Fast Concurrent Reinforcement Learners

Bikramjit Banerjee & Sandip Sen
Math & Computer Sciences Department
University of Tulsa
Tulsa, OK 74104
$\{$*bikram, sandip*$\}$*@euler.mcs.utulsa.edu*

Jing Peng
Dept. of Computer Science
Oklahoma State University
Stillwater, OK 74078
jpeng@cs.okstate.edu

Abstract

When several agents learn concurrently, the payoff received by an agent is dependent on the behavior of the other agents. As the other agents learn, the reward of one agent becomes non-stationary. This makes learning in multiagent systems more difficult than single-agent learning. A few methods, however, are known to guarantee convergence to equilibrium in the limit in such systems. In this paper we experimentally study one such technique, the minimax-Q, in a competitive domain and prove its equivalence with another well-known method for competitive domains. We study the rate of convergence of minimax-Q and investigate possible ways for increasing the same. We also present a variant of the algorithm, minimax-SARSA, and prove its convergence to minimax-Q values under appropriate conditions. Finally we show that this new algorithm performs better than simple minimax-Q in a general-sum domain as well.

1 Introduction

The reinforcement learning (RL) paradigm provides techniques using which an individual agent can optimize environmental payoff. However, the presence of a non-stationary environment, central to multi-agent learning, violates the assumptions underlying convergence proofs of single agent RL techniques. As a result, standard reinforcement learning techniques like Q-learning are not guaranteed to converge in a multi-agent environment. The focus of convergence for multiple, concurrent learners is on an equilibrium strategy-profile rather than optimal strategies for the individual agents.

The stochastic-game (or *Markov Games*) framework, a generalization of Markov Decision Processes for multiple controllers, has been used to model learning by agents in purely competitive domains [Littman, 1994]. In that work, the author has presented a minimax-Q learning rule and evaluated its performance experimentally. The Q-learning rule, being a model-free and online learning technique, is particularly suited for multi-stage games, and as such, an attractive candidate for multiagent-learning in uncertain environments. Hu and Wellman (1998) extended Littman's framework to enable the agents to converge to a mixed-strategy Nash equilibrium. There have also been other research on using the stochastic-game framework in multiagent learning, such as an empirical study of multiagent Q-learning in semi-competitive domains [Sandholm and Crites, 1996] among others.

The first two techniques mentioned above are known to converge in the limit. In this paper we show that these two techniques are essentially identical in purely competitive domains. To increase the convergence rate of the minimax-Q rule, we extend it to incorporate SARSA [Rummery, 1994; Sutton and Burto, 1998] and Q(λ) [Peng and Williams, 1996] techniques, and study the convergence rates of these different methods in a competitive soccer domain [Littman, 1994].

2 Multiagent Q-learning

Definition 1 *A Markov Decision Process (MDP) is a quadruple $\{S, A, T, R\}$, where S is the set of states, A is the set of actions, T is the transition function, $T : S \times A \rightarrow PD(S)$, PD being a probability distribution, and R is the reward function, $R : S \times A \rightarrow \Re$.*

A multiagent reinforcement-learning task can be looked upon as an extended MDP, with S specifying the joint-state of the agents, A being the joint-actions of the agents, $A_1 \times A_2 \times \ldots A_n$ where A_i is the set of actions available to the ith agent, T as the joint-transition function, and the reward function is redefined as $R : S \times A \rightarrow \Re^n$. The functions T and R are usually unknown. The goal of the ith agent is to find a strategy π_i that maximizes its expected sum of discounted rewards, $v(s, \pi_i) = \sum_{t=0}^{\infty} \gamma^t E(r_t^i | \pi_i, \pi_{-i}, s_0 = s)$ where s_0 is the initial joint-state, r_t^i is the reward of the ith agent at time t, $\gamma \in [0, 1)$ is the discount factor, and π_{-i} is the strategy-profile of i's opponents.

Definition 2 *A bimatrix game is given by a pair of matrices, (M_1, M_2), (each of size $|A_1| \times |A_2|$ for a two-agent game) where the payoff of the ith agent for the joint action (a_1, a_2) is given by the entry $M_k(a_1, a_2)$, $\forall (a_1, a_2) \in A_1 \times A_2$, $k = 1, 2$.*

Each stage of an extended-MDP for two agents (it can be extended to n agents using n-dimensional tables instead of matrices) can be looked upon as a bimatrix game. A *zero-sum game* is a special bimatrix game where $M_1(a_1, a_2) + M_2(a_1, a_2) = 0$, $\forall (a_1, a_2) \in A_1 \times A_2$.

Definition 3 *A mixed-strategy Nash Equilibrium for a bimatrix game* (M_1, M_2) *is a pair of probability vectors* (π_1^*, π_2^*) *such that*

$$\pi_1^{*T} M_1 \pi_2^* \geq \pi_1^T M_1 \pi_2^* \;\; \forall \pi_1 \in PD(A_1).$$

$$\pi_1^{*T} M_2 \pi_2^* \geq \pi_1^{*T} M_2 \pi_2 \;\; \forall \pi_2 \in PD(A_2).$$

where $PD(A_i)$ *is the set of probability-distributions over the* ith *agent's action space.*

No player in this game has any incentive for unilateral deviation from the Nash equilibrium strategy, given the other's strategy. There always exists at least one such equilibrium profile for an arbitrary finite bimatrix game [Nash, 1951].

An individual learner may, but need not, use a model of the environment to learn the transition and reward functions. Q-learning is one example of model-free learning. In greedy policy Q-learning, an agent starts with arbitrary initial Q-values (or action values) for each state-action pair (s, a), and repeatedly chooses actions, noting its rewards and transitions and updating Q as

$$Q^{t+1}(s_t, a_t) = (1 - \alpha_t) Q^t(s_t, a_t) + \alpha_t [r_t + \gamma v^t(s_{t+1})]$$

where

$$v^t(s_{t+1}) = \max_a Q^t(s_{t+1}, a). \tag{1}$$

and $\alpha_t \in [0, 1)$ is the learning rate. Watkins and Dayan (1992) have proved that the above iteration converges to optimal action values under infinite sampling of each state-action pair and a particular schedule of the learning rate.

In the case of multiagent learning, the above iteration would not work, since the maximization over one's action is insufficient in the presence of multiple actors. However, if the reward-function of the opponent is negatively correlated, then actions can be selected by solving the bimatrix-game $(M(s), -M(s))$ greedily for the opponent, and pessimistically for oneself, to guarantee a minimum expected payoff. This produces Littman's minimax-Q algorithm for simultaneous-move zero-sum games, for which the value-function for agent 1 is

$$v_1^t(s_{t+1}) = \max_{\pi \in PD(A)} \min_{o \in O} \pi^T Q_1^t(s_{t+1}, ., o), \tag{2}$$

where A and O are the action-sets of the learning-agent (agent 1) and its opponent respectively, and $Q_1^t(s_{t+1}, ., o)$ is a vector of action values of the learner corresponding to its opponent's action o. The current policy $\pi(s)$ can be solved by linear-programming for the constrained, minimax optimization on $Q(s, ., .)$. The minimax-Q learning rule has been proved to converge to optimal action values [Szepesvári and Littman, 1997].

For general-sum games, however, the ith agent needs to know π_{-i}, in absence of which it has to model its opponents. In such games, each agent can observe the other agent's actions and rewards and maintains separate Q-tables for each of them in addition to its own [Hu and Wellman, 1998].

The value-function for agent 1 in this case is

$$v_1^t(s_{t+1}) = \pi_1^*(s_{t+1})^T Q_1^t(s_{t+1}, ., .) \pi_2^*(s_{t+1}), \tag{3}$$

where (π_1^*, π_2^*) are the Nash-strategies of the agents for the bimatrix game $\{Q_1^t(s_{t+1}, ., .), Q_2^t(s_{t+1}, ., .)\}$, which can be solved by quadratic-programming technique [Mangasarian and Stone, 1964]. In zero-sum games, the value-function in (3) simplifies to

$$v_1^t(s_{t+1}) = \max_{\pi_1 \in PD(A_1)} \min_{\pi_2 \in PD(A_2)} \pi_1^T Q_1^t(s_{t+1}, ., .) \pi_2. \tag{4}$$

This algorithm converges to a Nash equilibrium, for a restrictive class of Nash-equilibria [Hu and Wellman, 1998], that in addition to the constraints imposed by its definition, satisfies the following

$$\pi_1^{*T} Q_1^t(s_{t+1}) \pi_2^* \leq \pi_1^{*T} Q_1^t(s_{t+1}) \pi_2 \;, \forall \pi_2 \in PD(A_2),$$

$$\pi_1^{*T} Q_2^t(s_{t+1}) \pi_2^* \leq \pi_1^T Q_2^t(s_{t+1}) \pi_2^* \;, \forall \pi_1 \in PD(A_1).$$

Though this may not be true in general, it holds for zero-sum domains, where any deviation by the opponent from its equilibrium-strategy decreases its expected payoff, thus increasing the modeler's expected payoff. Hence, the convergence of this algorithm is guaranteed in zero-sum domains. Furthermore, the algorithms developed by Littman (1994) and Hu and Wellman (1998) (we call the latter Nash-Q) differ structurally only in the use of rules (2) and (4) respectively, in updating Q. But contrary to the statement made by Hu and Wellman (1998), these expressions are functionally equivalent in zero-sum games. Game theory [Thie, 1998] states that in a zero-sum game

$$\max_{\pi_1 \in PD(A_1)} \min_{\pi_2 \in PD(A_2)} \pi_1^T Q_1(s, ., .) \pi_2$$

$$= \max_{\pi_1 \in PD(A_1)} \min_{o \in A_2} \pi_1^T Q_1(s, ., o), \; \forall s \in S. \tag{5}$$

An informal argument may go as follows. Let $Q_1(s, ., .) = (q_1, q_2, \cdots, q_n)$, where q_i represents the ith column of $Q_1(s, ., .)$, and $n = |A_2|$. Then $\pi_1^T Q_1(s, ., .) = (\pi_1^T q_1, \pi_1^T q_2, \cdots, \pi_1^T q_n)$ for any $\pi_1 \in PD(A_1)$. When $\pi_1^T q$ is treated as a random variable from the distribution π_2, we have

$$\begin{aligned} \min_{\pi_2 \in PD(A_2)} \pi_1^T Q_1(s, ., .) \pi_2 &= \min_{\pi_2 \in PD(A_2)} E_{\pi_2}[\pi_1^T q] \\ &\geq \min_{o \in A_2} \pi_1^T Q_1(s, ., o). \end{aligned}$$

On the other hand, since any pure strategy is also a mixed strategy, we have

$$\begin{aligned} \min_{\pi_2 \in PD(A_2)} \pi_1^T Q_1(s, ., .) \pi_2 &= \min_{\pi_2 \in PD(A_2)} E_{\pi_2}[\pi_1^T q] \\ &\leq \min_{o \in A_2} \pi_1^T Q_1(s, ., o). \end{aligned}$$

Consequently we have the equality (5). Therefore we observe that both minimax-Q and Nash-Q compute identical policies and value-functions in zero-sum domains.

3 Expediting minimax-Q learning

Since minimax-Q and Nash-Q algorithms are equivalent in the purely competitive domains that we consider in this paper, we focus on the minimax-Q algorithm since it does not require maintenance of opponents' Q functions and hence is resource-efficient. We now turn to explore possible ways to speed-up the chosen algorithm.

3.1 Fast minimax-Q(λ)

Since Q-learning updates only the last state with the reinforcements, it is substantially slow in updating the action values. A well-known technique to speed-up single-agent Q-learning is to integrate it with TD(λ) reward-estimation scheme, producing the Q(λ)-learning rule [Peng and Williams, 1996]. For experimentation, we have used a faster version of the Peng-Williams' algorithm, where the Q updates are 'lazily' postponed until necessary [Wiering and Schmidhuber, 1998]. Q(λ) can be applied to each of the two learning schemes in the previous section, by defining $v(s_{t+1})$ in the Q(λ) algorithm by the equation in (2) or (3) as the case may be. The guarantee of convergence, however, may no longer hold.

3.2 Minimax-SARSA learning

The previous techniques were all off-policy learning rules, where the expected value of the next state is used to update the value of the current state. The SARSA(0) technique is, on the other hand, an on-policy learning rule that depends heavily on the actual learning policy followed [Rummery, 1994; Sutton and Burto, 1998]. In general, off-policy algorithms can separate control from exploration while on-policy reinforcement learning algorithms cannot. Despite this, on-policy algorithms with function approximation in single agent learning appear to be superior to off-policy algorithms in control as well as prediction problems [Boyan and Moore, 1995; Sutton, 1996; Tsitsiklis and Roy, 1997]. On-policy algorithms can learn to behave consistently with exploration [Sutton and Burto, 1998] Moreover, on-policy algorithms are more natural to combine with eligibility traces than off-policy algorithms are. This raises the following question that this research effort endeavors to answer: Can on-policy RL algorithms perform equally well in multiagent domains? In that case we can possibly achieve faster convergence of a hybrid of the SARSA technique and minimax-Q rule.

In a simple Q-learning scenario, the SARSA technique modifies the update rule (1) as $v^t(s_{t+1}) = Q^t(s_{t+1}, a_{t+1})$. Thus a SARSA rule learns the values of its own actions, and can converge to optimal values only if the learning policy chooses optimal actions in the limit. In a multiagent minimax-Q setting, the rule (2) would be replaced (for agent 1) by

$$v_1^t(s_{t+1}) = Q_1^t(s_{t+1}, a_{t+1}, o_{t+1}),$$

while the policy to choose actions would still be computed by the original minimax-Q rule. To achieve convergence of this rule to minimax-Q values, we follow an ϵ-minimax strategy that satisfies the need of infinite exploration while being minimax in the limit, i.e., ϵ decays to 0 in the limit. We call such exploration 'Minimax in the limit with infinite exploration' or MLIE. Our convergence result rests on the following lemma established by Singh *et al.* (2000).

Lemma 1 *Consider a stochastic process (α_t, Δ^t, F^t), $t \geq 0$, where $\alpha_t, \Delta^t, F^t : X \to \Re$ satisfy the equations*

$$\Delta^{t+1}(x) = (1 - \alpha_t(x))\Delta^t(x) + \alpha_t(x)F^t(x),$$

where $x \in X$, $t = 0, 1, 2, \ldots$. Let P_t be a sequence of increasing σ-fields such that α_0 and Δ^0 are P_0 measurable and α_t, Δ^t and F^{t-1} are P_t measurable, $t = 1, 2, \ldots$. Assume that the following hold:
1. X is finite.
2. $0 \leq \alpha_t(x) \leq 1$, $\sum_t \alpha_t(x) = \infty$, $\sum_t \alpha_t^2(x) < \infty$ w.p.1.
3. $\|E\{F^t(.)|P_t\}\|_W \leq \delta\|\Delta^t\|_W{}^1 + c_t$, where $\delta \in [0, 1)$ and c_t converges to 0 w.p.1.
4. $Var\{F^t(x)|P_t\} \leq \beta(1 + \|\Delta\|_W)^2$, for some constant β.
Then, Δ^t converges to 0 with probability 1 (w.p.1).

The update rule for SARSA for agent 1 say, is

$$Q_1^{t+1}(s_t, a_t, o_t) = (1 - \alpha_t)Q_1^t(s_t, a_t, o_t) + \alpha_t[r_t^1 + \gamma Q_1^t(s_{t+1}, a_{t+1}, o_{t+1})]. \quad (6)$$

We also note that the fixed point of the minimax-Q rule [Szepesvári and Littman, 1997] (for agent 1) is

$$Q_1^*(s_t, a_t, o_t) = R_1(s_t, a_t, o_t) + E_y[\max_{\pi_1} \min_o \pi_1^T Q_1^*(y, ., o)]. \quad (7)$$

Now we state and prove the theorem for convergence of minimax-SARSA learning using Lemma 1.

Theorem 1 *The learning rule specified in (6) converges to the values in equation (7) with probability 1 provided a_t is chosen using an MLIE scheme at each step t, the immediate rewards are bounded and have finite variance, the Q-values are stored in lookup tables, the learning rate, α_t, satisfies condition 2 in Lemma 1, and the opponent plays greedily in the limit.*

Proof: (Outline) Writing x in Lemma 1 as (s_t, a_t, o_t) and Δ^t as $Q_1^t(s_t, a_t, o_t) - Q_1^*(s_t, a_t, o_t)$, and defining $\alpha_t(s, a, o) = 0$ unless $(s, a, o) = (s_t, a_t, o_t)\ \forall t$, we have

$$\begin{aligned} F^t(s_t, a_t, o_t) = \; & r_t^1 + \gamma \max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o) \\ & - Q_1^*(s_t, a_t, o_t) + \gamma Q_1^t(s_{t+1}, a_{t+1}, o_{t+1}) \\ & - \gamma \max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o)], \end{aligned} \quad (8)$$

which gives rise to

$$F^t(s_t, a_t, o_t) = F_M^t(s_t, a_t, o_t) + \gamma[d_t(s_t, a_t, o_t)].$$

It can be shown that the measurability and variance conditions are satisfied and that

$$\|E\{F_M^t(., ., .)|P_t\}\| \leq \gamma_M \|\Delta^t\|$$

for some $\gamma_M \in [0, 1)$ (since minimax-Q operator is a contraction), according to the outline provided by Singh *et al.* (2000). The remaining task is to show that $\|E\{d_t(., ., .)|P_t\}\|$ vanishes in the limit, under MLIE exploration. We consider the following cases:

Case 1: $Q_1^t(s_{t+1}, a_{t+1}, o_{t+1}) \geq \max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o)$.
Since $\max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o) \geq \min_o Q_1^t(s_{t+1}, a_{t+1}, o)$, we have $d_t(s_t, a_t, o_t) = |d_t(s_t, a_t, o_t)| \leq Q_1^t(s_{t+1}, a_{t+1}, o_{t+1}) - \min_o Q_1^t(s_{t+1}, a_{t+1}, o)$ and the corresponding expected value vanishes in the limit if the opponent plays greedily in the limit.

[1] for our purpose $\|.\|_W$ is a max-norm for a uniform weight-vector, W.

Case 2: $\underline{\max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o) \geq Q_1^t(s_{t+1}, a_{t+1}, o_{t+1})}$.

Again, $\max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o) \leq \pi_1^{*T} Q_1^t(s_{t+1}, ., o_{t+1})$ where $\pi_1^* = \arg\max_{\pi_1} \min_o \pi_1^T Q_1^t(s_{t+1}, ., o)$. Hence $-d_t(s_t, a_t, o_t) = |d_t(s_t, a_t, o_t)| \leq \pi_1^{*T} Q_1^t(s_{t+1}, ., o_{t+1}) - Q_1^t(s_{t+1}, a_{t+1}, o_{t+1})$. The associated expected value vanishes again in the limit due to the assumption of an MLIE policy on part of agent 1, and independent of the opponent's behavior.

Let $C_t(s_t, a_t, o_t)$ be the maximum of the two upper limits on $|d_t(s_t, a_t, o_t)|$ established above. We see that $E\{|d_t(s_t, a_t, o_t)|\} \leq E\{C_t(s_t, a_t, o_t)\}$ and the r.h.s vanishes for each state-action tuple. Hence, $E\{|d_t(s_t, a_t, o_t)|\}$ vanishes for each state-action tuple, which implies that $\|E\{d_t(., ., .)|P_t\}\|$ vanishes in the limit under MLIE exploration and optimal play by the opponent in the limit. Setting c_t in Lemma 1 to $\|E\{d_t(., ., .)|P_t\}\|$, we conclude that minimax-SARSA rule converges to the minimax-Q values under MLIE exploration with probability 1, if the opponent plays greedily in the limit, and under appropriate structure of α_t. [Q.E.D.]

Note that **Case 1** needs the boundedness of Q_1^t that follows easily under additional assumptions. It might be argued that the condition of greedy play by the opponent in the limit is restrictive. However, this is typical of a convergence proof of on-policy algorithms that requires more details of the actions taken by the agents. Gains from on-policy algorithms in terms of learning efficiency and cost offset the condition of greedy play in the limit.

4 Experiments in a Competitive Domain

To evaluate the proposed schemes, we used the purely competitive soccer domain [Littman, 1994]. It is a 4×5 grid containing two agents, A and B, as shown in figure 1, that always occupy distinct squares. The goal of agent A is on the left, and that of B on right. The Figure 1 shows the initial positions of the agents, with the ball being given to an agent at random at the start of each game (agent B in figure). Each agent can choose from a fixed set of five actions at each state: going up, left, down or right, or staying where it is.

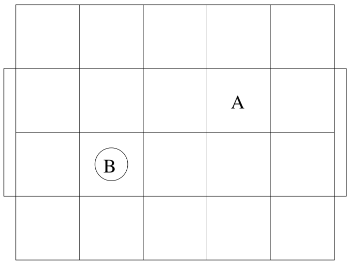

Figure 1: The experimental soccer domain.

When both the agents have selected their actions, these actions are executed in a random order. If an agent bumps onto another, the stationary agent receives the ball, and the movement fails. An agent receives reinforcements of +1 for a goal (or a sameside by the opponent) and -1 for a self-goal (or a goal by the opponent) to maintain the zero-sum character of the game, and in all other cases the reinforcement is zero. Whenever a non-zero reward is received, the game resets to the initial configuration. We shall call an agent following Littman's minimax-Q algorithm an M-agent.

In the training phase of the experiments, we performed symmetric training between two ordinary M-agents, two M-agents both using the $Q(\lambda)$ rule, and two M-agents both using the SARSA rule. The respective policies learnt, are denoted as MM_i, λMM_i, sMM_i, which are recorded at the end of each $i \times 10000$ iterations. Each training lasted 100,000 iterations in all. We used identical exploration-probabilities as that by Littman (1994) and the decay-factor for the learning-rate was set to 0.999954.

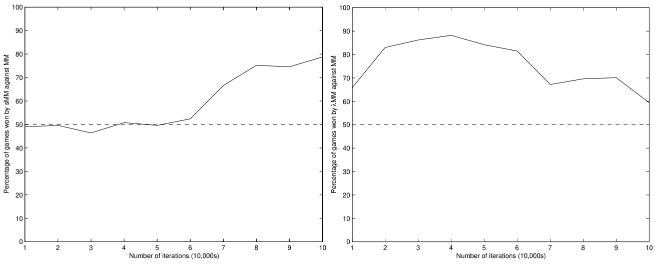

Figure 2: Games are played by sMM_i (left) and λMM_i (right) against MM_i for various values of i (horizontal axis). The percentages of wins (vertical axis) by the former in each case are plotted (averaged over 10 runs).

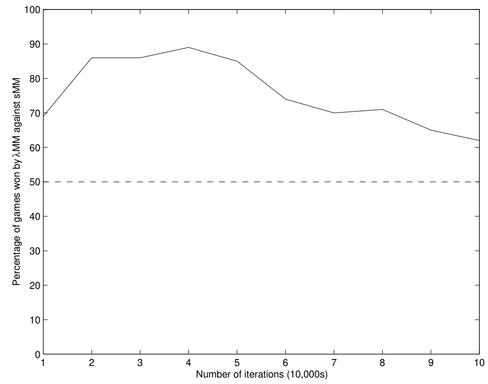

Figure 3: Games are played by λMM_i against sMM_i for various values of i. The percentages of wins (averaged over 10 runs) by the former in each case are plotted.

In the test phase, we allowed an sMM_i policy to play against an MM_i policy, for $i = 1 \ldots 10$. Each test iteration results in a draw with a probability of 0.05, to break possible deadlocks. The resultant percentages of win by the sMM_i policies over its opponent are reported in Figure 2 (left). The approximate trend suggests that ordinary minimax-Q initially dominates but minimax-SARSA gradually catches up and outperforms the former. In Figure 2 (right), the corresponding results from playing λMM_i against MM_i are shown. In this case the minimax-Q(λ) rule outperforms the ordinary minimax-Q algorithm from the very beginning. However, the λMM policies gradually lose their edge as the ordinary minimax-Q rule learns better progressively. The figure 3 corroborates these observations, as λMM_i performs well against sMM_i, but this performance decays with increasing i. λ was set to 0.7 in both experiments.

We note that percentage of wins in such games may not be a good comparative estimator of the policies. A better estimator would be the average RMS deviations of the Q-values from their convergence values. However, the latter can be calculated in this domain only with extensive off-line computation. We also stress that the results reported are far from convergence, at which all the algorithms should perform equally well. The reason why sMM beats MM can be understood in the context of Q updates. While sMM uses the actual action value from the next state to update the current state, MM still uses the minimax value from the next state, which postpones relying on the individual table-entries. As a result, we expect MM to catch up (in Fig. 2 left) when learning continues.

5 Experiments in a General-sum Domain

We note that minimax-Q rule is applicable in general-sum domains as well, where the rationale of the assumption of minimizing policy of the opponent is to guarantee a minimum security level to the learner, instead of maximizing the reward of the opponent itself as in the zero-sum interpretation. The SARSA and $Q(\lambda)$ versions will still work in such domains. For the purpose of experimentation, we introduce a general-sum domain that we call "tightly coupled navigation." This is a 4×3 gridworld as shown in figure 4. The values in the lower left corner of each cell in figure 4 is the reward to agent 1 for reaching the state corresponding to that cell. Similarly the values in the upper right corner are those for agent 2. The rewards in this domain are state-based, i.e. the reward corresponding to a cell is received if the agents reach or remain in that cell. Here the agents are tightly coupled as they must always occupy the same cell. Each agent has three available actions in each state, viz. up, down, right. However, since they are coupled, they can move only when they choose the same action; otherwise they remain in the same state. The starting and the absorbing states have been shown in the figure 4. When the agents reach the goal state, each receives the reward 20 and without making any update in this iteration, the game restarts with the agents reshifted to the start-state and updates begin once again.

A more realistic scenario for this domain is car-driving. Consider two agents in the same car and each having a steering wheel in its hands. The car moves in a given direction if both move the wheels in that direction; otherwise the car does not move. There may be different paths that the agents wish to follow to reach their common goal. However, since they are tightly coupled, they must strike a compromise and find an intermediate path that both can be maximally satisfied with, given the coupling.

We have symmetrically trained two minimax-Q and two minimax-SARSA agents in this domain. The exploration probabilities for the agents in each iteration were the same as in the

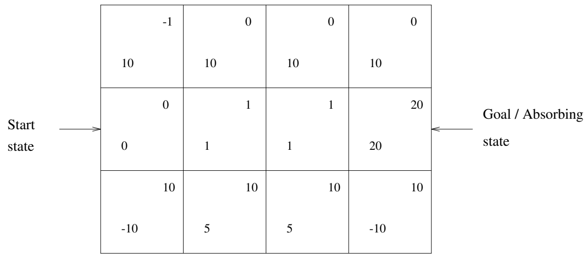

Figure 4: The tightly coupled navigation domain.

soccer domain, viz. 0.2. We varied the probability of reward-generation in each iteration using three values, viz. 0, 0.5 and 1.0, where 0 stands for the case where rewards are generated only when the agents reach the goal state. We wanted to study the effect of infrequent rewards, which is a realistic scenario in most practical domains, on the convergence of our algorithms. We expected the convergence rates to fall with more and more infrequent rewards. In order to study the convergence, the *exact* minimax-Q tables were computed off-line and an average RMS deviation of the learned Q-tables every 1000 training-iterations were plotted. The trainings lasted a total of 10,000 iterations.

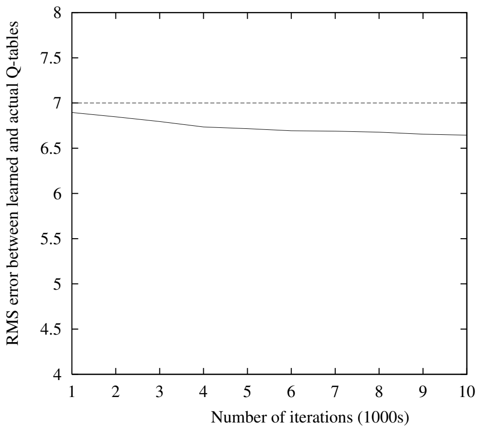

Figure 5: Mean RMS deviation plots of minimax-SARSA (solid) and ordinary minimax-Q for probability of reward-generation = 0.

From figures 5, 6 and 7, we can see that the minimax-SARSA rule always performs better than the ordinary minimax-Q rule. The errors in all the cases decrease monotonically which suggests that both the rules will eventually converge. As expected, the error-levels fall with increasing probability of reward-generation. A scrutiny of the minimax-Q tables show that the minimax path learned by each agent should be different from the Nash-equilibrium path that is corroborated by the learned Q-tables.

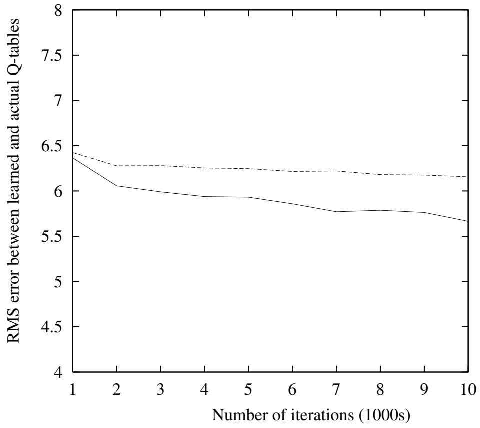

Figure 6: Mean RMS deviation plots of minimax-SARSA (solid) and ordinary minimax-Q for probability of reward-generation = 0.5.

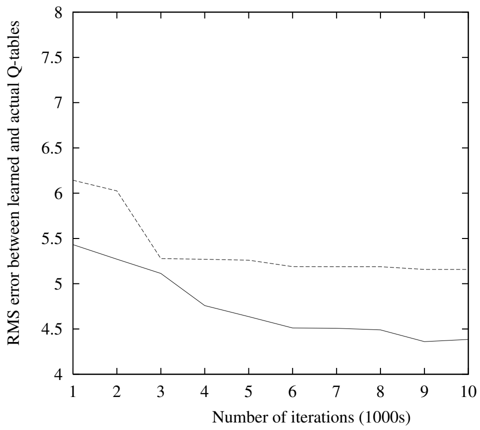

Figure 7: Mean RMS deviation plots of minimax-SARSA (solid) and ordinary minimax-Q for probability of reward-generation = 1.0.

6 Conclusion and Future work

We conclude that both the SARSA and $Q(\lambda)$ versions of minimax-Q learning achieve speed-up on Littman's minimax-Q rule, and more so for the $Q(\lambda)$ rule. Though this latter rule works well, we are not aware of the theoretical convergence properties of this method. Exploring these properties is one open area. We also note that a combination of minimax-SARSA and $Q(\lambda)$ to form what could be called minimax-SARSA(λ), would probably be more expedient than either of the two, by naturally combining their disjoint areas of expedience, seen in the plots in figure 2. Results from associated experiments are awaited. We could also substitute Nash-learning for minimax-learning and achieve Nash-Q(λ) and Nash-SARSA, specialized fast learning procedures for general-sum domains. A theoretical proof of convergence of such a Nash-SARSA would be along the same lines as presented in this paper for minimax-SARSA. We plan to conduct experiments with all these hybrid algorithms.

References

[Boyan and Moore, 1995] J.A. Boyan and A.W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems 7*, pages 369–376, 1995.

[Hu and Wellman, 1998] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proc. of the 15th Int. Conf. on Machine Learning (ML'98)*, pages 242–250, San Francisco, CA, 1998. Morgan Kaufmann.

[Littman, 1994] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of the 11th Int. Conf. on Machine Learning*, pages 157–163, San Mateo, CA, 1994. Morgan Kaufmann.

[Mangasarian and Stone, 1964] O. L. Mangasarian and H. Stone. Two-person nonzero-sum games and quadratic programming. *Journal of Mathematical Analysis and Applications*, 9:348 – 355, 1964.

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

[Peng and Williams, 1996] J. Peng and R. Williams. Incremental multi-step Q-learning. *Machine Learning*, 22:283 – 290, 1996.

[Rummery, 1994] G. A. Rummery. *Problem solving with reinforcement learning*. PhD thesis, Cambridge University Engineering Department, 1994.

[Sandholm and Crites, 1996] T. Sandholm and R. Crites. On multiagent Q-learning in a semi-competitive domain. In G. Weiß and S. Sen, editors, *Adaptation and Learning in Multi-Agent Systems*, pages 191–205. Springer-Verlag, 1996.

[Sutton and Burto, 1998] R. Sutton and A. G. Burto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Sutton, 1996] R. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, 1996.

[Szepesvári and Littman, 1997] Csaba Szepesvári and M.L. Littman. A unified analysis of value-function-based reinforcement-learning algorithms. In *Neural Computation*, 1997. submitted.

[Thie, 1998] P. R. Thie. *An Introduction to Linear Programming and Game Theory*. John Wiley and Son, 2nd. edition, 1998.

[Tsitsiklis and Roy, 1997] J.N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 1997.

[Wiering and Schmidhuber, 1998] M. Wiering and J. Schmidhuber. Fast online Q(λ). *Machine Learning, 33(1)*, pages 105–116, 1998.