Predicting and Preventing Coordination Problems in Cooperative Q-learning Systems

Nancy Fulda and **Dan Ventura**
Computer Science Department
Brigham Young University
Provo, UT 84602
fulda@byu.edu, ventura@cs.byu.edu

Abstract

We present a conceptual framework for creating Q-learning-based algorithms that converge to optimal equilibria in cooperative multiagent settings. This framework includes a set of conditions that are sufficient to guarantee optimal system performance. We demonstrate the efficacy of the framework by using it to analyze several well-known multi-agent learning algorithms and conclude by employing it as a design tool to construct a simple, novel multi-agent learning algorithm.

1 Introduction

Multiagent reinforcement learning systems are interesting because they share many benefits of distributed artificial intelligence, including parallel execution, increased autonomy, and simplicity of individual agent design [Stone and Veloso, 2000; Cao *et al.*, 1997]. Q-learning [Watkins, 1989] is a natural choice for studying such systems because of its simplicity and its convergence guarantees. Also, because the Q-learning algorithm is itself so well-understood, researchers are able to focus on the unique challenges of learning in a multiagent environment.

Coaxing useful behavior out of a group of concurrently-learning Q-learners is not a trivial task. Because the agents are constantly modifying their behavior during the learning process, each agent is faced with an unpredictable environment which may invalidate convergence guarantees. Even when the agents converge to individually optimal policies, the combination of these policies may not be an optimal system behavior.

Poor group behavior is not the universal rule, of course. Many researchers have observed emergent coordination in groups of independent learners using Q-learning or similar algorithms [Maes and Brooks, 1990; Schaerf *et al.*, 1995; Sen *et al.*, 1994]. However, failed coordination attempts occur frequently enough to motivate a host of Q-learning adaptations for cooperative multiagent environments. [Claus and Boutilier, 1998; Lauer and Riedmiller, 2000; Littman, 2001; Wang and Sandholm, 2002]. These algorithms are critical steps towards a better understanding of multiagent Q-learning and of multiagent reinforcement learning in general. However, most of these algorithms become intractable as the number of agents in the system increases. Some of them rely on global perceptions of other agents' actions or require a unique optimal equilibrium, conditions that do not always exist in real-world systems. As reinforcement learning and Q-learning are applied to real-world problems with real-world constraints, new algorithms will need to be designed.

The objective of this paper is to understand why the algorithms cited above are able to work effectively, and to use this understanding to facilitate the development of algorithms that improve on this success. We do this by isolating three factors that can cause a system to behave poorly: suboptimal individual convergence, action shadowing, and the equilibrium selection problem. We prove that the absence of these three factors is sufficient to guarantee optimal behavior in cooperative Q-learning systems. Hence, any algorithm that effectively addresses all three factors will perform well, and system designers can select means of addressing each problem that are consistent with the constraints of their system.

2 Background and Terminology

The simplicity of the Q-learning algorithm [Watkins, 1989] has led to its frequent use in reinforcement learning research and permits a clear, concise study of multiagent coordination problems in simple environments. Multiagent coordination problems are not a direct consequence of the Q-learning algorithm, however. Thus, although we focus on Q-learning in this paper, the analysis presented should be applicable in other reinforcement learning paradigms as well.

A Q-learning agent may be described as a mapping from a state space S to an action space A. The agent maintains a list of expected discounted rewards, called Q-values, which are represented by the function $Q(s, a)$ where $s \in S$ and $a \in A$ are the current state and chosen action. The agent's objective is to learn an *optimal policy* $\pi^* : S \rightarrow A$ that maximizes expected discounted reward over all states s. At each time step, the agent chooses an action $a_t \in A$, receives a reward $r(s_t, a_t)$, and updates the appropriate Q-value as

$$\Delta Q(s_t, a_t) = \alpha[r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

where $0 < \alpha \leq 1$ is the learning rate and $0 \leq \gamma < 1$ is the discount factor. At any point in time, the agent's best estimate of the optimal policy is its *learned policy*

$\hat{\pi}(s) = \text{argmax}_a\{Q(s,a)\}$. The learned policy may differ from the optimal policy because it is based on Q-value estimates. In Q-learning, the learned policy also differs from the exploration policy executed during learning (*off-policy* learning). Note that other approaches might employ identical learned and exploration policies (*on-policy* learning).

Under certain conditions [Tsitsiklis, 1994], Q-learning will converge to a set of optimal Q-values

$$Q^*(s,a) = r(s,a) + \sum_t \sum_{s_t} \gamma^t p(s_t|s_{t-1}, a_{t-1}) r(s_t, \pi^*(s_t))$$

where $t = \{1, 2, ...\infty\}$ and $p(s_t|s_{t-1}, a_{t-1})$ is the probability of transitioning to state s_t given the previous state and action. If the agent has converged to an optimal Q-value set, then the optimal and learned policies are identical: $\pi^*(s) = \hat{\pi}(s) = \text{argmax}_a\{Q(s,a)\}$.

2.1 Q-learning in Multiagent Systems

Let S_i and A_i represent the state and action space of the ith agent in an n-agent system. The state of the system can be expressed as a vector of individual agent states $\mathbf{s} = [s_1, ..., s_n]$, $s_i \in S_i$, and the combination of the agents' individual actions is $\mathbf{a} = [a_1, ..., a_n]$, $a_i \in A_i$, resulting in joint state and action spaces, $\mathbf{S}$ and $\mathbf{A}$.

The combined policy for the system is a mapping $\Pi : \mathbf{S} \rightarrow \mathbf{A}$, where $\Pi(\mathbf{s}_t) = [\pi_1(s_{1(t)}), ..., \pi_n(s_{n(t)})]$. In each time step, a joint action $\mathbf{a} = \Pi(\mathbf{s}_t)$ is executed, and each agent receives a reward $r_i(s_i, \mathbf{a})$ and updates the corresponding Q-value $Q_i(s_i, a_i)$. Note that the rewards the agents receive are based on the *joint* action (and individual state), but the agents update Q-values for their corresponding *individual* action (and state). Thus, multiple joint actions are aliased to a single action perception of the agent.

It is sometimes useful to describe system behavior in terms of which joint policy an agent would prefer if it maintained a separate Q-value for each joint action. This preferred joint policy is a mapping $\Pi_i^* : S_i \rightarrow \mathbf{A}$ and represents the joint policy which provides maximum discounted reward for agent i over all s_i. An agent's *joint Q-values* $\mathbf{Q}_i(s_i, \mathbf{a})$ can then be defined as the average (over all joint states that contain s_i) expected reward received by agent i when joint action $\mathbf{a}$ is executed in s_i and Π_i^* is followed thereafter. The relationship between the joint Q-values and the agent's actual Q-value set $Q_i(s_i, a_i)$ is described by

$$Q_i(s_i, a_i) = \sum_{\mathbf{a}} p(\mathbf{a}|a_i) \mathbf{Q}_i(s_i, \mathbf{a})$$

where $p(\mathbf{a}|a_i)$ is the probability of joint action $\mathbf{a}$ being executed when agent i selects individual action a_i. Note that the probability is conditional because for some $\mathbf{a}$ (those that do not contain a_i) the probability is 0 but for others it is not. For those $\mathbf{a}$ that do contain a_i, the probability depends on the actions of the other agents (in other words, $p(\mathbf{a}|a_i)$ is a function of the joint exploration policy $\Pi(\mathbf{s}, t)$). The agent's optimal joint Q-values are defined as

$$\mathbf{Q}_i^*(s_i, \mathbf{a}) = r_i(s_i, \mathbf{a}) + \sum_t \sum_{s_{i(t)}} \gamma^t p(s_{i(t)}|s_{i(t-1)}, \mathbf{a}_{t-1}) r_i(s_{i(t)}, \Pi_i^*(s_{i(t)})))$$

where $t = \{1, 2, ...\infty\}$ and $p(s_{i(t)}|s_{i(t-1)}, \mathbf{a}_{t-1})$ is the probability of transitioning to individual state $s_{i(t)}$ given the previous state and action. An agent's preferred joint policy can be described in terms of the optimal joint Q-values, $\Pi_i^*(s_i) = \text{argmax}_{\mathbf{a}}\{\mathbf{Q}_i^*(s_i, \mathbf{a})\}$.

Definition 1. *A system of Q-learning agents is* cooperative *if and only if any joint action that maximizes $Q_i^*(s_i, a_i)$ for one agent also maximizes it for all agents.*

This definition of cooperation does not require that the agents share the same reward signal for all joint actions; it requires only that the agents share a set of mutually-preferred joint actions. It thus allows for scenarios where the agents' preferences differ, but where the agents' needs can (and must) be simultaneously satisfied.

We consider a solution to be *system optimal* if it is both a Nash equilibrium and Pareto-optimal. In a cooperative setting, this definition of optimality can be restricted to a subset of Pareto-optimal Nash equilibria called *coordination equilibria*. A coordination equilibrium is *strict* if the inequality in Definition 2 is strict (the inequality will be strict if there is exactly one joint action that results in all agents receiving their max reward; in other cases there are multiple such joint actions and some sort of coordinated equilibrium selection becomes necessary).

Definition 2. *Joint action $\mathbf{a}^*$ is a* coordination equilibrium *for state $\mathbf{s}$ if and only if $\forall\{\mathbf{a}, i \mid a_i \in \mathbf{a}$ and $a_i^* \in \mathbf{a}^*\}, Q_i^*(s_i, a_i^*) \geq Q_i^*(s_i, a_i)$.*

3 Factors That Can Cause Poor System Behavior

We now identify three factors that can cause poor system behavior. The first and third factors are represented in various guises in the literature. The second factor, action shadowing, is likely to be less familiar to readers. In the next section, we will show that the absence of these three factors is sufficient to guarantee that the system will perform optimally.

3.1 Poor Individual Behavior

An agent has learned an *individually optimal* policy $\pi_i^*(s_i) = \text{argmax}_{a_i}\{Q_i^*(s_i, a_i)\}$ if its behavior is a best response to the strategies of the other players. Q-learning is guaranteed to converge to an optimal Q-value set, and hence to an optimal policy, in the individual case. However, this convergence guarantee breaks down in multiagent systems because the changing behavior of the other agents creates a non-Markovian environment.

Despite this loss of theoretical guarantees, Q-learning agents often converge to optimal policies in multiagent settings because (1) the agents do not necessarily need to converge to an optimal Q-value set in order to execute an optimal policy and (2) if all agents are playing optimally, they must

settle to a Nash equilibrium, and Nash equilibria tend to be self-reinforcing. Individual behavior of Q-learning agents is well-studied in the literature and in what follows we focus particularly on the following two potential problems.

3.2 Action Shadowing

Action shadowing occurs when one individual action appears better than another, even though the second individual action is potentially superior. This can occur because typical Q-learning agents maintain Q-values only for individual actions, but receive rewards based on the joint action executed by the system. As a consequence, the agent's optimal policy may preclude the possibility of executing a coordination equilibrium.

Definition 3. *A joint action* $\mathbf{a}^\dagger$ *is* shadowed *by individual action* $\hat{a}_i$ *in state* $\mathbf{s}$ *if and only if* $\hat{a}_i = \pi_i^*(s_i)$ *and* $\forall \mathbf{a} | \hat{a}_i \in \mathbf{a}, \mathbf{Q}_i^*(s_i, \mathbf{a}^\dagger) > \mathbf{Q}_i^*(s_i, \mathbf{a})$.

A special case is *maximal action shadowing*, which occurs when the shadowed joint action provides maximal possible reward for the affected agent:

Definition 4. *An agent* i *experiences* maximal action shadowing *in state* $\mathbf{s}$ *if and only if there exist* $\mathbf{a}^*$ *and* $\hat{a}_i$ *such that* $\mathbf{a}^*$ *is shadowed by* $\hat{a}_i$ *and* $\forall \mathbf{a} \in \mathbf{A}$, $\mathbf{Q}_i^*(s_i, \mathbf{a}^*) \geq \mathbf{Q}_i^*(s_i, \mathbf{a})$

The cause of the action shadowing problem lies in the Q-value update function for agents in a multiagent system. In each time step, each agent receives a reward $r_i(s_i, \mathbf{a})$ based on the *joint* action space, but it updates the Q-value $Q_i(s_i, a_i)$, based on its *individual* action selection. Consequently, agent i is unable to distinguish between distinct rewards that are all aliased to the same individual action. Action shadowing is a consequence of the well-known credit assignment problem but is more precisely defined (and thus addressable).

Action shadowing is sometimes prevented by on-policy learning – agents seeking to maximize individual reward will tend to gravitate toward coordination equilibria. On-policy learning does not assure that an action shadowing problem will not occur, however. Maximal action shadowing is likely to occur despite on-policy learning in situations where failed coordination attempts are punished, as in penalty games [Claus and Boutilier, 1998]. Maximal action shadowing will always cause a coordination problem when agent interests do not conflict.

3.3 Equilibrium Selection Problems

An *equilibrium selection problem* occurs whenever coordination between at least two agents is required in selecting between multiple system optimal solutions. This is a somewhat stricter definition than the standard game-theoretic term, which refers to the task of selecting an optimal equilibrium from a set of (possibly suboptimal) potential equilibria.

Definition 5. *An* equilibrium selection problem *occurs whenever* $\exists(\mathbf{a}^1, \mathbf{a}^2 \neq \mathbf{a}^1) | \forall(i, j \in \{1, 2\}, \mathbf{a})$ $\mathbf{Q}_i^*(s_i, \mathbf{a}^j) \geq \mathbf{Q}_i^*(s_i, \mathbf{a})$.

The existence of an equilibrium selection problem does not necessarily result in suboptimal system behavior. However, an equilibrium selection problem creates a potential that the agents will mis-coordinate. Whether or not this happens will depend on the complete reward structure of the task as well as on the exploration strategy used by the system. Partially exploitive exploration strategies have proven particularly effective at encouraging convergence to a set of mutually compatible individual policies [Claus and Boutilier, 1998; Sen and Sekaran, 1998].

4 Achieving Optimal Performance

Cooperating Q-learners will behave optimally if each agent learns an individually optimal policy and if maximal action shadowing and the equilibrium selection problem are absent.

Theorem 1. *For any cooperative Q-learning system,* $\cup_{i=1}^n \hat{\pi}_i(s_i)$ *is a system optimal solution if the following conditions hold:*

(1) $\forall i, \hat{\pi}_i(s_i) = \pi_i^*(s_i)$

(2) $\nexists(\mathbf{a}^\dagger, \hat{a}_i) | \hat{a}_i = \pi_i^*(s_i)$ *and* $\forall \mathbf{a} | \hat{a}_i \in \mathbf{a}$, $\mathbf{Q}_i^*(s_i, \mathbf{a}^\dagger) > \mathbf{Q}_i^*(s_i, \mathbf{a})$

(3) $\nexists(\mathbf{a}^1, \mathbf{a}^2 \neq \mathbf{a}^1) | \forall(i, j \in \{1, 2\}, \mathbf{a}) \mathbf{Q}_i^*(s_i, \mathbf{a}^j) \geq \mathbf{Q}_i^*(s_i, \mathbf{a})$

Proof. Let $\hat{\mathbf{a}} = \cup_{i=1}^n \hat{\pi}_i(s_i)$ be the joint action selected by the learned joint policy of the system. Then $\forall i, \hat{a}_i = \hat{\pi}_i(s_i)$ and by Condition (1) $\forall i, \hat{a}_i = \pi_i^*(s_i)$.

We know from Condition (2) that there cannot be a joint action $\mathbf{a}^\dagger$ such that $\forall \mathbf{a} | \hat{a}_i \in \mathbf{a}, \mathbf{Q}_i^*(s_i, \mathbf{a}^\dagger) > \mathbf{Q}_i^*(s_i, \mathbf{a})$. This implies that $\hat{a}_i$ enables one or more joint actions that maximize agent i's joint Q-value function: $\exists\{\mathbf{a}^1, ..., \mathbf{a}^m\} | \forall(j \in \{1, ..., m\}, \mathbf{a}), \hat{a}_i \in \mathbf{a}^j$ and $\mathbf{Q}_i^*(s_i, \mathbf{a}^j) \geq \mathbf{Q}_i^*(s_i, \mathbf{a})$.

Because the system is cooperative, any joint action that maximizes expected discounted reward for one agent must maximize it for all other agents as well. Hence, we have a set of joint actions $\{\mathbf{a}^1, ..., \mathbf{a}^m\}$ such that $\forall(i, j \in \{1, ..., m\}, \mathbf{a}), \mathbf{Q}_i^*(s_i, \mathbf{a}^j) \geq \mathbf{Q}_i^*(s_i, \mathbf{a})$.

From Condition (3) we know that there can be at most one joint action that maximizes the expected discounted reward for all agents. It follows that $m = 1$ and there is a unique joint action $\mathbf{a}^1$ such that $\forall(i, \mathbf{a}), \mathbf{Q}_i^*(s_i, \mathbf{a}^1) \geq \mathbf{Q}_i^*(s_i, \mathbf{a})$.

Since each agent's individual action $\hat{a}_i$ enables a joint action that maximizes its expected discounted reward, it must be the case that $\hat{\mathbf{a}} = \mathbf{a}^1$. Because it maximizes expected discounted reward for every agent, $\hat{\mathbf{a}}$ is a (strict) coordination equilibrium (by Definition 2) and hence must be a system optimal solution. □

Naturally, these are not the only possible sufficient conditions to guarantee optimal system behavior. However, the conditions that make up our framework are preferable over many other possibilities because they can be addressed by modifying the learning algorithm directly, without placing additional constraints on the cooperative learning environment.

5 Improving System Performance

Given the framework imposed by Theorem 1, we consider various approaches to preventing coordination problems. Topics are grouped according to two factors that affect

system behavior: action shadowing, and equilibrium selection. The third factor, suboptimal individual convergence, is quite prevalent in the Q-learning literature, and is too broad a topic to be examined here.

There are two basic ways to improve system performance: control the task or modify the learning algorithm. In task-oriented approaches, reward structures are constrained so that action shadowing and the equilibrium selection problem are not present for any agent. Algorithm-oriented approaches attempt to design algorithms that cope effectively with the above-mentioned problems. In general, algorithm-oriented approaches are superior because they enable the creation of general-purpose learners that learn effective policies regardless of the reward structure. Still, it is useful to be acquainted with task-oriented approaches because algorithms designed for constrained environments will not need to explicitly address issues that are implicitly resolved by the environment.

5.1 Task-oriented Approaches for Action Shadowing

Dominant Strategies: A dominant strategy is a policy that maximizes an agent's payoff regardless of the actions of the other agents. If the task is structured in a way that creates dominant strategies for all agents, no agent can experience action shadowing.

5.2 Algorithm-oriented Approaches for Action Shadowing

Joint Action Learning: If each agent is able to perceive the actions of its counterparts, then it will be able to distinguish between high and low payoffs received for different joint actions, rather than indiscriminately attributing the payoffs to a single individual action. This technique is often called *joint action learning* [Claus and Boutilier, 1998]. Other examples of joint action learning include Friend-or-Foe Q-learning [Littman, 2001], Nash Q-learning [Hu and Wellman, 2003], and cooperative learners [Tan, 1997].

Optimistic Updates and Optimistic Exploration: For deterministic environments, *distributed reinforcement learning* [Lauer and Riedmiller, 2000] has the same effect as joint action learning, but without giving the agents any extra information – agents optimistically assume that all other agents will act to maximize their reward, and thus store the maximum observed reward for each action as that action's utility. A variation for stochastic domains uses a weighted sum of the actual Q-value and an heuristic to select an action for execution [Kapetanakis and Kudenko, 2002] . This heuristic results in effective convergence to optimal equilibria in some stochastic climbing games, but does not do so in all stochastic environments.

Variable Learning Rate: Another approach to addressing the problem is to minimize the effect that the learning of other agents has on a given agent's own learning. This is the approach taken by WoLF variants [Bowling, 2004], in which a variable learning rate for updating the Q-values has the effect of holding some agents' policies constant while others learn against the (temporarily) stationary environment.

5.3 Task-oriented Approaches for Equilibrium Selection

Unique Optimal Solution: The simplest way to prevent the equilibrium selection problem is to design a system that has only a single optimal solution. When this is the case, the agents do not need to coordinate in selecting between multiple optimal equilibria. This is the premise behind the convergence proofs in [Hu and Wellman, 1998] and [Littman, 2001]. It is also the only possibility for avoiding an equilibrium selection problem using WoLF [Bowling, 2004] (note that the WoLF variants have focused on adversarial general sum games, and not at all on cooperative ones).

5.4 Algorithm-oriented Approaches for Equilibrium Selection

Emergent Coordination: Emergent Coordination describes the tendency of a set of non-communicating reinforcement learners to learn compatible policies because each agent is constantly seeking a best response to the other agents' actions. This has been demonstrated, for example, in hexapedal robot locomotion [Maes and Brooks, 1990], network load balancing [Schaerf *et al.*, 1995], and a cooperative box-pushing task [Sen *et al.*, 1994].

Social Conventions: A social convention is a pre-arranged constraint on behavior that applies to all agents, such as driving on the right side of the street. This is the premise behind social learners [Mataric, 1997] and homo egualis agents [Nowe *et al.*, 2001], and it has also been used as a coordination mechanism in Q-learning systems [Lauer and Riedmiller, 2000].

Strategic Learners: Strategic learners are agents that model their counterparts and select an optimal individual strategy based on that model. One commonly-used model in games where the agents can see each others' actions is *fictitious play* [Claus and Boutilier, 1998] and its variant, adaptive play [Wang and Sandholm, 2002; Young, 1993]. Another example is that of concurrent reinforcement learners [Mundhe and Sen, 2000].

6 Applying the Framework: Incremental Policy Learning

In this section we describe a simple learning algorithm designed by addressing each of the conditions of Theorem 1. This algorithm, called Incremental Policy Learning, addresses the issues of optimal individual convergence, action shadowing and the equilibrium selection problem. It consistently learns to play a coordination equilibrium in deterministic environments.

Achieving Optimal Individual Behavior: Incremental Policy Learning achieves optimal individual behavior by using a standard Q-learning update equation to estimate Q-values.

Preventing Action Shadowing: Following the example of [Claus and Boutilier, 1998], Incremental Policy Learning prevents action shadowing by learning Q-values over the entire joint action space. Each agent can perceive the action selections of its counterparts (but only its own reward signal) and uses this information to learn Q-values for all possible joint

actions. This enables the agents to clearly determine which individual actions may lead to coordination equilibria.

Addressing the Equilibrium Selection Problem: Incremental Policy Learning uses a sequence of incremental policy adjustments to select between multiple optimal equilibria. Each agent maintains a probability distribution $P = \{p(a_1), ..., p(a_m)\}$ over its available action set $A = \{a_1, ..., a_m\}$. These probabilities are initialized and modified according to the algorithm described below.

6.1 The Incremental Policy Learning Algorithm

- **Initialization**
  $\forall i, p(a_i) = v_i$, where v is arbitrarily chosen such that $\sum_{i=0}^{n} v_i = 1$ and $\forall i, v_i > 0$.
- **Action Selection**
  In each time step t, the agent selects an action $a(t) \in A$ according to probability distribution P. The agent executes this action, receives a reward signal $r(t)$, updates its joint Q-values, and updates P as described below.
- **Probability Updates**
  Let r_{max} be the maximum Q-value stored in the joint action table
  Let $0 < \alpha \leq 1$.
  If $r(t) \geq r_{max}$ then $\forall i$:

  if $(a(t) = a_i)$ then $p(a_i) = p(a_i) + \alpha(1 - p(a_i))$
  if $(a(t) \neq a_i)$ then $p(a_i) = p(a_i) - \alpha p(a_i)$

Here r_{max} is the reward for the target equilibrium (the reward for the preferred joint action). Intuitively, whenever r_{max} is received as a reward, the action selection probability distribution is skewed somewhat toward the action that resulted in its receipt.

A proof sketch that the IPL algorithm meets the criteria of Theorem 1 (and thus will result in optimal system performance) proceeds as follows. Condition 1 is met because the individual agents use standard Q-learning. Condition 2 is met because the agents are allowed to see the joint action space. An argument for meeting condition 3 is given in [Fulda and Ventura, 2004].

6.2 Results

We first allow the agents to select random actions until their joint Q-values converge, and only then use the coordination mechanism described above. This results in the agents consistently learning to play a coordination equilibrium. However, such strictly controlled situations are of limited interest.

We next experiment with two agents learning Q-values and the coordination policy simultaneously. These agents repeatedly play a (stateless) single-stage game in which each agent has five possible action selections. Each cell of the payoff matrix was randomly initialized to an integer between 0 and 24 (different random payoffs were assigned to each agent), with the exception of five randomly placed coordination equilibria whose payoff was 25 for both agents. The algorithm was tested in both deterministic and stochastic environments (each reward signal was summed with Gaussian noise).

Figure 1 shows the algorithm's performance as a function of α, averaged over 100 trials. Because the Q-values and the policy are learned simultaneously, the agents do not always achieve their maximum expected rewards. This occurs because the agents' policies sometimes settle before the Q-value estimates for the coordination equilibria are large enough to be distinguished from the Q-values of less desirable actions. As expected, the algorithm performs better with lower values of α. The smaller α is, the more likely it is that the joint Q-values will converge to their correct values before the agents' policies settle, which in turn enables the agents to easily learn a coordination equilibrium. Interestingly, even when α approaches 1, the performance of the algorithm degrades rather gracefully.

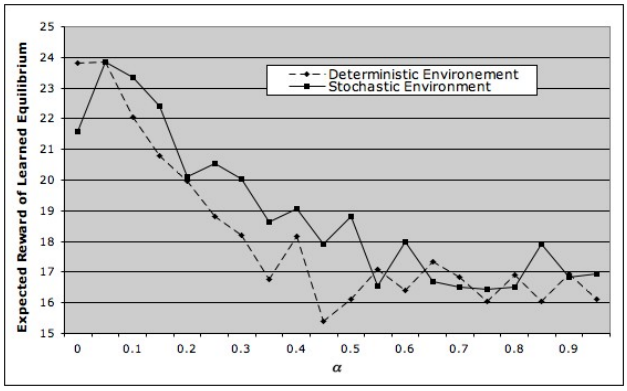

Figure 1: Incremental Policy Learning performance as a function of α

6.3 Discussion

The methods used by Incremental Policy Learning are simple, but the principle demonstrated is powerful. An algorithm that successfully achieves individual optimal performance, avoids maximal action shadowing, and addresses the equilibrium selection problem will learn an optimal group behavior in cooperative environments. Incremental Policy Learning satisfies these requirements in deterministic environments when α is sufficiently small. In fact, the algorithm performs well even when these requirements are violated.

Incremental Policy Learning is particularly suited to environments with small numbers of interacting agents. If the number of agents becomes very large, a method of addressing the action shadowing problem other than joint action learning would be required. A possible alternative is to represent only significant subsets of the joint action space, as in [Fulda and Ventura, 2003].

7 Conclusion

We have identified a set of conditions sufficient to guarantee optimal performance for systems of cooperative, concurrently learning agents. Each condition can be met in multiple different ways, thus enabling the creation of learning algorithms that are suited to the constraints of a particular environment or task. As an example, a learning algorithm has been presented that addresses each of the conditions.

The major advantage of our framework is that the conditions can all be satisfied through algorithm-oriented ap-

proaches. In contrast, many other conditions sufficient for optimal performance require additional constraints on the environment or the task (for example, iterated strict dominance games or the generic condition in which all solutions are coordination equilibria). Since our stated objective is to assist in creating algorithms that are uniquely adapted to the environment, we require conditions that can be addressed through the algorithm itself.

Future work will concentrate on generalizing the approach to competitive environments and eventually to environments of conflicting interest (such as battle of the sexes and the prisoner's dilemma). One approach to this replaces the Q-function with an evaluation function so that although agent preferences may differ, they will coordinate in seeking a compromise between those preferences, essentially converting an adversarial system into a cooperative one. This is the approach taken by Hu and Wellman's Nash Q-learning, in which the agents seek to play Nash equilibria rather than seeking to maximize rewards directly [Hu and Wellman, 2003].

Also, the framework presented here makes an underlying assumption of the independence of the individual states s_i. That is, it assumes that state s_i of agent i (at time t) will not affect the state s_j of agent j at some later time. It would be interesting to generalize this work to consider the case when this independence assumption does not hold.

References

[Bowling, 2004] Michael Bowling. Convergence and no-regret in multiagent learning. In *Neural Information Processing Systems*, pages 209–216, 2004.

[Cao *et al.*, 1997] Y. Cao, A. Fukunaga, A. Kahng, and F. Meng. Cooperative mobile robots: Antecedents and directions. *Autonomous Robots*, 4:1–23, 1997.

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

[Fulda and Ventura, 2003] Nancy Fulda and Dan Ventura. Dynamic joint action perception for q-learning agents. In *Proceedings of the International Conference on Machine Learning and Applications*, pages 73–78, 2003.

[Fulda and Ventura, 2004] Nancy Fulda and Dan Ventura. Incremental policy learning: An equilibrium selection algorithm for reinforcement learning agents with common interests. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1121–1126, 2004.

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

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

[Kapetanakis and Kudenko, 2002] S. Kapetanakis and D. Kudenko. Improving on the reinforcement learning of coordination in cooperative multi-agent systems. In *Second AISB Symposium on Adaptive Agents and Multi-Agent Systems*, 2002.

[Lauer and Riedmiller, 2000] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the International Conference on Machine Learning*, pages 535–542, 2000.

[Littman, 2001] Michael Littman. Friend or foe q-learning in general-sum markov games. In *Proceedings of the International Conference on Machine Learning*, pages 322–328, 2001.

[Maes and Brooks, 1990] Pattie Maes and Rodney A. Brooks. Learning to coordinate behaviors. In *Proceedings of the National Conference on Artificial Intelligence*, pages 796–802, 1990.

[Mataric, 1997] M. J. Mataric. Learning social behavior. *Robotics and Autonomous Systems*, 20:191–204, 1997.

[Mundhe and Sen, 2000] M. Mundhe and S. Sen. Evaluating concurrent reinforcement learners. In *Proceedings of the International Conference on Multiagent Systems*, pages 421–422, 2000.

[Nowe *et al.*, 2001] Ann Nowe, Katja Verbeeck, and Tom Lenaerts. Learning agents in a homo egualis society. In *Proceedings of Learning Agents Workshop, Agents 2001 Conference*, 2001.

[Schaerf *et al.*, 1995] Andrea Schaerf, Yoav Shoham, and Moshe Tennenholtz. Adaptive load balancing: A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2:475–500, 1995.

[Sen and Sekaran, 1998] Sandip Sen and Mahendra Sekaran. Individual learning of coordination knowledge. *JETAI*, 10(3):333–356, 1998.

[Sen *et al.*, 1994] Sandip Sen, Mahendra Sekaran, and John Hale. Learning to coordinate without sharing information. In *Proceedings of the National Conference on Artificial Intelligence*, pages 426–431, 1994.

[Stone and Veloso, 2000] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[Tan, 1997] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative learning. In *Readings in Agents*, pages 487–494, 1997.

[Tsitsiklis, 1994] John N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16:185–202, 1994.

[Wang and Sandholm, 2002] X. Wang and T. Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Neural Information Processing Systems*, pages 1571–1578, 2002.

[Watkins, 1989] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, 1989.

[Young, 1993] H. Peyton Young. The evolution of conventions. *Econometrica*, 61:57–84, 1993.