
Experimental Results on Q-Learning for General-Sum Stochastic Games

Junling Hu

School of Business, University of Mississippi, University, MS 38677, USA

JUNLING@OLEMISS.EDU

Michael P. Wellman

Artificial Intelligence Laboratory, University of Michigan, Ann Arbor, MI 48109-2110, USA

WELLMAN@UMICH.EDU

Abstract

The Q-learning method we previously introduced for general-sum stochastic games is guaranteed to converge given some restrictions on the form of the game, both the actual game and the agents' model of the game during learning. In experimental trials on a two-agent grid world, we find that violations of the restriction to unique equilibrium values for the underlying game impede convergence, whereas violations for the modeled game during learning are less important. In comparisons of multiagent Q-learning with multiple single-agent Q-learning, we find that the former performs better in general.

1. Introduction

Stochastic games (Filar & Vrieze, 1997; Thusijnsman, 1992) are multi-stage games in which agents' payoff functions may change from stage to stage. Each agent in such a game faces a Markov decision process (MDP), which is intertwined with other agents' MDPs. The framework of stochastic games can be used to model a wide range of dynamic multiagent systems such as coordination games (Claus & Boutilier, 1998), pursuit games (Denzinger & Fuchs, 1996; Ono & Fukumoto, 1996), and robotic soccer (Balch, 1997; Stone, 1998).

In a previous paper (Hu & Wellman, 1998), we introduced a multiagent Q-learning algorithm for general-sum stochastic games. In our learning algorithm, an agent learns about Q-values that are functions of states and agents' joint actions. An agent's Q-table is a set of Q-values for every state and joint action. The optimal Q-value is defined in *Nash equilibrium* (Nash, 1951), where each agent's strategy is a best response to the others' and no agent can gain by unilateral deviation. We proved the convergence of our learning

algorithm under the restrictions on the Nash equilibrium Q-tables and Q-tables during learning.

The restriction on the agents' Q-tables during learning is particularly strong because we can hardly guarantee the form of these Q-tables during learning. This limits the scope of the application of this learning algorithm. We are interested in relaxing this restriction. Before developing a theoretical proof, we use experiments to test the possibility. Our experiments also serve to shed light on the possibility of relaxing the other restriction on the equilibrium Q-tables.

We have constructed two grid-world games. The first game satisfies the restriction on equilibrium Q-tables but violates the restriction on Q-tables during learning. The second game violates both restrictions. The multiagent Q-Learning converges in the first game, but does not converge in the second game. Our experimental results suggest that the restriction on Nash equilibrium Q-tables is necessary for the convergence, but the restriction on Q-tables during learning is less important.

We compare the performance of this multiagent Q-learning method with single-agent Q-learning method. In both games, our learning agent is more likely to reach a joint optimal path when it uses multiagent Q-learning than using single-agent Q-learning. This implies the better offline-learning performance of multiagent Q-learning than single-agent Q-learning.

2. Review of Multiagent Q-learning

In standard (single-agent) Q-learning, an agent updates its Q-values based on the following equation:

$$Q_{t+1}(s, a) = (1 - \alpha_t)Q_t(s, a) + \alpha_t[r_t + \beta \max_b Q_t(s_{t+1}, b)], \quad (1)$$

where $s \in S$ is a state, $a \in A$ is an action, $\alpha_t \in [0, 1]$ is the learning rate, r_t is the reward at time t , and β is

the discount rate. Watkins and Dayan (1992) proved that sequence (1) converges to the optimal $Q^*(s, a)$, which is defined as the total discounted reward attained by taking action a in state s and then following the optimal policy thereafter.

For an n -player stochastic game, we define the *Nash equilibrium Q-values* of agent k , $Q_*^k(s, a^1, \dots, a^n)$, as the agent's total discounted rewards when all agents execute their joint actions (a^1, \dots, a^n) in state s and then follow their Nash equilibrium strategies π_*^1, \dots, π_*^n thereafter. A set of strategies $(\pi_*^1, \dots, \pi_*^n)$ is a Nash equilibrium if for $k = 1, \dots, n$, the strategy π_*^k is the best response to the other strategies $(\pi_*^1, \dots, \pi_*^{k-1}, \pi_*^{k+1}, \dots, \pi_*^n)$. A *Nash equilibrium Q-table* of agent k , Q_*^k , is a set of Nash equilibrium Q-values for every state and joint actions.

The fact that an agent's Nash equilibrium Q-values depend on other agents' strategies requires the agent to learn about those strategies. One way to know those strategies is through learning other agents' Q-tables. Our learning agent, say agent k , maintains n Q-tables, one for itself and one for every other agent. Let agent k 's internal Q-table of agent i be Q^i . $Q^i(s)$ is part of the Q-table Q^i under state s . Agent k updates the entries in each table $Q^i, i = 1, \dots, n$, according to the following rule:

$$Q_{t+1}^i(s, a^1, \dots, a^n) = (1 - \alpha_t)Q_t^i(s, a^1, \dots, a^n) + \alpha_t[r_t^i + \beta \pi^1(s_{t+1}) \cdots \pi^n(s_{t+1})Q_t^i(s_{t+1})], \quad (2)$$

where $\pi^1(s_{t+1}) \cdots \pi^n(s_{t+1})Q_t^i(s_{t+1})$ represents the product of $\pi^1(s_{t+1}), \dots, \pi^n(s_{t+1})$, and $Q_t^i(s_{t+1})$. The tuple $(\pi^1(s_{t+1}), \dots, \pi^n(s_{t+1}))$ is a mixed-strategy Nash equilibrium of the normal-form game $(Q_t^1(s_{t+1}), \dots, Q_t^n(s_{t+1}))$.

We proved the convergence of this learning algorithm for 2-player stochastic games. Our convergence theorem relies on the following assumptions:

Assumption 1 *Every state $s \in S$ and action $a^1 \in A^1, a^2 \in A^2$ are visited infinitely often.*

Assumption 2 *The learning rate α_t satisfies the following conditions for all s, t, a^1, a^2 :*

1. $0 \leq \alpha_t(s, a^1, a^2) < 1$, $\sum_{t=0}^{\infty} \alpha_t(s, a^1, a^2) = \infty$, and $\sum_{t=0}^{\infty} [\alpha_t(s, a^1, a^2)]^2 < \infty$, and the latter two hold uniformly and with probability 1.
2. $\alpha_t(s, a^1, a^2) = 0$ if $(s, a^1, a^2) \neq (s_t, a_t^1, a_t^2)$.

The second condition on α_t implies that the learning agent updates only the entry in the Q-tables corresponding to current state and current actions chosen by the agents.

Assumption 3 *Let Q_*^1 and Q_*^2 be the agents' Nash equilibrium Q-tables. Let Q_t^1 and Q_t^2 be Q-tables learned at time t . For every t and every state $s \in S$, a Nash equilibrium of the bimatrix game $(Q_*^1(s), Q_*^2(s))$ and a Nash equilibrium of the bimatrix game $(Q_t^1(s), Q_t^2(s))$ satisfy the same one of the following properties:¹*

1. *The Nash equilibrium is optimal for both agents, meaning both agents receive their highest payoffs when they choose the Nash equilibrium strategies.*
2. *The Nash equilibrium is a saddle point, which means an agent receives a higher payoff when the other agent deviates from the equilibrium strategy.*

Note that the two properties are exclusive. A Nash equilibrium satisfying property 1 in general should not satisfy property 2 except in special cases.²

Assumption 3 puts strong restrictions on the Q-tables during learning, and the Nash equilibrium Q-tables. In general, the restriction on the Q-tables during learning can hardly be met because the updates are asynchronous. Therefore we are interested in relaxing these restrictions. Before developing a theoretical proof, we would like to use experiments to test the possibility.

3. Experimental Setup

We construct two grid-world games with different properties to test the convergence of our learning algorithm. Our goal is to see to what extent the restrictions in Assumption 3 might be relaxed. Our first game satisfies the condition on games, but does not ensure that it is invariant during learning. Our second game violates the condition even for the actual game.

Even though grid-world games are highly simplified multiagent worlds, they have all the elements needed in a dynamic game: the changing states (when agents move around), the actions taken at each location, the reward functions, and the optimal decision that takes an agent from its initial location to its destination. Similar grid games with one agent have been studied by Sutton and Barto (1998) and Mitchell (1997, Chapter 13). A two-person zero-sum grid game has

¹In our statement of the assumption in a previous paper (Hu & Wellman, 1998), we neglected to include the qualification that the *same* condition be satisfied by both bimatrix games. As Bowling (Bowling, 2000) observed, this qualification is crucial.

²If every entry of $Q^1(s)$ is the same, then any strategy would be agent 1's Nash equilibrium strategy. In this case, a Nash equilibrium point is both a global optimum and a saddle point.

been studied by Littman (1994). Our grid games are two-person general-sum games in which both agents can win at the same time.

Our first game is shown in Figure 1. The second game is shown in Figure 2. In both games, two agents start from the lower left corner and lower right corner, trying to reach their goal cells. In Grid Game 1, the goal cells are at the upper right and the upper left corner. In Grid Game 2, both agents' goal cells are the upper middle cell. An agent can move only one cell a time and in 4 possible directions: *Left*, *Right*, *Up*, *Down*. If two agents attempt to move into the same cell (excluding a goal cell), they are bounced back to their previous cells. The game ends as soon as one agent reaches its goal. The agent who reaches its goal get a positive payoff, while the one who does not gets nothing. Both agents may reach their goal cells at the same time. In that case, both are rewarded with positive payoffs.

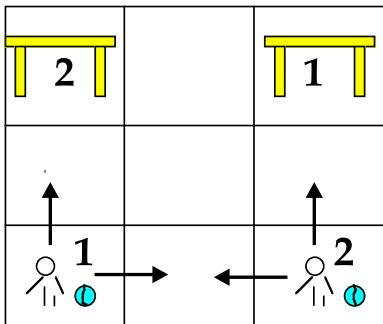


Figure 1. Grid game 1

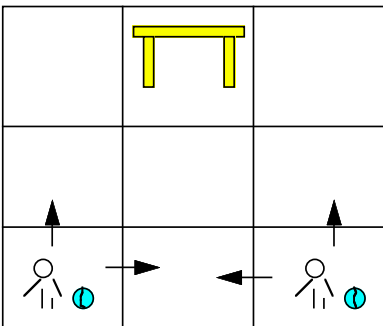


Figure 2. Grid game 2

The objective of an agent in this game is therefore to reach its goal with minimum steps. If the other agent takes less steps to reach its goal cell, our agent will get nothing. One possibility arising from zero-sum games is that an agent may block the other agent's path so that none of them can reach the goal. This is not an

optimal strategy in our game. The fact that the other agent's win does not prevent our agent's winning makes agents more prone to coordination. The task of an agent is to find a shortest path that does not interfere with the other agent's path.

A path (plan) is a sequence of actions from the starting position to the final destination. In the terminology of stochastic games, such a plan is called a *strategy*, or policy. A shortest path that does not interfere with the other agent's path represents an optimal strategy. Two shortest paths that do not interfere with each other constitute a Nash equilibrium, since each path (strategy) is a best response to the other.

We assume that agents do not know the locations of their goals at the beginning of the game. Furthermore, agents do not know their own and the other agent's payoff functions.³ Agents choose their actions simultaneously. They can observe the previous actions of both agents and the current state (the joint position of both agents). They also observe the immediate rewards after both agents choose their actions.

3.1 Representation as a Stochastic Game

The grid-world games can be easily modeled as stochastic games. The action space for each agent is $A^1 = A^2 = \{Left, Right, Down, Up\}$. The state space is $S = \{s | s = (l^1, l^2)\}$, where each state $s = (l^1, l^2)$ represents the agents' joint location. Agent i 's location l^i , $i = 1$ or 2 , is represented by (X, Y) coordinates, as shown in Figure 3. Given that two agents cannot occupy the same position, and excluding the cases where at least one agent is in its goal cell, the number of possible joint positions is $72 - 8 - 7 = 57$ for Grid Game 1 and $8 \times 7 = 56$ for Grid Game 2.

If an agent reaches its goal position, it scores 100 points. If it reaches other positions without colliding with the other agent, it scores 0 points. If it collides with the other agent, it scores -1 and both agents are bounced back to their previous positions. These scores are the agents' immediate rewards.

The state transitions are deterministic in Grid Game 1, which means the current state and agents' joint action will uniquely determine the next state. In Grid Game 2, most state transitions are deterministic except in the following case: If an agent chooses *Up* from position $(1, 1)$ or $(3, 1)$, it moves up with probability 0.5 and remains in its previous position with probability 0.5.

³Note that a payoff function is a correspondence from all state-action tuples to rewards. An agent may be able to observe a particular reward, but still lacks the knowledge of the overall payoff function.

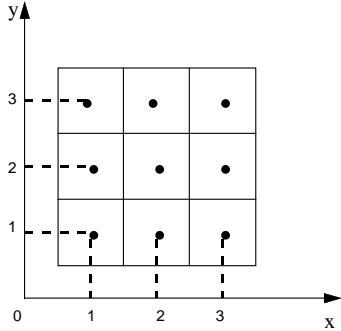


Figure 3. The coordinates for a grid-world game

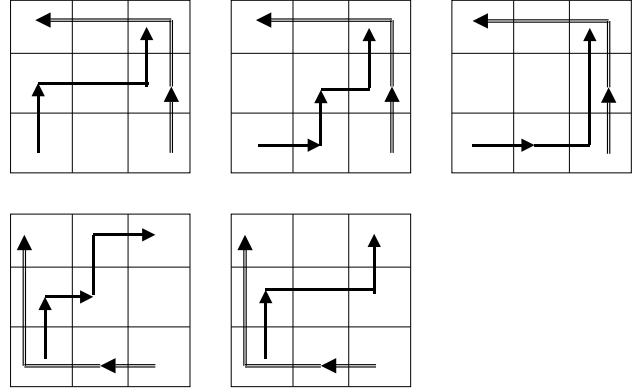


Figure 4. Nash equilibrium paths, Grid Game 1

We limit our study to stationary strategies. A stationary strategy $\pi^i = (\pi^i(s^1), \dots, \pi^i(s^m))$ assigns a probability distribution over available actions for each state s^j , $j = 1, \dots, m$, where m is the number of states. $\pi^i(s^j)$ is called a *pure strategy* if it assigns probability 1 to one of the actions. From a stationary strategy with a pure strategy for each state, we can derive a path, which is an agent's movements from its initial position to its goal cell.

3.2 Nash Equilibrium

A Nash equilibrium consists of a pair of stationary strategies (π_*^1, π_*^2) in which each strategy is a best response to the other. Two shortest paths that do not interfere with each other constitute a Nash equilibrium solution. Table 1 shows one Nash equilibrium of Grid Game 1. This Nash equilibrium corresponds to the first graph in Figure 4, which also shows several other Nash equilibrium paths of Grid Game 1.

Table 1. A Nash equilibrium of Grid Game 1

STATE s	$\pi^1(s)$	$\pi^2(s)$
$s^1 = ((1,1)(3,1))$	UP	UP
$s^2 = ((1,1)(2,1))$	UP	UP
\vdots	\vdots	\vdots
$s^7 = ((1,2)(3,2))$	RIGHT	UP
$s^8 = ((1,2)(3,3))$	RIGHT	LEFT
\vdots	\vdots	\vdots
$s^{5b} = ((3,2)(2,3))$	UP	LEFT
$s^{5f} = ((3,2)(3,3))$	UP	LEFT

Examples of Nash equilibrium paths for Grid Game 2 are shown in Figure 5.

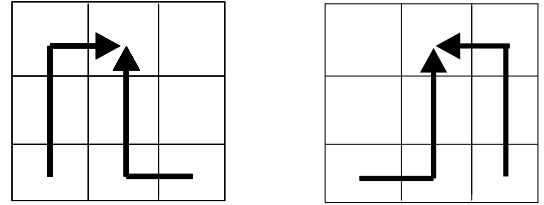


Figure 5. Nash equilibrium paths, Grid Game 2

3.3 Nash Equilibrium Q-values

As defined in Section 2, a Nash equilibrium Q-value for the tuple (s, a^1, a^2) is an agent's total discounted reward when both agents execute the joint action (a^1, a^2) in state s and follow a pair of Nash equilibrium strategies (π_*^1, π_*^2) thereafter.

3.3.1 EQUILIBRIUM Q-VALUES OF GRID GAME 1

Agents 1 and 2's Nash equilibrium Q-values of Grid Game 1 for state $((1,1)(3,1))$ are shown in Table 2. The value 97 is derived from the fact that agents receive 0 at the current time period, and will receive 100 after 3 periods if they follow their joint optimal path, $97 = 0 + 0.99^3 \times 100$. In the table, the values of R_1 and R_2 depend on which Nash equilibrium the agents choose in the next state. The next state under joint action $(Right, Left)$ and current state $((1,1)(3,1))$ is still $((1,1)(3,1))$. There exist three pure-strategy Nash equilibria for the bimatrix game in state $((1,1)(3,1))$: $\{Right, Up\}$, $\{Left, Up\}$ and $\{Up, Up\}$. Therefore $R_1 = 97$ and $R_2 = 97$. Thus Table 2 can be further simplified as Table 3. Every Nash equilibrium is a global optimum and has the same value as the other equilibria. Thus the first condition in Assumption 3 is satisfied.

Table 2. Grid Game 1: Nash equilibrium Q-tables in state $((1, 1)(3, 1))$

		AGENT 2	
		<i>Left</i>	<i>Up</i>
AGENT 1	<i>Right</i>	$-1 + 0.99R_1, -1 + 0.99R_2$	97, 97
	<i>Up</i>	97, 97	97, 97

Table 3. Grid Game 1: Nash equilibrium Q-values in state $((1, 1)(3, 1))$

		AGENT 2	
		<i>Left</i>	<i>Up</i>
AGENT 1	<i>Right</i>	95, 95	97, 97
	<i>Up</i>	97, 97	97, 97

3.3.2 EQUILIBRIUM Q-VALUES OF GRID GAME 2

The Nash equilibrium Q-values of Grid Game 2 for state $((1, 1)(3, 1))$ are shown in Table 4. In the table, the values of R_1 and R_2 depend on which Nash equilibrium the agents choose in the next state. There are three different solutions for (R^1, R^2) , which lead to different Nash equilibrium Q-tables for state $((1, 1)(3, 1))$, shown in Table 5. The mixed-strategy Nash equilibrium of the third table is $(\pi^1(s^1), \pi^2(s^1)) = ((0.97, 0.03), (0.97, 0.03))$. The two pure-strategy Nash equilibria are $\{Right, Up\}$ and $\{Left, Up\}$. It is easy to check that these Nash equilibria are neither global optima nor saddle points. Therefore the first condition of Assumption 3 is violated.

Table 4. Grid Game 2: Nash equilibrium Q-tables in state $((1, 1)(3, 1))$

		AGENT 2	
		<i>Left</i>	<i>Up</i>
<i>Right</i>	$-1 + 0.99R_1, -1 + 0.99R_2$	98, 48.5	
<i>Up</i>	48.5, 98	$48.5 + \frac{1}{4}R_1, 48.5 + \frac{1}{4}R_2$	

4. Experimental Results

4.1 The Learning Process

A learning agent learns its own Q-table and the Q-table of its counterpart. In cases where both agents are learning agents, each of them learns two Q-tables. If the two agents start from the same initial Q-tables and use the same Nash equilibrium to update every Q-table during learning, they would learn the same Q-tables in the end of the game.

A learning agent initializes $Q^1(s, a^1, a^2) = 0$ and $Q^2(s, a^1, a^2) = 0$ for all s, a^1, a^2 . A game starts from the initial state $((1, 1)(3, 1))$. After observing the cur-

Table 5. Grid Game 2: Two pure-strategy Nash equilibrium Q-tables and one mixed-strategy Nash equilibrium Q-table in state $((1, 1)(3, 1))$

		AGENT 2	
		<i>Left</i>	<i>Up</i>
AGENT 1	<i>Right</i>	47, 96	98, 48.5
	<i>Up</i>	48.5, 98	60.6, 73

		AGENT 2	
		<i>Left</i>	<i>Up</i>
AGENT 1	<i>Right</i>	96, 47	98, 48.5
	<i>Up</i>	48.5, 98	73, 60.6

MIXED-STRATEGY NASH EQUILIBRIUM

		AGENT 2	
		<i>Left</i>	<i>Up</i>
AGENT 1	<i>Right</i>	47.34, 47.34	98, 48.5
	<i>Up</i>	48.5, 98	60.6, 60.6

rent state, agents choose their actions simultaneously. They then observe the new state, both agents' rewards and the action taken by the other agent. The learning agent updates its Q-tables according to (2). The discount rate $\beta = 0.99$. In the new state, agents repeat the process above. When at least one agent move into its goal position, the game re-starts with a new episode. In the new episode, each agent is randomly assigned a new position (except its goal cell). The learning agent still keeps the Q-values learned from previous episodes. The experiment (training) stops after 5000 episodes. Each episode on average takes about 8 steps. So one experiment usually takes about 40,000 steps.⁴ The total number of state-action tuples in Grid Game 1 is 424. This suggests that each tuple is visited about 95 times. This is an approximation to "infinitely often" stated in Assumption 1. As shown in the next paragraph, the learning rate is defined as the inverse of the time of visits. When $\alpha_t = \frac{1}{95} = 0.01$, the results from new visits would hardly change the Q-values already learned.

The learning rate we adopt is $\alpha_t(s, a^1, a^2) = \frac{1}{n_t(s, a^1, a^2)}$, where $n_t(s, a^1, a^2)$ is the number of times the tuple (s, a^1, a^2) has been visited. It is easy to show that this definition of learning rate satisfies the conditions $\sum_t \alpha_t(s, a^1, a^2) = \infty$ and $\sum_t \alpha_t^2(s, a^1, a^2) < \infty$ required by Assumption 2.

⁴Note that a stochastic game is represented by one episode. In each episode, there are finite periods before the game ends. The decision problem of each individual agent can still be modeled as an infinite-horizon problem because the agent is uncertain when the game will end. Different episodes represent the training periods during learning.

When updating the Q-values, an agent has to choose a Nash equilibrium value from the next bimatrix game $(Q^1(s'), Q^2(s'))$, where s' is the next state. There may be multiple Nash equilibria. The Nash equilibria are calculated by Lemke-Howson method (Cottle et al., 1992) and are listed in a certain order. An agent can have different ways of picking a Nash equilibrium from the solution list. A *First Nash agent* always picks the first one from the list. A *Second Nash agent* always picks the second one if there are more than one in the list, otherwise it uses the first one. A *Best expected Nash agent* picks the Nash equilibrium which yields the highest expected payoff to itself if there are more than one in the list, otherwise it uses the first one.

4.2 Q-tables during Learning

We find out that in Grid Game 1 the bimatrix games derived from Q-tables during learning violates the second restriction in Assumption 3. Table 6 shows the results for state $((1,1)(3,1))$ after certain learning episodes. The pure-strategy Nash equilibrium in this game, $(Right, Up)$, is neither a global optimum nor a saddle point. In Grid Game 2, we find the similar violation of Assumption 3. The pure-strategy Nash equilibrium (Up, Up) in Table 7 is neither a global optimum nor a saddle point.

Table 6. Grid Game 1: Q-tables in state $((1,1)(3,1))$ after 20 episodes if always choosing the first Nash

		AGENT 2	
		Left	Up
AGENT 1	Right	-1, -1	48.5, 0
	Up	0, 0	0, 97

Table 7. Grid Game 2: Q-tables in state $((1,1)(3,1))$ after 61 episodes if always choosing the first Nash

		AGENT 2	
		Left	Up
AGENT 1	Right	31, 31	0, 65
	Up	0, 0	49, 49

4.3 Convergence Results: The Final Q-tables

In each experiment, after 5,000 episodes of training the learning agent's Q-tables settle down to certain values. One example of such learned Q-tables for Grid Game 1 is shown in Table 8. Two learned Q-tables for Grid Game 2 are shown in Table 9 and Table 10. Two things are to be noted here: First, a learning agent always uses the same Nash equilibrium value to update both Q-tables. Second, an agent's learning results are not affected by the other agent's action choice

or learning algorithm. That's because the convergence of Q-learning only requires an agent to observe (for infinite times) all possible actions and states, regardless of the sequence of these actions and states.

Table 8. Final Q-values in state $((1,1)(3,1))$ if choosing the first Nash in Grid Game 1

		AGENT 2	
		Left	Up
AGENT 1	Right	86, 87	83, 85
	Up	96, 91	95, 95

Table 9. Final Q-values in state $((1,1)(3,1))$ if choosing the first Nash in Grid Game 2

		AGENT 2	
		Left	Up
AGENT 1	Right	39, 84	97, 51
	Up	46, 93	59, 74

Table 10. Final Q-values in state $((1,2)(2,1))$ if choosing the first Nash in Grid Game 2

NASH EQUILIBRIUM Q-VALUES

		AGENT 2		
		Left	Right	Up
AGENT 1	Right	99, 0	99, 0	97, 97
	Down	97, 97	98, 98	0, 99
	Up	99, 0	99, 0	99, 99

FINAL Q-VALUES

		AGENT 2		
		Left	Right	Up
AGENT 1	Right	97, 0	98, 0	84, 87
	Down	86, 86	41, 88	0, 99
	Up	99, 0	99, 0	99, 96

We can see that the results in Table 8 are very close to the theoretical derivation in Table 3, and the results in Table 9 are close to the theoretical derivation in the first table of Table 5. For each state s^j , we derive a Nash equilibrium $(\pi^1(s^j), \pi^2(s^j))$ from the learned Q-tables $(Q^1(s^j), Q^2(s^j))$. A sequence of such solutions of every state, $((\pi^1(s^1), \pi^2(s^1)), \dots, (\pi^1(s^m), \pi^2(s^m)))$, is supposed to be a Nash equilibrium for the whole grid game. We then see if this solution is a Nash equilibrium derived from theory. In Grid Game 1, our experiments show that multiagent Q-learning reaches a Nash equilibrium 100% of times if the learning agent is either a First Nash agent or a Second Nash agent. In Grid Game 2, however, multiagent Q-learning only reaches a Nash equilibrium 68% of times if the agent is a First Nash agent, and 90% of times if the agent

is a Second Nash agent. Therefore the convergence of this Q-learning is not guaranteed in Grid Game 2.

4.4 Offline-Learning Performance of Different Types of Learning Agents

One of the practical concerns in applying this multiagent Q-learning method is whether an agent gains by modeling the other agent. So far in practice, most people still uses single-agent Q-learning (Ono & Fukumoto, 1996; Stone, 1998) for learning in multiagent systems. We investigate through our experiments the performance under multiagent Q-learning and under the single-agent Q-learning when the other agent uses different learning methods.

An agent can assume one of the following four types: the First Nash agent, the Second Nash agent, the Best Expected Nash agent, and the Single agent. The Single agent uses single-agent Q-learning method specified in (1). Such an agent ignores the actions of the other agent. We assume that the Single agent knows the other agent’s current position so that it can use that information as part of the state variable. We want to see if two agents can reach a joint optimal path when taking on these different types.

The experimental results for Grid Game 1 are shown in Table 11. For each case, we ran 50 experiments. The percentage represents the number of times of reaching a joint optimal path out of 50 runs. As we can see from the table, when both agents are Single agents, they reach a Nash equilibrium only 20% of times. This is not surprising since the Single agent never models the other agent’s actions. It is therefore difficult for such agents to coordinate their paths. When one agent is Nash agent and the other is Single agent, the chance of reaching a Nash equilibrium increases to 62% on average. This is as we expected because a Nash agent takes the other agent’s action into account and tries to avoid conflicting paths. When both agents are Nash agents, but use different updating rules, they will end up with a Nash equilibrium 80% of times.⁵ Finally, when both agents are Nash agents and use the same updating rule, they end up with a Nash equilibrium solution all the time.

The experimental results for Grid Game 2 are shown in Table 12. As we can see from the table, when both agents are Single agents, they reach a Nash equilibrium 50% of times. When one agent is a Nash agent and

⁵Note that when both agents are Expected Nash agents, they may use different Nash equilibrium from the solution list. A Nash equilibrium solution giving the best expected payoff to one agent may not lead to the best expected payoff for another agent.

Table 11. Learning performance in Grid Game 1

LEARNING STRATEGY		RESULTS OF LEARNING
AGENT 1	AGENT 2	PERCENTAGE OF TOTAL RUNS THAT REACH A NASH EQUILIBRIUM
SINGLE	SINGLE	20%
SINGLE	FIRST NASH	60%
	SECOND NASH	50%
	BEST EXPECTED NASH	76%
FIRST NASH	SECOND NASH	60%
	BEST EXPECTED NASH	76%
SECOND NASH	BEST EXPECTED NASH	84%
BEST EXPECTED NASH	BEST EXPECTED NASH	100%
FIRST NASH	FIRST NASH	100%
SECOND NASH	SECOND NASH	100%

the other is a Single agent, the chance of reaching a Nash equilibrium increases to 51.3% on average. When both agents are Nash agents, but use different updating rules, they will end up with a Nash equilibrium 55%⁶ of times on average. Finally, when both agents are Nash agents and use the same updating rule, they end up with a Nash equilibrium 79%⁷ of times.

The above experiments show that agents are more likely to reach a joint optimal path when they follow multiagent Q-learning than follow single-agent Q-learning. This implies that an agent in general performs better under multiagent Q-learning than under single-agent Q-learning, regardless the learning method used by another agent.

5. Conclusions

We test the convergence of a multiagent Q-learning algorithm for general-sum stochastic games in two grid-world games. The grid-world games are constructed to relax the assumptions on the property of Q-tables during learning and the property of the Nash equilibrium Q-tables. Our experiments show that the multiagent Q-learning converges in the game satisfying the restriction on Nash equilibrium Q-tables but violating the restriction on Q-tables during learning. The learning fails to converge in the game violating both

⁶This is calculated from $\frac{64\%+78\%+36\%+42\%}{4}$.

⁷This is calculated from $\frac{68\%+90\%}{2}$.

Table 12. Learning performance in Grid Game 2

LEARNING STRATEGY		RESULTS OF LEARNING
AGENT 1	AGENT 2	PERCENTAGE OF TOTAL RUNS THAT REACH A NASH EQUILIBRIUM
SINGLE	SINGLE	50%
SINGLE	FIRST NASH	54%
	SECOND NASH	62%
	BEST EXPECTED NASH	38%
FIRST NASH	SECOND NASH	64%
	BEST EXPECTED NASH	78%
SECOND NASH	BEST EXPECTED NASH	36%
BEST EXPECTED NASH	BEST EXPECTED NASH	42%
FIRST NASH	FIRST NASH	68%
SECOND NASH	SECOND NASH	90%

restrictions. These results suggest that the restriction on Nash equilibrium Q-tables is necessary for convergence, but the restriction on Q-tables during learning is not essential. This poses the possibility to relax the latter restriction in future theoretical study.

We compare the performance of this multiagent Q-learning method with single-agent Q-learning method. When the other agent assumes different types, in both games, our learning agent is more likely to reach a joint optimal path when it uses multiagent Q-learning than using single-agent Q-learning. This implies that the offline-learning performance of multiagent Q-learning is better than single-agent Q-learning. In the future, we would like to investigate the online learning performance of this multiagent Q-learning algorithm.

Acknowledgments

We would like to thank Michael Littman and Csaba Szepesvári for their helpful discussions.

References

Balch, T. (1997). Learning roles: Behavioral diversity in robot teams. *Collected Papers from the AAAI-97 Workshop on Multiagent Learning*. AAAI Press.

Bowling, M. (2000). Convergence problems of general-sum multiagent reinforcement learning. Unpub-

lished manuscript, Computer Science Department, Carnegie Mellon University, Pittsburgh. <http://www.cs.cmu.edu/~mhb/publications/index.html>.

- Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 746–752). Madison, WI.
- Cottle, R. W., Pang, J.-S., & Stone, R. E. (1992). *The linear complementarity problem*. New York: Academic Press.
- Denzinger, J., & Fuchs, M. (1996). Experiments in learning prototypical situations for variants of the pursuit game. *Proceedings of the Second International Conference on Multiagent Systems* (pp. 48–55). Kyoto, Japan: AAAI Press.
- Filar, J., & Vrieze, K. (1997). *Competitive markov decision processes*. Springer-Verlag.
- Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 242–250). Madison, WI: AAAI Press.
- Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 157–163). New Brunswick.
- Mitchell, T. (1997). *Machine learning*, 367–390. McGraw-Hill.
- Nash, J. F. (1951). Non-cooperative games. *Annals of Mathematics*, 54, 286–295.
- Ono, N., & Fukumoto, K. (1996). Multi-agent reinforcement learning: A modular approach. *Proceedings of the Second International Conference on Multiagent Systems* (pp. 252–258). Kyoto, Japan: AAAI Press.
- Stone, P. (1998). *Layered learning in multi-agent systems*. Doctoral dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.
- Thusijnsman, F. (1992). *Optimality and equilibria in stochastic games*. Amsterdam: Centrum voor Wiskunde en Informatica.
- Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, 3, 279–292.