FF-Q: in general-sum games, the set of correlated equilibria contains the set of Nash (and thus, coordination) equilibria; in constant-sum games, where Nash and minimax equilibria coincide, the set of correlated equilibria contains the set of minimax equilibria.

A Nash equilibrium (NE) is a vector of *independent* probability distributions over actions, in which all agents optimize with respect to one another's probabilities. A correlated equilibrium (CE) is more general than a NE, since it permits dependencies among the agents' probability distributions, while maintaining the property that agents are optimizing. 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 0.5 and (GO,STOP) with probability 0.5. No probability mass is assigned to (GO,GO) or (STOP,STOP). Note that it is optimal for agents to obey their respective traffic signals.

The set of CE is a convex polytope; thus, unlike Nash equilibria (NE), CE can be computed easily via linear programming. Also, CE that are not NE can achieve higher rewards than NE, by avoiding positive probability mass on less desirable outcomes, unlike mixed strategy Nash equilibria. Also unlike NE, to which no learning algorithm is known to converge in general, no-regret algorithms (*e.g.*, Foster and Vohra [2]) converge to CE in repeated games. Finally, CE is consistent with the usual AI model of independent agent behavior: 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 games, there exist multiple equilibria with multiple payoff values. We attempt to resolve this equilibrium selection problem by introducing four variants of CE-Q, based on four equilibrium selection functions. We define utilitarian, egalitarian, republican, and libertarian CE-Q learning. This paper demonstrates empirical convergence to equilibrium policies for all four CE-Q variants on a testbed of Markov games. We also discuss the theory of stochastic stability, which could be employed to describe the convergence properties of our algorithms.

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

2. Markov Games

Stochastic games generalize repeated games and Markov decision processes (MDPs). A *stochastic game* is a tuple $\langle I, S, (A_i(s))_{s \in S, 1 \le i \le n}, P, (R_i)_{1 \le i \le n} \rangle$, where I is a set of n players, S is a set of states, $A_i(s)$ is the ith player's set of actions at state s, P is a probability transition function that describes state transitions, conditioned on past states and joint actions, and $R_i(s, \vec{a})$ is the ith player's reward for state $s \in S$ and joint actions $\vec{a} \in A(s) = A_1(s) \times \ldots \times A_n(s)$. Stochastic games for which the probability transitions satisfy the Markov property are called *Markov games*: *i.e.*, for $\vec{a}_t = (a_1, \ldots, a_n)_t$, $P[s_{t+1} | s_t, \vec{a}_t, \ldots, s_0, \vec{a}_0] = P[s_{t+1} | s_t, \vec{a}_t]$.

An MDP is a one-player Markov game. Recall Bellman's equations that characterize the optimal state- and action-values for a single agent and an MDP:

$$Q^*(s,a) = (1-\gamma) R(s,a) + \gamma \sum_{s'} P[s'|s,a] V^*(s') \quad (1)$$

$$V^*(s) = \max_{a \in A(s)} Q^*(s,a) \quad (2)$$

or $0 \le \gamma < 1$. In words, the value $Q^*(s,a)$ is the normalized sum of the immediate reward obtained at state s for taking action a and the expected discounted value of the future rewards obtained by following the optimal policy thereafter. The value function $V^*(s)$ at state s is defined as the value of the action a that maximizes $Q^*(s,a)$. The actions that maximize $Q^*(s,a)$ at each state s describe the (deterministic) optimal policy π^*: *i.e.*,

$$\pi^*(s) \in \arg \max_{a \in A(s)} Q^*(s,a) \quad (3)$$

In Markov games, player i's Q-values are defined over states and action-vectors $\vec{a} = (a_1, \ldots, a_n)$, rather than state-action pairs:

$$Q_i(s, \vec{a}) = (1-\gamma) R_i(s, \vec{a}) + \gamma \sum_{s'} P[s'|s, \vec{a}] V_i(s') \quad (4)$$

Intuitively, the notion of state-value function also carries over from MDPs to Markov games. But the obvious analogue of Eq. 2, in which all players maximize their respective rewards with respect to one another's actions is not adequate, since (deterministic) actions that satisfy these simultaneous equations need not exist. (As a consequence, Markov games need not exhibit deterministic equilibrium policies: *e.g.*, *Rochambeau*—Rock-paper-scissors.)

Several alternative definitions of the value function have been proposed. Littman [10] studied two-player, zero-sum Markov games and von Neumann's minimax value function [14]. Let $\Sigma_i(s)$ be the probabilistic action space of player i at state s. Now, if player 1 is the maximizer and player 2 is the minimizer, then

$$V_1(s) = \max_{\sigma_1 \in \Sigma_1(s)} \min_{a_2 \in A_2(s)} Q_1(s, \sigma_1, a_2) = -V_2(s) \quad (5)$$

where $Q(s, \sigma_1, a_2) = \sum_{a_1 \in A_1} \sigma_1(a_1) Q(s, a_1, a_2)$. At the opposite extreme, Littman's friend-Q [11] value function is suited to coordination games—games for which all the players' reward functions are equivalent—with uniquely-valued equilibria:

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

For the general case of n-player, general-sum games, Hu and Wellman [8] proposed the following definition of the value function:

$$V_i(s) \in \text{NASH}_i(Q_1(s), \ldots, Q_n(s)) \quad (7)$$

and $Q_i(s)$ denotes the ith player's reward matrix at state s, and $\text{NASH}_i(X_1, \ldots, X_n)$ denotes the ith player's reward according to some Nash equilibrium in the general-sum game determined by reward matrices $X_1, \ldots, X_n$. Note that existence of such values, which is implied by Nash's theorem [13] relies on probabilistic actions. This definition generalizes the minimax value function, since Nash equilibria and minimax strategies coincide in zero-sum games. But this value function need not be well-defined: in general, the set of Nash equilibria need not be a singleton.

We propose an alternative definition of the value function in Markov games:

$$V_i(s) \in \text{CE}_i(Q_1(s), \ldots, Q_n(s)) \quad (8)$$

where $\text{CE}_i(X_1, \ldots, X_n)$ denotes the ith player's reward according to some *correlated equilibrium* in the general-sum game determined by the rewards $X_1, \ldots, X_n$. Eq. 8 generalizes Eq. 7, since a Nash equilibrium is a correlated equilibrium that can be factored into independent distributions over each individual player's action space. Thus, equilibria that are consistent with Eq. 8 exist, but this value function, too, need not be well-defined.

For each choice of value function, it is necessary to establish the existence of Q-values that support equilibrium policies defined by the value function. In particular, we seek a solution to the system of equations given by Eq. 4 and either Eq. 5, 6, 7, or 8: *i.e.*, a set of action-values Q^*, and corresponding state-values V^*. The existence of such solutions are established in Greenwald [5], using Kakutani's and Brouwer's fixed point theorems. Most of these results were known previously, but new and direct proofs of these four results are presented in this recent work.

2.1. Correlated Equilibrium

A Nash equilibrium (NE) is a vector of *independent* probability distributions over actions, in which all agents optimize with respect to one another's probabilities. A correlated equilibrium (CE) allows for the possibility of dependencies in the agents' randomizations: a CE is a probability distribution over the *joint* space of actions, in which all agents optimize with respect to one another's probabilities, conditioned on their own.

In contrast to Nash equilibria, for which no efficient method of computation is known, correlated equilibria can be computed easily via linear programming. As an example, consider "Chicken" a two-player, two-action, one-shot, general-sum game.

	L	R
T	6,6	2,7
B	7,2	0,0

The correlated equilibria in this game are described by the probability constraints $\pi_{TL}+\pi_{TR}+\pi_{BL}+\pi_{BR}=1$ and $\pi_{TL},\pi_{TR},\pi_{BL},\pi_{BR}\geq 0$ together with the following so-called "rationality" constraints:

$$\begin{array}{rr} -1\pi_{TL}+2\pi_{TR}\geq 0 & \quad -1\pi_{TL}+2\pi_{BL}\geq 0 \\ 1\pi_{BL}-2\pi_{BR}\geq 0 & \quad 1\pi_{TR}-2\pi_{BR}\geq 0 \end{array}$$

These constraints have a natural interpretation in terms of conditional probabilities. Let $\pi(y|x)$ denote the conditional probability of y given x, and let $\pi(y)=\sum_x \pi(y|x)$ denote the marginal probability of y. Now $\pi_{TL}=\pi(L|T)\pi(T)$ and $\pi_{TR}=\pi(R|T)\pi(T)$. Thus, the first constraint, which describes row's rewards, can be restated as $-1\pi(L|T)+2\pi(R|T)\geq 0$, or equivalently, $6\pi(L|T)+2\pi(R|T)\geq 7\pi(L|T)+0\pi(R|T)$. Intuitively, the expected reward to the row player of action T is at least that of action B whenever he in fact plays action T. The other three rationality constraints can be interpreted analogously.

3. Multiagent Q-Learning

In principle, the generalization of dynamic programming and reinforcement learning from MDPs to Markov games is straightforward. A template for multiagent Q-learning, is presented in Table 1. In this generic formulation, the algorithm takes as input an equilibrium selection function f, which computes the value function V, given matrix-vector $\vec{Q}=(Q_1,\ldots,Q_n)$. Littman's FF-Q algorithm computes V according to either Eq. 5 or Eq. 6, as appropriate. Hu and Wellman's Nash-Q algorithm computes V according to Eq. 7. *Correlated-Q* computes V via Eq. 8.

MULTIQ(MarkovGame, f,γ,α,S,T)

Inputs	selection function f
	discount factor γ
	learning rate α
	decay schedule S
	total training time T
Output	state-value functions V_i^*
	action-value functions Q_i^*
Initialize	$s,a_1,\ldots,a_n$ and $Q_1,\ldots,Q_n$

```
for t = 1 to T
  1. simulate actions a_1,...,a_n in state s
  2. observe rewards R_1,...,R_n and next state s'
  3. for i = 1 to n
     (a) V_i(s') = f_i(Q_1(s'),...,Q_n(s'))
     (b) Q_i(s,a⃗) = (1-α)Q_i(s,a⃗)
                    + α[(1-γ)R_i + γV_i(s')]
  4. agents choose actions a'_1,...,a'_n
  5. s = s', a_1 = a'_1, ..., a_n = a'_n
  6. decay α according to S
```

Table 1. Multiagent Q-Learning.

3.1. CE-Q

The difficulty in learning equilibria in Markov games stems from the equilibrium selection problem: how can multiple agents select among multiple equilibria? We introduce four variants of correlated-Q learning, based on four correlated equilibrium selection mechanisms. Each variant of CE-Q learning resolves the equilibrium selection problem with its respective choice of objective function, which ensures that the equilibrium *value* of a game is unique, albeit not the equilibrium policy.

1. maximize the *sum* of the players' rewards:

$$\sigma\in\arg\max_{\sigma\in\mathrm{CE}}\sum_{i\in I}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{9}$$

2. maximize the *minimum* of the players' rewards:

$$\sigma\in\arg\max_{\sigma\in\mathrm{CE}}\min_{i\in I}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{10}$$

3. maximize the *maximum* of the players' rewards:

$$\sigma\in\arg\max_{\sigma\in\mathrm{CE}}\max_{i\in I}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{11}$$

4. maximize the *maximum* of each individual player i's rewards: let $\sigma=\prod_i \sigma^i$, where

$$\sigma^i\in\arg\max_{\sigma\in\mathrm{CE}}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{12}$$

Thus, $\mathrm{CE}_i(\vec{Q}(s)) = \{\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a})\}$, where σ satisfies either Eq. 9, 10, 11, or 12.

We refer to these algorithms as *utilitarian* (uCE-Q), *egalitarian* (eCE-Q), *republican* (rCE-Q), and *libertarian* (lCE-Q) correlated Q-learning, respectively. Note that all these equilibria can be computed via linear programming by incorporating the objective function of choice into the linear programming formulation (*i.e.*, the probability and rationality constraints) described in Sec. 2.1. Note also, the implementation of all four selection functions necessitates the sharing of Q-tables among agents. Hu and Wellman [8] resolve this issue by allowing all agents to observe all other agents' actions and rewards; thus, one agent can simulate another's update procedure, thereby maintaining a copy of the other's Q-table.

4. Grid Games

The first set of detailed experimental results on which we report pertain to grid games [9]. We describe three grid games: grid game 1 (GG1), a multi-state coordination game; grid game 2 (GG2), a stochastic version of Battle of the Sexes; and grid game 3 (GG3), a multi-state version of Chicken. In the following section, we describe experiments with grid soccer, a constant-sum Markov game that extends Matching Pennies.

Fig. 1 depicts the initial states of the three grid games. In GG1, there are two agents and two goals. The agents' action sets include one step in any of the four compass directions. Actions are executed simultaneously. If both agents attempt to move into the same cell, they cannot; instead, they both lose 1 point. If ever an agent reaches its goal, it scores 100 points, and the game ends. Note that it is possible for both agents to score 100 points since actions are simultaneous.

Other than the board setup, GG2 is identical to GG1. In GG2, there are two agents, one goal, 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, but there are no probabilistic transitions, and the reward structure differs: if both agents enter the goal from the side, they both earn 120; but, if one agent enters the goal through the center, while the other enters through the side, the former earns 125, while the latter earns only 100.

In all three of these grid games there exist *deterministic* (Nash and correlated) equilibrium policies for both agents. In GG1, there are several pairs of deterministic equilibrium policies in which the agents coordinate their behavior all of which yield equivalent rewards. In GG2 (and GG3), there are exactly two deterministic equilibrium policies: one agent moves up the center and the other moves up the side, and the same again with the agents' roles reversed. Note that these equilibria are asymmetric: the agent that moves up the center scores 100, but the agent that attempts to move through the barrier scores only 50 on average.

In addition, all the grid games exhibit *nondeterministic* correlated (and Nash) equilibrium policies. In GG2, there exists a continuum of symmetric, nondeterministic, 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 exist symmetric, nondeterministic, correlated equilibrium policies in which both agents move up the sides with high probability and each of the deterministic equilibria is played with equally low probability.

Our experiments reveal that off-policy correlated-Q, foe-Q, friend-Q ($\alpha \to 0.001$ and $\gamma = 0.9$.) and on-policy Q-learning (*i.e.*, ϵ-greedy, with $\epsilon \to 0.001$, $\alpha \to 0.001$, and $\gamma = 0.9$) all converge empirically in the three grid games. Littman [11] proves that FF-Q converges in general-sum Markov games. Fig. 2 shows that in fact ordinary Q-learning (on-policy) and correlated-Q (off-policy) also converge in these games.

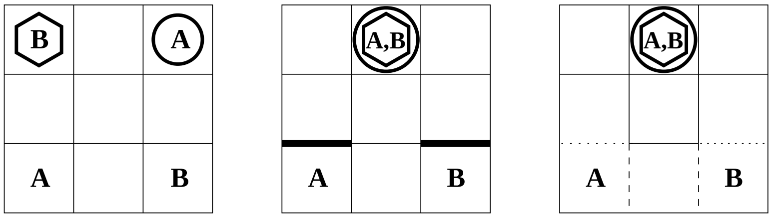
The values plotted in Fig. 2 are computed as follows. The error ERR_i^t at time t for agent i is the difference between $Q(s,\vec{a})$ at time t and $Q(s,\vec{a})$ at time $t-1$: $\text{ERR}_i^t = \left|Q_i^t(s,\vec{a}) - Q_i^{t-1}(s,\vec{a})\right|$. The values on the x-axis represent time, and the corresponding y-values are the means of the distributions of the errors ERR_i^t for all $t = 0,\ldots,x$. This mean is converging to zero for all algorithms in all grid games.[1]

4.1. Equilibrium Policies

Since all the learning algorithms converge, the following question arises: to what equilibrium policies do these algorithms converge? Since the state space in the grid games is large, rather than enumerate the equilibrium policies, we addressed this question by appending to the training phase an auxiliary testing phase in which the agents play according to the learned policies. Our results are depicted in Table 2.

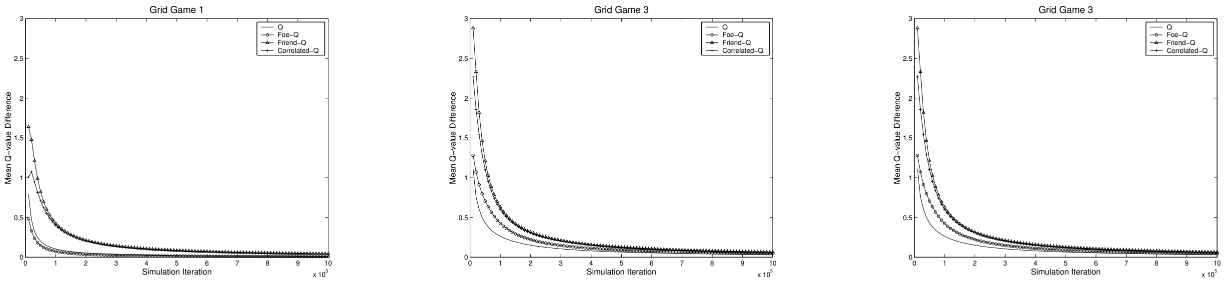
On-policy Q-learning is successful in grid games: it consistently converges to equilibrium policies in which the two agents coordinate their behavior perfectly. In GG1, this leads to symmetric scores, but in GG2 and GG3 their policies and their scores are asymmetric.

[1] In fact, the actual Q-value differences are converging at all state-action pairs. The mean is simply a useful statistic for summarizing this aggregate behavior.



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

Figure 1. Grid games. Initial States. Shapes indicate goals.



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

Figure 2. Convergence in the grid games: all algorithms are converging. The CE-Q algorithm shown is uCE-Q.

Grid Games	GG1		GG2		GG3	
Algorithm	Score	Games	Score	Games	Score	Games
Q	100,100	2500	49,100	3333	100,125	3333
Foe-Q	0,0	0	67,68	3003	120,120	3333
Friend-Q	$-10^4, -10^4$	0	$-10^4, -10^4$	0	$-10^4, -10^4$	0
uCE-Q	100,100	2500	50,100	3333	116,116	3333
eCE-Q	100,100	2500	51,100	3333	117,117	3333
rCE-Q	100,100	2500	100,49	3333	125,100	3333
lCE-Q	100,100	2500	100,51	3333	$-10^4, -10^4$	0

Table 2. Grid Games played repeatedly, allowing 10^4 moves. Average scores are shown. The number of games played varied with the agents' policies: some move directly to the goal, while others digress.

Foe-Q learners perform poorly in GG1. Rather than progress toward the goal, they cower in the corners, avoiding collisions, and avoiding the goal. 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, 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 moves 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 is guaranteed to converge to equilibrium values. Indeed, friend-Q learns Q-values that support equilibrium policies, but friends lack the ability to coordinate their play. Whenever one friend chooses a policy that collides with the policy of its so-called friend, both agents obtain negative scores in one never-ending game. In GG2 and GG3, friend-Q's performance is always poor: both friends learn to play the equilibrium policy that uses the center passage, which causes friends to collide repeatedly.

4.2. CE-Q Learning

In GG1, uCE-Q, eCE-Q, and rCE-Q all learn Q-values that coincide exactly with those of friend-Q: *i.e.*, Q-values that support equilibrium policies. But unlike friend-Q, these variants of CE-Q always obtain positive scores. In our implementation of CE-Q learning, a centralized mechanism computes a correlated equilibrium. Thus, CE-Q play is always coordinated, and uCE-Q, eCE-Q, and rCE-Q learners do not collide while playing the grid games. Were we to implement a decentralized version of CE-Q, such learners could fail to coordinate and earn negative scores.

The libertarian operator is one way to eliminate CE-Q's dependence on a centralized mechanism. In lCE-Q, each agent solves an independent optimization problem during learning; thus, play is not necessarily coordinated. Like the other variants of CE-Q, lCE-Q converges, and its Q-values coincide exactly with those of friend-Q in GG1. Also like the other variants of CE-Q, but unlike friend-Q, lCE-Q achieved positive scores in GG1. In fact, lCE-Q learners are indifferent between multiple equilibrium policies, but in this test run both agents happened upon coordinated equilibrium policies.

In GG2, all variants of CE-Q learning converge to policies much like ordinary Q-learners. Interestingly, taking long-run rewards into account, this game does not retain its Battle of the Sexes-like structure. On the contrary, GG2 is a dominance-solvable game. The Q-table below depicts the Q-values in the initial state that were learned by uCE-Q. (The other algorithms learned similar, although possibly transposed, values.) The column player eliminates the strategy SIDE, since it is dominated, after which the row player eliminates the strategy CENTER. Thus, the equilibrium outcome is (SIDE, CENTER), as the scores indicate.

	SIDE	CENTER
SIDE	48.70, 58.76	38.66, 81.00
CENTER	81.00, 40.30	34.42, 71.90

In both GG1 and GG2, CE-Q learning is indifferent between all correlated equilibrium policies, deterministic and nondeterministic, since they all yield equal sums of rewards. In GG3, however, uCE-Q (and eCE-Q) learn the particular nondeterministic correlated equilibrium policies that yield symmetric scores, because the sum (and the minimum) of rewards at this equilibrium exceeds that of any deterministic equilibrium policy. Consequently, the sum of the scores of uCE-Q and eCE-Q exceed that of Q-learning. CE-Q's rewards do not exceed the sum of the foe-Q learners' scores, however; but foe-Q learners do not behave rationally. In contrast, rCE-Q converges to a pure strategy equilibrium policy that is among those policies that maximize the maximum of all agents' rewards. Finally, each lCE-Q agent attempts to play the equilibrium policy that maximizes its own rewards, but this yields repeated collisions and negative scores.

Like Nash-Q, correlated-Q learning generalizes friend-Q, since it converges to precisely the same Q-values as friend-Q in games where friend-Q learns equilibrium values. In the next section, we show that again like Nash-Q, correlated-Q learning also generalizes foe-Q.

5. Soccer Game

The grid games are general-sum games for which there exist deterministic equilibria. In this section, we consider a simplified version of the soccer game that appears in Littman [10]; this game is zero-sum and there do not exist deterministic equilibrium policies.

The soccer field is a grid. The circle represents the ball. There are two players, whose possible actions are N, S, E, W, and stick. The players' actions are executed in random order. If this sequence of actions causes the players to collide, then only the first moves. But if the player with the ball moves *second*, then the ball changes possession.[2] 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.

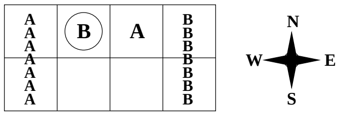
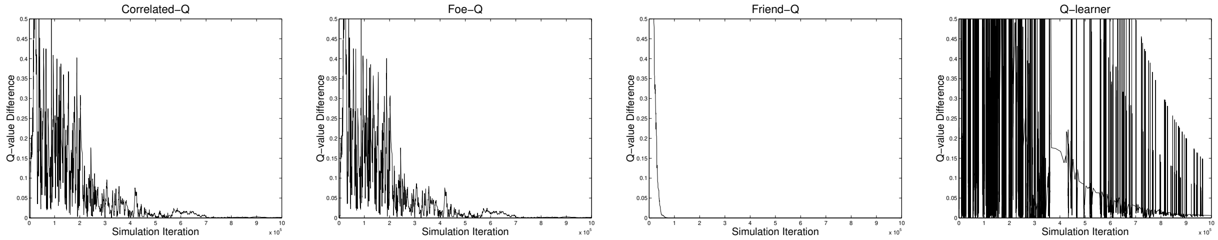


Figure 4. Soccer Game. State s.

In this simple soccer game, there do not exist deterministic equilibrium policies, since at some states there do not exist deterministic equilibria. For example, at the state depicted in Fig. 4 (hereafter, state s), any deterministic policy for player B is subject to indefinite blocking by player A. But if player B employs a nondeterministic policy, then player B can hope to pass player A on his next move.

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

[2]In other words, if the player without the ball moves into the player with the ball, attempting to steal the ball, he cannot. But if the player with the ball moves into the player without the ball, the former loses the ball to the latter. This form of the game is due to Littman [10].



(a) Correlated-Q (b) Foe-Q (c) Friend-Q (d) Q-learning

Figure 3. Convergence in the soccer game. All algorithms—except Q-learning—converge. As above, the CE-Q algorithm shown is uCE-Q.

Fig. 3 presents an example of a state-action pair at which Q-learning does not converge. The values on the x-axis represent time, and the corresponding y-values are the error terms $\text{ERR}_i^t = \left|Q_i^t(s,\vec{a}) - Q_i^{t-1}(s,\vec{a})\right|$. The error values shown in Figs. 3(a), (b), and (c) reflect player A's Q-values corresponding to state s, with player A taking action S and player B sticking. These three graphs depict converging sequences of error values for uCE-Q, foe-Q, and friend-Q, respectively,

Q-learners compute Q-values for each of their own possible actions, ignoring their opponents' actions. The error values shown in Fig. 3(d) reflect player A's Q-values, corresponding to state s and action S. In this figure, although the Q-value differences are decreasing, they are not converging. They are decreasing only because the learning rate $\alpha \to 0.001$. At all times, the amplitude of the oscillations in error values is as great as the envelope of the learning rate.

At state s, CE-Q and foe-Q converge to nondeterministic policies for both players, where each one randomizes between sticking and heading south.

Friend-Q, however, converges to a deterministic policy for player B at state s, namely E. Learning according to friend-Q, player B (fallaciously) anticipates the following sequence of events: player A sticks at state s, and player B takes action E. Thus, by taking action E, player B passes the ball to player A, with the intent that player A score for him. Player A is indifferent among her actions, since she assumes player B plans to score a goal for her immediately.

In this soccer 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. Friend-Q converges but its policies are irrational. Correlated-Q learning, however, converges to the same solution as foe-Q learning—the Q-values learned by the two algorithms are identical. Thus, CE-Q learns minimax equilibrium policies in this two-player, zero-sum game.

6. Discussion

In MDPs, single agent Q-learning has remarkable properties of global convergence: it provably converges to an optimal policy, from any initial condition. Similarly, if one applies multiagent Q-learning to a two-player constant-sum Markov game, it continues to yield global convergence to the unique equilibrium value of the game [12]. However, beyond this class of games, the strong convergence properties of Q-learning (either single agent or multiagent) cease to hold. Although Hu and Wellman [8] have identified sufficient conditions for convergence of Nash-Q learning, their theorems are of limited applicability because the conditions are extremely demanding. Furthermore, generalizing the solution concept from Nash equilibrium to allow for correlation does not necessarily help in this respect. Although this paper, like Hu and Wellman's work, provides some empirical evidence for convergence, we, too, offer no general result.

These difficulties with Q-learning in general-sum Markov games are to be expected because there are often multiple equilibria in such games, which renders the Q-learning dynamics non-ergodic. Thus, Q-learning (empirically) converges to Q-values (and an equilibrium policy) that depend on initial conditions. If we add noise to the Q-learning system so that at each iteration the agents play an equilibrium with high probability, but with low probability they choose their actions arbitrarily, we arrive at the notion of stochastic stability [3]. The system with noise is ergodic: it has a unique stationary distribution, which gives a precise estimate of the proportion of time that the system spends at each collection of Q-values (and at each equilibrium policy) in the long run. An equilibrium is stochastically stable if Q-learning with noise picks it up with positive probability in the limit, as the amount of noise go to zero. In future work, we intend to characterize the stochastically stable states of this ergodic Q-learning system.

7. Conclusion

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

In this paper, we discussed algorithms for learning Q-values in Markov games, given a game-theoretic solution concept. In past work, we have studied algorithms for learning game-theoretic equilibria in repeated games [7]. In ongoing work, we are combining these two types of learning. Specifically, we are replacing the linear programming call in CE-Q learning with an adaptive procedure that converges to correlated equilibrium [2, 6]. Similarly, we are studying an adaptive version of minimax-Q that replaces its linear programming call with an adaptive procedure that converges to minimax equilibrium [4]. (No adaptive algorithm is known to converge to Nash equilibrium.) This adaptive approach could lead to decentralized algorithms that learn correlated equilibrium policies in Markov games. The results contained in the present paper serve as the foundation for ongoing research, in which agents simultaneously learn Q-values and game-theoretic equilibria.

References

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

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

[3] D. Foster and P. Young. Stochastic evolutionary game dynamics. *Theoretical Population Biology*, 38:219–232, 1990.

[4] 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.

[5] A. Greenwald. Equilibrium policies in Markov games. Working Paper, August 2002.

[6] A. Greenwald, D. Gondek, and A. Jafari. On no-regret learning and game-theoretic equilibria. Draft, November 2001.

[7] A. Greenwald, A. Jafari, G. Ercal, and D. Gondek. On no-regret learning, Nash equilibrium, and fictitious play. In *Proceedings of Eighteenth International Conference on Machine Learning*, pages 226–233, June 2001.

[8] J. Hu and M. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 242–250, July 1998.

[9] J. Hu and M. Wellman. Experimental results of multiagent reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 407–414, July 2000.

[10] 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.

[11] 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.

[12] 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.

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

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