Fuzzy Q-Learning for a Multi-Player Non-Cooperative Repeated Game

Hisao Ishibuchi, Tomoharu Nakashima, Hiromitsu Miyamoto and Chi-Hyon Oh
Department of Industrial Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka 593, Japan
{hisaoi, nakashi, miyamoto, oh}@ie.osakafu-u.ac.jp

Abstract

In this paper, we examine the applicability of fuzzy Q-learning to a multi-player non-cooperative repeated game. First we formulate a transportation problem as a repeated game where many agents (i.e., many game players) compete with one another at several markets. Each agent is supposed to choose one market for maximizing his own profit obtained by selling his product at that market. It is assumed in our game that the market price of the product is determined by the demand-supply relation at each market. For example, if many agents bring their products to a particular market, the market price becomes low. On the contrary, the market price is high if the total amount of products brought to that market is small. In this manner, the price at each market is determined by the actions of all agents. After formulating the repeated game, we explain how Q-learning can be employed by each agent for choosing a market. Then the Q-learning is extended to fuzzy Q-learning for utilizing the information about the previous market prices when each agent chooses a market. The previous price of each market is represented by two fuzzy linguistic values "low" and "high." By computer simulations on a numerical example with 100 agents and five markets, we clearly show that the fuzzy Q-learning can learn effective strategies as fuzzy if-then rules for choosing a market.

Keywords: *Repeated games, Q-learning, multi-agent systems, fuzzy rules, reinforcement learning.*

1. Introduction

Strategies for repeated games have been mainly investigated for the Prisoner's Dilemma [1]. Genetic algorithms [2,3] were employed for evolving strategies of the iterated Prisoner's Dilemma game [4-6]. In this paper, we try to apply a reinforcement learning scheme to a multi-player non-cooperative repeated game. Our game, which is a kind of quadratic transportation problem, involves much more players (*e.g.*, 100 players) and a more complicated payoff mechanism than the Prisoner's Dilemma. In our game, each agent (*i.e.*, each game player) is supposed to choose one market from several ones and to sell his product at the market price of the selected market. The aim of the market selection is to maximize his own profit obtained by selling his product at the market price. It is assumed in our game that the market price is determined by the demand-supply relation at each market. For example, if many agents bring their products to a particular market, the price of the products at that market becomes low. On the contrary, the market price is high if the total amount of products brought to that market is small. In this manner, the price at each market is determined by the actions of all agents. Thus the profit of a particular agent depends on the actions of the other agents.

In this paper, we first formulate the transportation problem with many agents and several markets as a multi-player non-cooperative repeated game. Then we show some strategies for our game, each of which is based on Q-learning [7] or fuzzy Q-learning [8-11]. The Q-learning is one of the most-well known reinforcement learning schemes. A Q-value is assigned to each state-action pair, and it is updated based on the reinforcement signal (*i.e.*, reward or punishment) given from the environment after a particular action or after a series of several actions. The Q-learning dose not use explicit targets that are usually required in supervised learning mechanisms such as the back-propagation algorithm of multi-layer feedforward neural networks. The fuzzy Q-learning [8-11] is an extension of the Q-learning to the case of continuous states and/or actions. In the fuzzy Q-learning, the Q-value is calculated by a fuzzy inference system based on a set of fuzzy if-then rules. In our game, the action is not continuous while the state variables (*i.e.*, market prices) are handled as real numbers. As an alternative strategy, we consider the optimal choice for the previous actions, in which an agent chooses the best market by assuming that the actions of the other agents are exactly the same as the previous ones. By computer simulations, we examine the performance of the Q-learning and the fuzzy Q-learning by comparing them with the optimal choice for the previous actions. Simulation results clearly demonstrate that the fuzzy Q-learning outperforms the Q-learning.

2. Formulation of a Repeated Game

In this section, we formulate a transportation problem with many agents and several markets as a multi-player non-cooperative repeated game.

1) Player of Game: i ($i = 1,2,...,n$)

Each agent (*i.e.*, game player) is indexed by i. We assume that n agents are involved in our game (*i.e.*, $i = 1,2,...,n$).

2) Period of Game: t ($t = 1,2,...,T$)

The number of iterations of our game is indexed by t. We assume that our game is iterated T times (*i.e.*, $t = 1,2,...,T$).

3) Market: j ($j = 1,2,...,m$)

Each market is indexed by j. We assume that m markets

are given in our game (*i.e.*, $j = 1,2,\ldots,m$). In Figure 1, we show an example of our game where 100 agents and five markets are given.

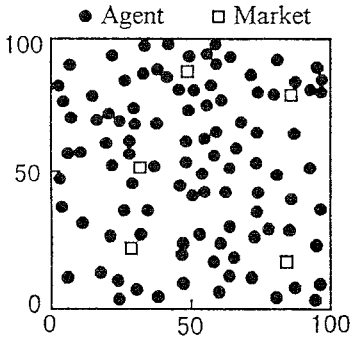

Figure 1. An example of our game.

4) Action: x_{ij}^t

We assume that each agent has a single product at each iteration of our game. Each agent is supposed to choose one market where his product is sold at the market price. The action of the i-th agent at the t-th iteration is to choose one market. Let us denote the action of the i-th agent at the t-th iteration of our game by x_{ij}^t, $j = 1,2,\ldots,m$, $t = 1,2,\ldots,T$, where

$$x_{ij}^t = \begin{cases} 1, & \text{if the } i\text{-th agent chooses the } j\text{-th market,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Because each agent is supposed to choose a single market from the given m markets for selling his product, the following relation holds:

$$\sum_{j=1}^{m} x_{ij}^t = 1 \quad \text{for } i = 1,2,\ldots,n;\ t = 1,2,\ldots,T. \quad (2)$$

5) Market Price: p_j^t

The total amount of products that are sold in the j-th market at the t-th iteration is calculated from (1) as follows:

$$X_j^t = \sum_{i=1}^{n} x_{ij}^t \quad \text{for } t = 1,2,\cdots,T, \quad (3)$$

where X_j^t is the total amount of products that are sold in the j-th market at the t-th iteration. We assume that the market price is determined by the following linear demand-supply relation:

$$p_j^t = a_j - b_j \cdot X_j^t \quad \text{for } t = 1,2,\cdots,T, \quad (4)$$

where p_j^t is the price in the j-th market at the t-th iteration, and a_j and b_j are positive constants that specify the demand-supply relation in the j-th market.

6) Transportation Cost: c_{ij}

We assume that the cost c_{ij} for the transportation of the product from the i-th agent to the j-th market depends on the distance between the agent and the market. Let us denote the distance between the i-th agent and the j-th market by d_{ij}. We assume that the transportation cost c_{ij} is given as follows:

$$c_{ij} = c \cdot d_{ij}, \quad (5)$$

where c is the transportation cost for the unit distance.

7) Profit of Agent: r_i^t

Let us denote the profit (*i.e.*, reward) of the i-th agent at the t-th iteration by r_i^t. We define the profit r_i^t as follows when the i-th agent chooses the j-th market for selling his product (*i.e.*, when $x_{ij}^t = 1$):

$$r_i^t = p_j^t - c_{ij}. \quad (6)$$

This can be rewritten from (1)-(5) as

$$r_i^t = \sum_{j=1}^{m} x_{ij}^t (p_j^t - c_{ij})$$

$$= \sum_{j=1}^{m} x_{ij}^t \left(a_j - b_j \cdot \sum_{k=1}^{n} x_{kj}^t - c \cdot d_{ij}\right). \quad (7)$$

Thus we can see that our game is a kind of quadratic programming problem. We can also see from (7) that the profit r_i^t depends on the actions of the other agents.

We assume that the aim of the i-th agent in our game is to maximize the total profit r_i over T iterations:

$$r_i = \sum_{t=1}^{T} r_i^t. \quad (8)$$

In our game, each agent chooses a single market for selling his product at each iteration (*i.e.*, at each t) in order to maximize his own profit. It should be noted that all the agents make such market selection at each iteration of our game. Thus our game is a multi-player non-cooperative repeated game. It should be also noted that the profit of each agent can not be represented by a simple payoff matrix because (i) many agents are involved in our game and (ii) the profit of each agent is determined through the demand-supply relation of each market in (4).

3. Strategies for Market Selection

3.1 *Q*-learning

When choosing a market for the t-th iteration, each agent has the complete information about the actions of all the other agents at the previous iteration (*i.e.*, $(t-1)$-th iteration) while he has no information about the actions at

the current iteration (*i.e.*, t-th iteration). The total number of possible combinations of the actions of the n agents at each iteration of the game is m^n because each agent chooses one from the m markets. This is intractably huge when many agents are involved in our game. For example, in the case of Figure 1 with 100 agents (*i.e.*, $n = 100$) and five markets (*i.e.*, $m = 5$), $m^n = 5^{100} \cong 7.9 \times 10^{69}$. Therefore the previous actions of all agents can not be used as state variables.

The simplest way for applying the Q-learning to our game is to use it with no state variables. A Q-value of each agent is assigned to each action. That is, a Q-value is assigned to each market. Let us denote the Q-value of the i-th agent assigned to the j-th market at the t-th iteration by Q_{ij}^t. This Q-value corresponds to the action "the i-th agent chooses the j-th market at the t-th iteration of the game." The Q-value is updated as follows:

$$Q_{ij}^t = \begin{cases} (1-\alpha) \cdot Q_{ij}^{t-1} + \alpha \cdot r_i^t, & \text{if } x_{ij}^t = 1, \\ Q_{ij}^{t-1}, & \text{otherwise,} \end{cases} \tag{9}$$

where α is a positive learning rate. From (9), we can see that the Q-value of the i-th agent for the j-th market is updated only when the j-th market is selected. It should be noted that all the Q-values are initialized to a prespecified value before the game.

The market selection by the i-th agent is done based on the Q-values for the m markets. Let $\Pr(x_{ij}^t = 1)$ be the probability that the i-th agent chooses the j-th market at the t-th iteration of our game. We define $\Pr(x_{ij}^t = 1)$ by the roulette wheel selection mechanism with the linear scaling [3] as follows:

$$\Pr(x_{ij}^t = 1) = \frac{Q_{ij}^t - \min\{Q_{ij}^t\}}{\sum_{j=1}^{m} (Q_{ij}^t - \min\{Q_{ij}^t\})}, \tag{10}$$

where $\min\{Q_{ij}^t\} = \min\{Q_{ij}^t \mid j = 1,2,...,m\}$. At each iteration of our game, each market is selected with the probability in (10). It should be noted that the relation in (2) holds by this market selection because each agent chooses only one market at each iteration.

3.2 Fuzzy Q-learning

As we have already mentioned, we can not use the previous actions of all agents as state variables because the number of possible combinations of previous actions is intractably huge. On the contrary, the number of markets is relatively small. Thus we can use the previous market prices as state variables. Because the market prices are continuous variables, we have to divide the domain of the market prices into several sub-domains. For such a partition of the continuous domain, we employ a fuzzy partition. In computer simulations, we use two fuzzy sets "*low*" and "*high*" in Figure 2 for dividing the continuous domain of the market price of each market.

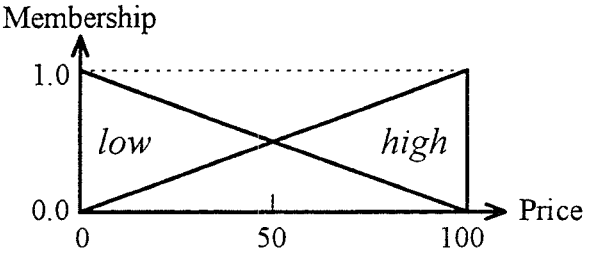

Figure 2. Fuzzy partition of the market price.

The Q-value of the i-th agent for the j-th market at the t-th iteration is inferred from the previous market prices p_j^{t-1}, $j = 1,2,...,m$, using the following fuzzy if-then rules:

$$\text{Rule } R_s\text{: If } p_1^{t-1} \text{ is } A_{s1} \text{ and } ... \text{ and } p_m^{t-1} \text{ is } A_{sm}$$
$$\text{then } Q_{i1}^t = q_{si1}^t \text{ and } ... \text{ and } Q_{im}^t = q_{sim}^t,$$
$$s = 1,2,...,N, \tag{11}$$

where R_s is the label of the fuzzy if-then rule, s is the rule index, A_{sj} is an antecedent fuzzy set such as "*low*" and "*high*," q_{sij}^t is a consequent real number, and N is the number of fuzzy if-then rules. When we have the two linguistic values in Figure 2 as antecedent fuzzy sets for each of the m markets, the number of fuzzy if-then rules is $N = 2^m$. In the case of Figure 1 with 100 agents (*i.e.*, $n = 100$) and five markets (*i.e.*, $m = 5$), $N = 2^5 = 32$.

The Q-value of the i-th agent for the j-th market at the t-th iteration is calculated by the fuzzy reasoning based on the N fuzzy if-then rules in (11). First let us define the compatibility of the previous market prices $\mathbf{p}^{t-1} = (p_1^{t-1}, p_2^{t-1}, ..., p_m^{t-1})$ with the fuzzy if-then rule R_s by the product operator as follows:

$$\mu_s(\mathbf{p}^{t-1}) = A_{s1}(p_1^{t-1}) \cdot A_{s2}(p_2^{t-1}) \cdot ... \cdot A_{sm}(p_m^{t-1}), \tag{12}$$

where $A_{sj}(\cdot)$ denotes the membership function of the antecedent fuzzy set A_{sj}. The Q-value of the i-th agent for the j-th market at the t-th iteration is calculated as follows:

$$Q_{ij}^t = \frac{\sum_{s=1}^{N} \mu_s(\mathbf{p}^{t-1}) \cdot q_{sij}^t}{\sum_{s=1}^{N} \mu_s(\mathbf{p}^{t-1})}. \quad (13)$$

The consequent q_{sij}^t of each fuzzy if-then rule is adjusted as follows:

$$q_{sij}^t = \begin{cases} \{1 - \alpha \cdot \frac{\mu_s(\mathbf{p}^{t-1})}{\sum_{s=1}^{N} \mu_s(\mathbf{p}^{t-1})}\} \cdot q_{sij}^{t-1} \\ \quad + \alpha \cdot \frac{\mu_s(\mathbf{p}^{t-1})}{\sum_{s=1}^{N} \mu_s(\mathbf{p}^{t-1})} \cdot r_{ij}^t, \text{ if } x_{ij}^t = 1, \\ q_{sij}^{t-1}, \text{ otherwise.} \end{cases} \quad (14)$$

From the comparison between (9) and (14), we can see that the amount of the modification of the consequent q_{sij}^t is proportional to $\mu_s(\mathbf{p}^{t-1}) / \sum \mu_s(\mathbf{p}^{t-1})$. The same learning procedure of each fuzzy if-then rule was used in Horiuchi *et al.*[10].

In the fuzzy Q-learning, the market selection is done in the same manner as in the Q-learning in Subsection 3.1. That is, the market selection is done according to the probability $\Pr(x_{ij}^t = 1)$ in (10).

3.3 Other Strategies

For evaluating the Q-learning and the fuzzy Q-learning, we compare them with other strategies. The most simple strategy is the random strategy. The random strategy can be obtained by specifying the market selection probability $\Pr(x_{ij}^t = 1)$ as

$$\Pr(x_{ij}^t = 1) = 1/m. \quad (15)$$

That is, each of the m market is randomly selected with the same probability.

In the Q-learning and the fuzzy Q-learning, each agent did not use any information about the demand-supply relation of each market. That is, each agent used only the current profit r_i^t in the Q-learning, and the current profit r_i^t and the previous market prices $\mathbf{p}^{t-1} = (p_1^{t-1}, p_2^{t-1}, \ldots, p_m^{t-1})$ in the fuzzy Q-learning. If the information about the demand-supply relation of each market is available, each agent can use a more complicated strategy. From the definition of the current profit r_i^t in (7), we can see that the optimal selection can be done when the i-th agent knows (i) the actions of all the other agents, (ii) the demand-supply relations of all the m markets (*i.e.*, a_j and b_j for all the m markets), and (iii) the transportation cost from the i-th agent to each market (*i.e.*, c and d_{ij} for all the m markets). Because the market selection at each iteration of our game has to be done simultaneously by all the n agents, each agent does not know the current actions of the other agents when he chooses a market. Thus let us consider the optimal market selection strategy for the previous actions. In this strategy, the optimal market is selected for the previous actions. That is, the i-th agent selects the best market by assuming that the other agents choose exactly the same markets as in the previous iteration. Of course, this assumption is not always valid. Thus the optimal market selection strategy for the previous actions is not always optimal for the current actions. It should be noted that every agent can not know the current actions of the other agents before the market selection. Thus the optimal market selection for the current actions is impossible.

4. Computer Simulations

4.1 Specifications of Game

We applied the four strategies in Section 3 (*i.e.*, the Q-learning, the fuzzy Q-learning, the random strategy, and the optimal strategy for the previous actions) to the transportation problem in Figure 1. Our game was iterated 500 times (*i.e.*, $t = 1,2,\cdots,500$). We used the same demand-supply relation for all the five markets:

$$p_j^t = 100 - b \cdot X_j^t. \quad (16)$$

As the value of b, we examined 11 cases: $b = 0.0$, 0.3, 0.6, ..., 3.0.

As the distance d_{ij} between the i-th agent and the j-th market, we used the Euclidean distance. The transportation cost for the unit distance is specified as $c = 1.0$. Thus the transportation cost from the agent to the market is exactly the same as the Euclidean distance in Figure 1.

In the Q-learning, the initial value of Q_{ij}^t was specified as $Q_{ij}^0 = 100$ for all agents and all markets. In the same manner, the initial value of q_{sij}^t was specified as $q_{sij}^0 = 100$ for all fuzzy if-then rules in the fuzzy Q-learning. For the fuzzy partition, we used the two linguistic values in Figure 2 in the fuzzy Q-learning. Thus each Q-value was calculated by $2^5 = 32$ fuzzy if-then rules because we have five markets. For comparison, we also examined the crisp partition in Figure 3 in the fuzzy Q-learning. In this case, the fuzzy Q-learning is equivalent to the Q-learning with 32 discrete states for the previous market prices. We specified the value

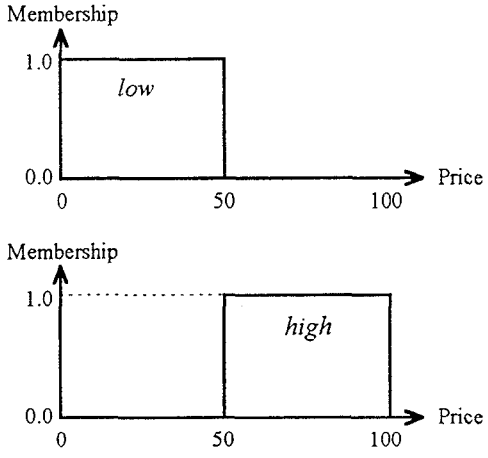

Figure 3. Crisp partition.

of α (*i.e.*, learning rate) as $\alpha = 0.9$ for the Q-learning and the fuzzy Q-learning.

In the optimal strategy for the previous actions, the initial action was selected by assuming $x_{ij}^0 = 0$ for $i = 1,2,\cdots,n$ and $j = 1,2,\cdots,m$.

4.2 Simulation Results

First we used each strategy for all the 100 agents. That is, all the 100 agents employed the same strategy through the 500 iterations of our game. In computer simulations, independent 10 trials were performed for each strategy for each value of b. Simulation results were summarized in Figure 4 for the Q-learning and the optimal strategy for the previous actions, and in Figure 5 for the two versions of the fuzzy Q-learning. For comparison, the simulation results by the random strategy are shown in both figures. From Figure 4, we can see that the average profit by the Q-learning over 100 agents is smaller than that by the optimal strategy for the previous actions when the value of b is not high (*i.e.*, $0.0 \le b \le 2.1$). On the contrary, when the value of b is high (*i.e.*, $2.4 \le b \le 3.0$), the average profit by the Q-learning is larger than that by the optimal strategy for the previous actions. From Figure 5, we can see that the fuzzy partition in Figure 2 outperforms the crisp partition in Figure 3.

Next we examined the performance of the Q-learning and the fuzzy Q-learning by the competition with the optimal strategy for the previous actions. In our computer simulations, one agent employed the Q-learning (or fuzzy Q-learning) and the other 99 agents employed the optimal strategy for the previous actions. This computer simulation was performed 100 times for each value of b such that each of the 100 agents was examined as the Q-learning (or fuzzy Q-learning) agent just once. Computer simulations were summarized in Figure 6 ~ Figure 8. From these figures, we can see that both the Q-learning and the fuzzy Q-learning

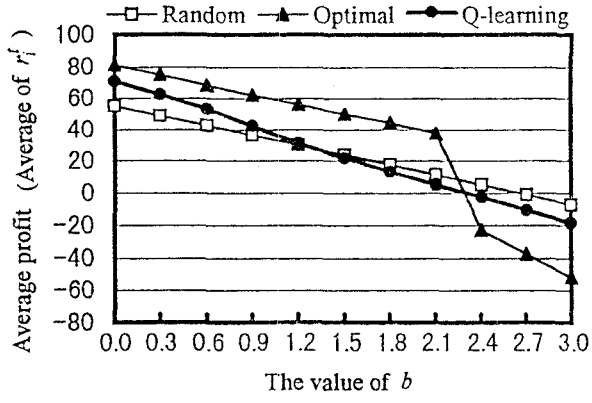

Figure 4. Simulation results by the random strategy, the Q-learning, and the optimal strategy for the previous actions.

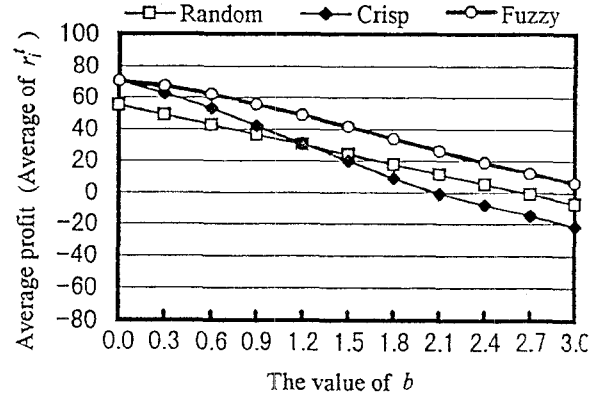

Figure 5. Simulation results by the random strategy and the two versions of the fuzzy Q-learning (the fuzzy partition in Figure 2 and the crisp partition in Figure 3).

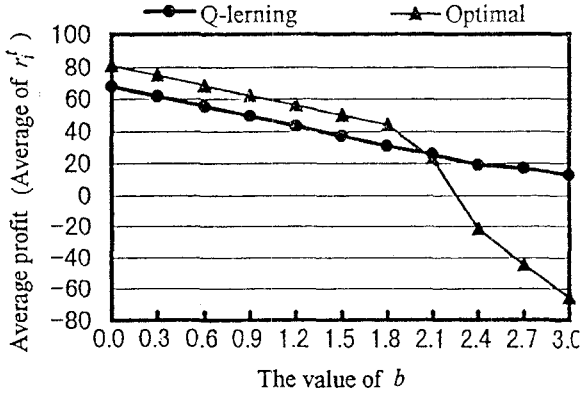

Figure 6. Simulation results by the competitior between the Q-learning and the optimal strategy for the previous actions.

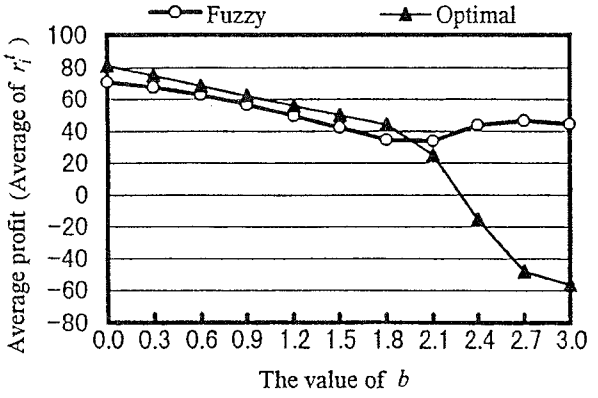

Figure 7. Simulation results by the competition between the fuzzy Q-learning with the fuzzy partition and the optimal strategy for the previous actions.

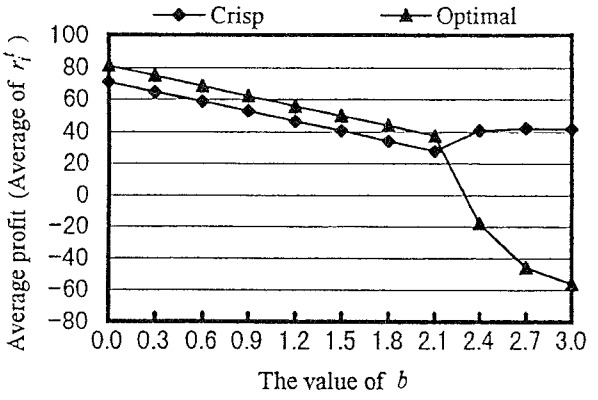

Figure 8. Simulation results by the competition between the fuzzy Q-learning with the crisp partition and the optimal strategy for the previous actions.

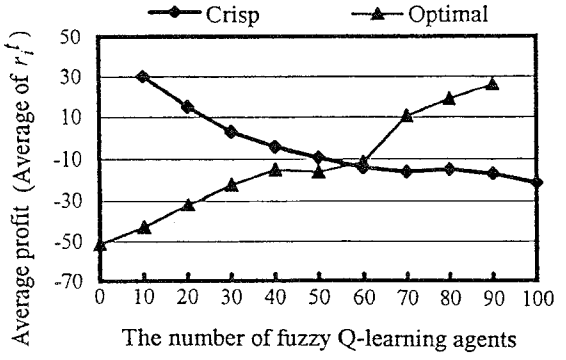

Figure 9. Simulation results by the competition between the fuzzy Q-learning with the crisp partition and the optimal strategy for the previous actions.

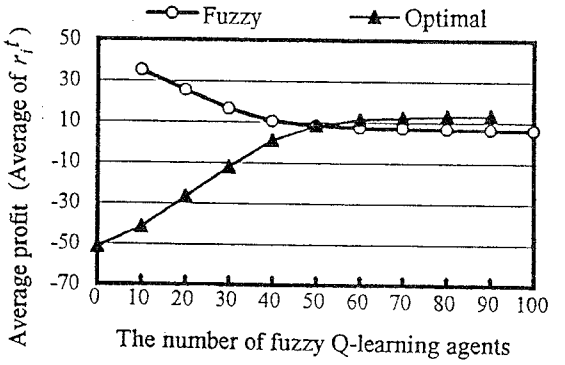

Figure 10. Simulation results by the competition between the fuzzy Q-learning with the fuzzy partition and the optimal strategy for the previous actions.

can not outperform the optimal strategy for the previous actions when the value of b is low (*i.e.*, $0.0 \leq b \leq 1.8$). On the contrary, they outperform the optimal strategy for the previous actions when the value of b is high.

We also performed similar computer simulations by varying the number of fuzzy Q-learning agents. The value of b was specified as $b = 3.0$. As the number of fuzzy Q-learning agents, we used 11 values: 0, 10, 20, ..., 100. For each specification of the number of fuzzy Q-learning agents, we performed 20 independent trials by randomly selecting the fuzzy Q-learning agents. Simulation results were summarized in Figure 9 and Figure 10. From these figures, we can see that better results were obtained by the optimal strategy for the previous actions when the number of fuzzy Q-learning agents is large (*i.e.*, when the number of the optimal strategy agents is small). On the contrary, when the number of fuzzy Q-learning agents is small, the average profit by the fuzzy Q-learning was larger than that by the optimal strategy for the previous actions. We can also see that the performance of the fuzzy Q-learning with the fuzzy partition was higher than that of the fuzzy Q-learning with the crisp partition.

4.3 Discussion

From Figure 4 and Figure 6 ~ Figure 8, we can see that the optimal strategy for the previous actions performed very well when the value of b is low, but it did not perform well when the value of b is high. This is because each agent tends to choose the market with the highest previous price in the optimal strategy for the previous actions. This leads to the concentration of the products at that market. From the linear demand-supply relation in (4), the concentration of the products and the higher value of b make the market price very low. Thus the optimal strategy for the previous actions

did not work well when the value of b was high. The effect of such concentration of products was slight when the number of the optimal strategy agents was small. Thus the optimal strategy for the previous actions worked well when the number of fuzzy Q-learning agents was large (see Figure 9 and Figure 10).

The random strategy evenly distributes the products among the given markets on the average. Thus such concentration of products never happens when all the agents employ the random strategy. This is the reason why the random strategy relatively worked well in Figure 4 and Figure 5 when the value of b was large.

The fuzzy Q-learning with the fuzzy partition in Figure 2 worked well for various situations (see Figure 5, Figure 7 and Figure 10). This is because the previous market prices were effectively utilized for determining the Q-values by fuzzy if-then rules.

Some of the 32 fuzzy if-then rules obtained by the fuzzy Q-learning with the fuzzy partition in Figure 2 are shown in Table 1 for the case of $b = 3.0$ and a single fuzzy Q-learning agent (the other 99 agents used the optimal strategy for the previous actions). These fuzzy if-then rules in Table 1 are typical fuzzy if-then rules among all the 32 rules. From this table, we can see that the Q-values are high for the markets with low previous market prices. Those fuzzy if-then rules suggest the product concentration at the current iteration by the other 99 agents with the optimal strategy for the previous actions. It should be noted that those rules were automatically extracted by the fuzzy Q-learning.

5. Conclusion

In this paper, we first formulated a transportation problem with the demand-supply relation at each market as a multi-player non-cooperative repeated game. The characteristic features of our game compared with the iterated Prisoner's Dilemma game are (i) many agents are involved and (ii) the profit of each agent is not represented by a simple payoff matrix. Next we illustrated how the Q-learning and the fuzzy Q-learning can be applied to our game. Finally, we demonstrated by computer simulations that the fuzzy Q-learning worked well in comparison with the Q-learniing, the random strategy, and the optimal strategy for the previous actions.

References

[1] R. Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[3] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, 1989.

[4] R. Axelrod, "The evolution of strategies in the iterated Prisoner's Dilemma," in L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, pp. 32-41, Morgan Kaufmann, Pitman, London, 1987.

[5] K. Lindgren, "Evolutionary Phenomena in Simple Dynamics," in C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen (eds.), *Artificial Life II*, pp. 295-312, Addison-Wesley, Reading, 1991.

[6] A. Ito and H. Yano, "The emergence of cooperation in a society of autonomous agents – the Prisoner's Dilemma game under the disclosure of contract histories –," *Proc. 1st International Conference on Multi-Agent Systems* (June 12-14, 1995, San Francisco, USA) pp. 201-208.

[7] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, Vol. 8, pp. 279-292, 1992.

[8] P.Y. Glorennec, "Fuzzy Q-learning and dynamical fuzzy Q-learning," *Proc. of 3th International Conference on Fuzzy Systems* (June 26-29, 1994, Orlando, USA) pp. 474-479.

[9] H.R.Berenji, fuzzy Q-learning: a new approach for fuzzy dynamic programming problems," *Proc. of 3th International Conference on Fuzzy Systems* (June 26-29, 1994, Orlando, USA) pp. 486-491.

[10] T. Horiuchi, A. Fujino, O. Katai and T. Sawaragi, "Fuzzy interpolation-based Q-learning with continuous states and actions," *Proc. 5th International Conference on Fuzzy Systems* (September 8-11, 1996, New Orleans, USA) pp. 594-600.

[11] L. Jouffe and P. -Y. Glorennec, "Comparison between connectionist and fuzzy Q-learning," *Proc. 4th International Conference on Soft Computing* (September 30 - October 5, 1996, Iizuka, Japan) pp. 557-560.

Table 1. Some examples of fuzzy if-then rules generated by the fuzzy Q-learning.

Antecedent: Market price					Consequent: Q-value				
p_1^{t-1}	p_2^{t-1}	p_3^{t-1}	p_4^{t-1}	p_5^{t-1}	q_{si1}^t	q_{si2}^t	q_{si3}^t	q_{si4}^t	q_{si5}^t
low	*high*	*low*	*high*	*low*	36.3	-30.7	54.0	-76.8	75.9
high	*low*	*high*	*low*	*high*	-34.2	72.7	-52.1	62.5	-26.1
high	*high*	*low*	*high*	*low*	36.3	-31.3	54.0	54.3	75.9