14

Objective-based Reinforcement Learning System for Cooperative Behavior Acquisition

Kunikazu Kobayashi, Koji Nakano, Takashi Kuremoto
and Masanao Obayashi
Yamaguchi University
Japan

1. Introduction

This chapter discusses the emergence of cooperative behavior in multiagent system and presents an objective-based reinforcement learning system in order to acquire cooperative behavior.
Reinforcement learning is a method that agents will acquire the optimum behavior by trial and error by being given rewards in an environment as a compensation for its behavior (Kaelbling et al., 1996; Sutton & Barto, 1998; Weber et al. 2008). Most of studies on reinforcement learning have been conducted for single agent learning in a static environment. The Q-learning which is a typical learning method is proved that it converges to an optimum solution for Markov decision process (MDP) (Watkins & Dayan, 1992). However, in a multiagent environment, as plural agents' behavior may affect the state transition, the environment is generally considered as non Markov decision process (non-MDP), and we must face critical problems whether it is possible to solve (Stone & Veloso, 2000).
On the above problems in a multiagent environment, Arai et al. have compared Q-learning with profit sharing (PS) (Grefenstette, 1988) using the pursuit problem in a grid environment (Arai et al., 1997). As a result, Q-learning has instability for learning because it uses Q values of the transited state in an updating equation. However, PS can absorb the uncertainty of the state transition because of cumulative discounted reward. Therefore, they concluded that PS is more suitable than Q-learning in the multiagent environment (Arai et al., 1997; Miyazaki & Kobayashi, 1998). Uchibe et al. have presented the capability of learning in a multiagent environment since relation between actions of a learner and the others is estimated as a local prediction model (Uchibe et al., 2002). However, PS has a problem of inadequate convergence because PS reinforces all the pairs of a state and an action irrespective of the achievement of a purpose (Nakano et al., 2005).
This chapter presents an objective-based reinforcement learning system for multiple autonomous mobile robots to solve the above problem and to emerge cooperative behavior (Kobayashi et al, 2007). The proposed system basically employs PS as a learning method but a PS table, which is used for PS learning, is divided into two kinds of PS tables to solve the above problem. One is to learn cooperative behavior using information on other agents' positions and the other is to learn how to control basic movements. Through computer

simulation and real robot experiment using a garbage collecting problem, the performance of the proposed system is evaluated. As a result, it is verified that agents select the most available garbage for cooperative behavior using visual information in an unknown environment and move to the target avoiding obstacles.
This chapter is organized as follows. At first, an outline of reinforcement learning is described. Next, an objective-based reinforcement learning system for multiple autonomous mobile robots is presented. After that, the performance of the proposed system is evaluated through both computer simulation and real robot experiment. Finally, this chapter is concluded with a discussion and future work.

2. Reinforcement learning

Reinforcement learning is a method that agents will acquire the optimum behavior by trial and error by being given rewards in an environment as a compensation for its behavior (Kaelbling et al., 1996; Sutton & Barto, 1998; Weber et al. 2008). It is a kind of unsupervised learning, which does not require direct teacher signals. It is originally modeled by conditional response of animals, where they tend to cause a specific action by giving reward, i.e. food or water, when and only when they cause a specific action for a cue.
Reinforcement learning can be classified into two categories, i.e. exploration oriented and exploitation oriented learning (Yamamura et al., 1995). In the exploration oriented learning, it will evaluate the action at each state as estimating an environment. On the other hand, in the exploitation oriented learning, it will propagate the evaluation obtained at a state into all the states and actions to reach the state. The representatives of exploration oriented and exploitation oriented learning are Q-learning (Watkins & Dayan, 1992) and PS (Grefenstette, 1988), respectively.

2.1 Q-learning

The Q-learning is guaranteed that every state will converge to the optimal solution by appropriately adjusting a learning rate in the MDP environment (Watkins & Dayan, 1992). The state-action value function is denoted by $Q(s,a)$ and updated so as to take the optimal action by exploring it in a learning space. The following is the algorithm of Q-learning.

<Algorithm: Q-learning>

Step 1. Initialize $Q(s,a)$ for any state $s \in S$ and action $a \in A$, where S means the set of all the states and A is the set of all the possible actions.
Step 2. Initialize s.
Step 3. Choose action a based on a policy, such as greedy policy, Boltzmann policy and so on.
Step 4. Take action a, obtain reward r and then transit the next state s'.
Step 5. Update $Q(s,a)$ by equation (1).

$$Q(s,a) \leftarrow Q(s,a) + \alpha\{r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\}, \quad (1)$$

where α denotes a learning rate, $0 < \alpha \leq 1$ and γ is a discount rate.
Step 6. Repeat from Step 2 to 5 until s reaches a terminal.

2.2 Profit sharing (PS)

The PS defines a rule as a pair of state and action, and reinforces a series of rules when it gets reward (Grefenstette, 1988). Then, it distributes reward into a weight of rule, $w(s,a)$. As described above, the PS empirically conducts learning through updating $w(s,a)$. The PS is guaranteed that it could obtain a rational policy (Miyazaki et al., 1994). A geometrically decreasing function is known as one of the most simple reinforcement function which satisfies the rationality theorem. The PS reinforces all the rules at the end of an episode, which is defined by trails from an initial state to the goal state. Therefore, it could reinforce many rules using only one reward. The following is the algorithm of PS.

<Algorithm: Profit sharing>

Step 1. Initialize $w(s,a)$ for any state $s \in S$ and action $a \in A$, where S means the set of all the states and A is the set of all the possible actions.

Step 2. Initialize s.

Step 3. Choose action a based on a policy, such as greedy policy, Boltzmann policy and so on.

Step 4. Take action a, obtain reward r, then reserve rule $\{s,a\}$ and reward r, and then transit the next state s'.

Step 5. If $r \neq 0$, then $w(s,a)$ is updated by equation (2).

$$W(s_t,a_t) \leftarrow W(s_t,a_t) + f(t,r), \tag{2}$$

where $f(t,r)$ denotes a reinforcement function and t is time, $t = 0,1,\cdots,T_G$, T_G means time at the goal state.

Step 6. Repeat from Step 2 to 5 until s reaches a terminal.

3. Objective-based reinforcement learning system

3.1 Architecture

This chapter presents an objective-based reinforcement learning system as illustrated in Fig. 1. The proposed system is composed of three parts; an action controller, a learning controller and an evaluator. The feature of the system is to divide behavior of an agent into cooperative and basic behavior to learn separately. The learning of cooperative behavior is using information of the other agents' positions and the present state. The learning of basic behavior is to learn how to control own basic behavior such as *go forward* or *turn right*.

In a general learning method, when an agent acquires a reward it can hardly estimates own action whether it can cooperate or not. To solve this problem, the proposed system divides behavior into two kinds of behavior and each one is evaluated using different criteria. Dividing behavior into the above two kinds of behavior results in the following two merits.

- It could prevent degrading learning efficiency due to mutual interference between both learning.
- It could promote fast convergence due to clarification of both learning.

For action control and evaluation, the proposed method employs individual groups of action and evaluation corresponding to cooperative and basic behavior, respectively.

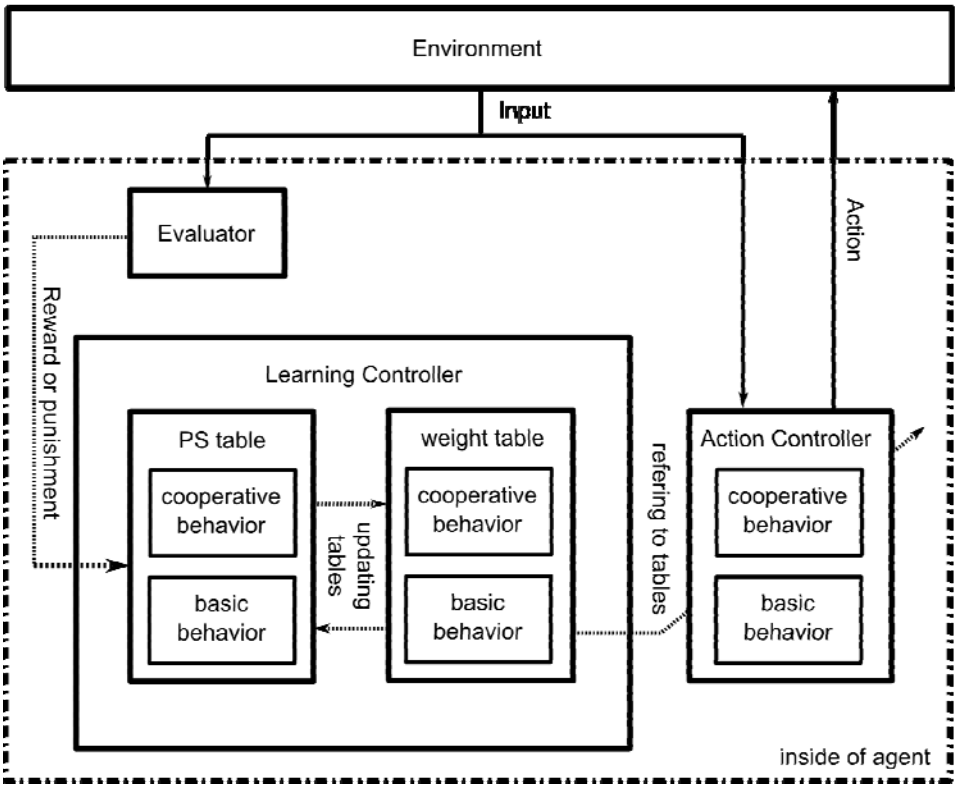


Fig. 1. Architecture of the proposed system.

3.2 Action controller

In the proposed system, an action is selected by the weight of rules, which is produced by inputs from an environment and groups of action. The action selection is conducted by the Boltzmann distribution. It is described by the weight $w(s,a)$ of rules created by the pairs of a state s and an action a and defined as equation (3).

$$B(a \mid s) = \frac{e^{w(s,a)/T}}{\sum_{b \in A} e^{w(s,b)/T}}, \tag{3}$$

where $B(a \mid s)$ is a probability selecting action a at state s, T is a positive temperature constant and A is a set of available actions.

The Boltzmann method is one of soft-max action selection methods and makes the probability of action change so as to preferentially select the action with large weight (Sutton & Barto, 1998). If T becomes large, all the possible actions tend to occur equally. On the other hand, If T becomes small, the action with the largest weight value tend to be selected because the difference of selection probability is enhanced. Then, if the extreme case $T \to 0$, the action selection becomes greedy.

3.3 Learning controller

The PS is employed as a learning method for an agent. The weight $w(s,a)$ of rules is updated by

$$w(s,a) = w(s,a) + f(t,r), \tag{4}$$

where t is a time, r is a reward and $f(\cdot,\cdot)$ is a reinforcement function. In this chapter, the following function is used as function f.

$$f(t,r) = r\gamma^{t_G - t}, \tag{5}$$

where γ is a decay rate and t_G is a time in the goal state. Equation (5) satisfies the rationality theorem of PS which guarantees successful convergence (Miyazaki et al., 1994).
In the proposed system, two PS tables are prepared to cooperative and basic behavior learning. These two tables are separated and the weights are updated independently.

3.4 Evaluator

The different criteria are prepared for cooperative and basic behavior. This is because one can judge whether success and failure of agent's behavior come from cooperative behavior or basic behavior.

4. Experiment

The proposed system was applied to a garbage collecting problem which is one of the standard multiagent tasks (Ishiguro et al, 1997). The two kinds of experiments, i.e. computer simulation and real robot experiment, were conducted to evaluate the performance of the proposed system.
In computer simulation, it is confirmed whether cooperative behavior could be emerged using the garbage collecting problem with two agents. After that, it is evaluated that the values of parameters obtained in computer simulation could apply to real robot experiment. Finally, it is shown that the proposed system could be improved its performance through further learning in real robot experiment.

4.1 Experimental setting

In the experiment, Khepera robot as shown in Fig. 2 is used for an agent. The Khepera robot is a small-size robot, developed by K-Team[1] at the Microprocessor Systems Laboratory, Swiss Federal Institute of Technology and widely used for research development. It equips two wheels drived by two DC motors, a color CCD camera DCC-2010N and a gripper.
The robot identifies garbage and other robots by using an image captured by the color CCD camera. Then, the images are processed by an image processing board IP7000BD developed by Hitachi Information & Control Solutions, Ltd.

[1] URL http://www.k-team.com.

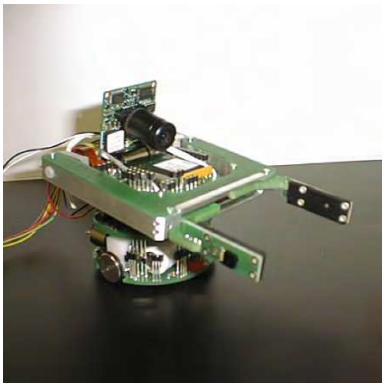
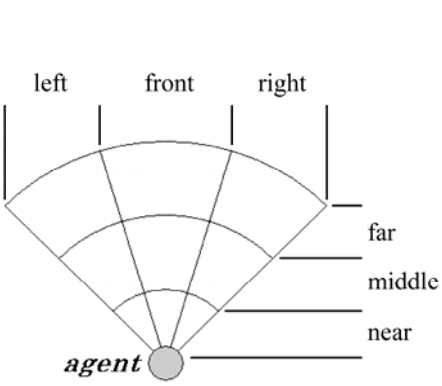


Fig. 2. Khepera robot equipped with a color CCD camera and a gripper.

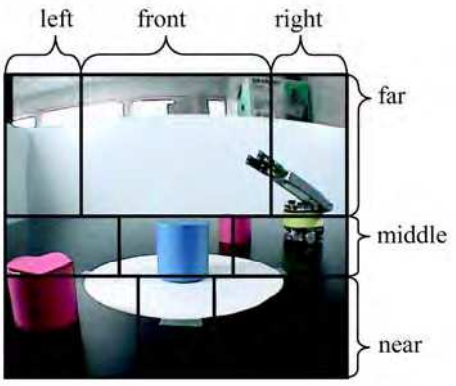
4.2 Problem setting

In an experimental field, there are two agents, some garbage and one garbage can, and then agents must collect all the garbage and take it to the garbage can.

As shown in Fig. 3, an input for the agent is classified into nine sub-states, combinations of three sorts of orientations (left, front or right) and three sorts of distances (near, middle or far). In computer simulation, an agent can perceive the front sector as shown in Fig. 3(a). In real robot experiment, an image captured by the color CCD camera is divided into nine areas as shown in Fig. 3(b). In Fig. 3(b), the far-front area is large because the robot tends to go forward, and the near-front area is small because the robot can grip the garbage.



(a) State classification in computer simulation



(b) State classification in real experiment

Fig. 3. State classification in computer simulation and in real experiment.

For cooperative behavior learning, an agent gets the relative coordinate of the other agent from visual information and selects the target garbage. For basic behavior learning, an agent gets the relative coordinate of the target garbage and information on obstacles and determines the direction of movement. Here, the obstacles refer to all the things except for

own agent and the target garbage, such as the other agent, non-target garbage, etc. To reduce the number of states for obstacles, it is focused only on the nearest obstacle.
For cooperative behavior learning, the number of states for the other agent including non-observable state is 10 and the number of pairs of rules is 9. Therefore, the total number of rules is 90. For basic behavior learning, the number of states is 9 for the relative coordinate of the target garbage and 8 for information on obstacles. So, the total number of rules is 216.
The reward and the decay rate for PS are defined as follows: $r = 2.0$ and $\gamma = 0.8$ for success and $r = -0.8$ and $\gamma = 0.6$ for failure.

The action of agents is evaluated using four kinds of criterion as shown in Table 1. In this table, the symbol 'o' means reward or punishment is considered and the symbol `x' is not considered.

Condition	Cooperative action	Basic action
Reward: an agent arrives at the target garbage or the garbage can.	o	o
Punishment: an agent decides the same garbage with other agents.	o	x
Punishment: an agent bumps obstacles.	x	o
Punishment: an agent loses the target.	x	o

Table 1. Definition of reward and punishment.

4.3 Computer simulation

A simulation field is a 21x21 grid world and there are ten garbage, two agents and one garbage can in the field as shown in Fig. 4.

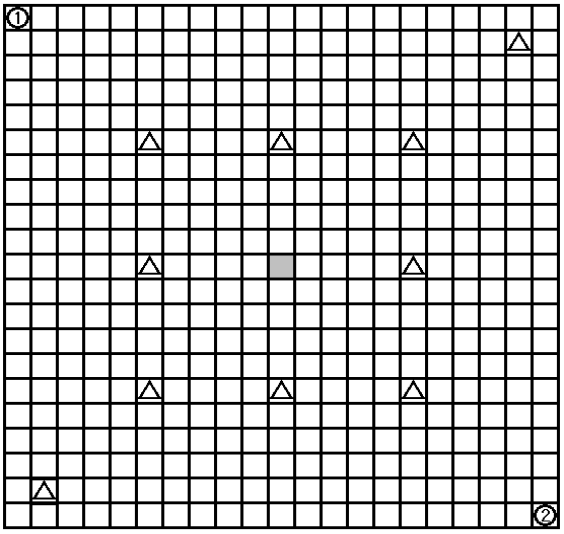


Fig. 4. Initial position of two agents denoted by circle, ten garbage by triangle and one garbage can by shaded square.

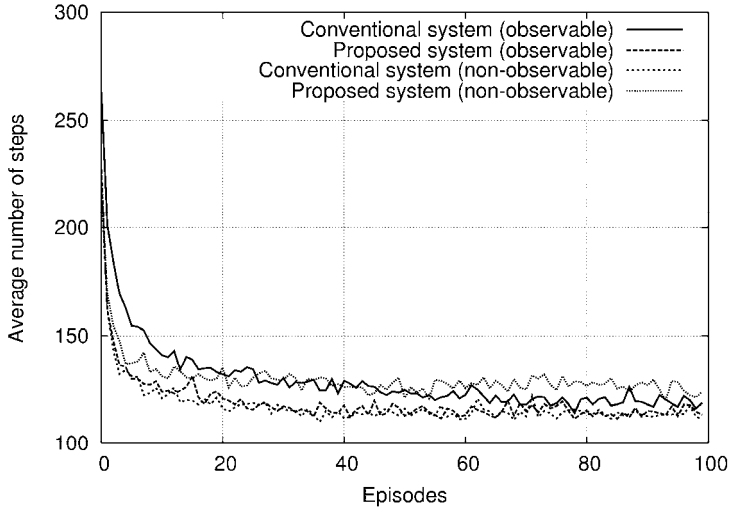
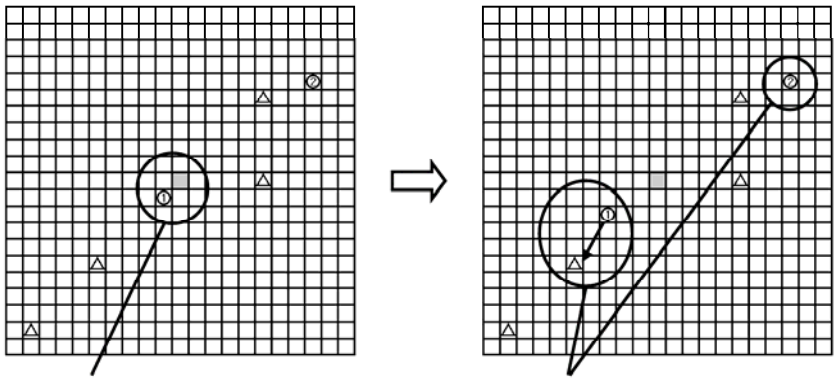


Fig. 5. Performance comparison of the proposed and conventional systems.

	Observable	Non-observable
Conventional method	118.7	111.1
Proposed method	113.3	123.8

Table 2. The average number of steps in the final trial.



(a) Agent 1 arrives at the trash can.

(b) Agent 1 moves to the garbage opposite to agent 2.

Fig. 6. An example of cooperative behavior acquired in the proposed system with observing the other agent.

One trial is defined as until all the garbage is collected, and 100 trials are considered as one episode. The number of average steps is calculated after repeating 100 episodes. At this time, $w(s,a)$ are initialized for each episode. To verify the effectiveness of the proposed system, it is compared with the standard PS system (conventional system).

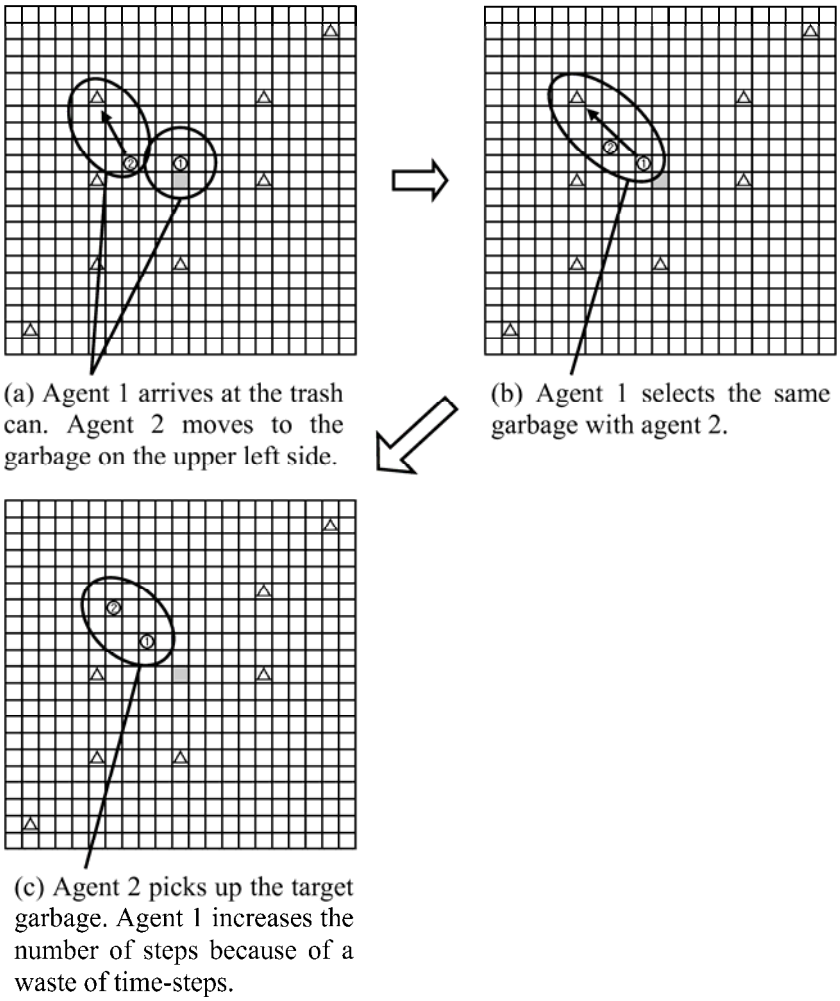


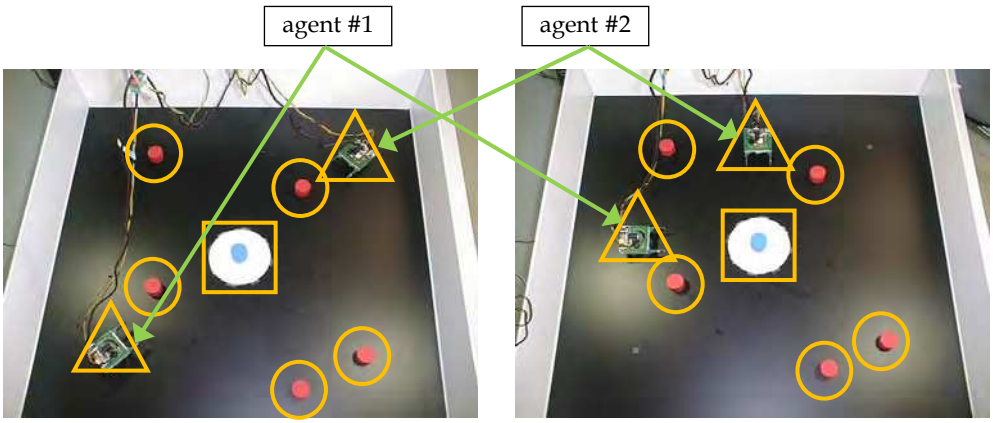
Fig. 7. An example of cooperative behavior acquired in the proposed system without observing the other agent.

Figure 5 and Table 2 show the result of the computer simulation. In the case that one agent can observe the other, the agent using the proposed system learns faster than the agent

using the conventional system. From this result, it is shown that the proposed system realizes cooperative behavior. However, when the agent is compared with the agent which is using the conventional system and do not observe the other agent, the performance of the proposed agent is similar to that of the conventional agent.
Figure 6 illustrates cooperative behavior observed in the experiment in which agent observes the other one. After agent 1 took the garbage to the garbage can (Fig. 6(a)), it does not select the garbage near agent 2 as the object, but another one opposite to agent 2 (Fig. 6(b)). Such behavior often occurred after learning with observing the other agents. On the other hand, Fig. 7 depicts cooperative behavior without observing the other agent. After agent 1 reached the garbage can (Fig. 7(a)), as it selected the garbage which are also targeted by agent 2 (Fig. 7(b)), it is clear that the number of steps is increased because of a waste of time-steps (Fig. 7(c)).

4.4 Real robot experiment

Two Khepera robots, an image processing board IP7000BD, a color CCD camera DCC-2010N and a robot control PC are used in the experiment. An experiment field is a 1[m]x1[m] square surrounded by white walls and there are five garbage, two robots and one garbage can as shown in Fig. 8.



(a) Arrangement #1 (b) Arrangement #2

Fig. 8. Initial position of two agents denoted by circle, five garbage by triangle and one garbage can by square.

The following three kinds of the experiments were conducted to evaluate the learning ability of the proposed system.

Exp. 1. The robots are controlled using the learned weights in the simulation, which are not updated during Exp. 1.
Exp. 2. The robots are controlled using the learned weights in the simulation, which are updated during Exp. 2.
Exp. 3. The robots are controlled using the learned weights in Exp. 2 after the initial position of robots is changed.

Table 3 shows that the number of average steps in Exp. 2 is decreased compared with that in Exp. 1. Thus, the learned weights in the simulation are available for the real robot environment, and furthermore, the proposed system can learn flexibly in real environment. On the other hand, the number of average steps in Exp. 3 is not increased compared with Exp. 2. Therefore, the weights learned in real environment are applicable to different environments, and this shows that the proposed system is robust.

	Average number of steps
Exp. 1	201.9
Exp. 2	178.2
Exp. 3	161.1

Table 3. The average number of steps in Exp. 1 to 3.

5. Conclusion

This chapter has proposed the objective-based reinforcement learning system for multiple autonomous mobile robots to acquire cooperative behavior. In the proposed system, robots select the most available target garbage for cooperative behavior using visual information in an unknown environment, and move to the target avoiding obstacles. The proposed system employs profit sharing (PS) and a characteristic of the system is using two kinds of PS tables. One is to learn cooperative behavior using information on other robot's positions, the other is to learn how to control basic movements. Through computer simulation and real robot experiment using a garbage collecting problem, it was verified that the proposed system is effective compared with the conventional system.
The future problem is to solve a trade-off between reducing the number of inputs and recognizing the external environment.

6. References

Arai, S.; Miyazaki, K. & Kobayashi, S. (1997). Generating Cooperative Behavior by Multi-Agent Reinforcement Learning, *Proceedings of the 6th European Workshop on Learning Robots (EWLR-6)*, pp.143-157.

Grefenstette, J. J. (1988). Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, *Machine Learning*, Vol.3, pp.225-245, ISSN: 0885-6125.

Ishiguro, A.; Watanabe, Y.; Kondo, T.; Shirai, Y. & Uchikawa, Y. (1997). Robot with a Decentralized Consensus-making Mechanism Based on the Immune System, *Proceedings of the International Symposium on Autonomous Decentralized Systems (ISADS'97)*, pp.231-237.

Kaelbling, L. P.; Littman, M. L. & Moore, A. P. (1996). Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, Vol.4, pp.237-285, ISSN: 11076-9757.

Kobayashi, K.; Nakano, K.; Kuremoto, T. & Obayashi, M., (2007). Cooperative Behavior Acquisition of Multiple Autonomous Mobile Robots by an Objective-based Reinforcement Learning System, *Proceedings of International Conference on Control, Automation and Systems (ICCAS2007)*, pp.777-780.

Miyazaki, K.; Yamamura, M. & Kobayashi, S. (1994). On the Rationality of Profit Sharing in Reinforcement Learning, *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing (Iizuka'94)*, pp.285-288.

Miyazaki, K. & Kobayashi, S. (1998). Learning Deterministic Policies in Partially Observable Markov Decision Processes, *Proceedings of International Conference on Intelligent Autonomous System (IAS-5)*, pp.250-257.

Nakano, K.; Obayashi, M.; Kobayashi, K. & Kuremoto, T., (2005). Cooperative Behavior Acquisition for Multiple Autonomous Mobile Robots, *Proceedings of the Tenth International Symposium on Artificial Life and Robotics (AROB2005)*, CD-ROM.

Stone, P. & Veloso, M. (2000). Multiagent systems: A Survey From a Machine Learning Perspective, *Autonomous Robots*, Vol.8, No.3, pp.345-383, ISSN: 0929-5593.

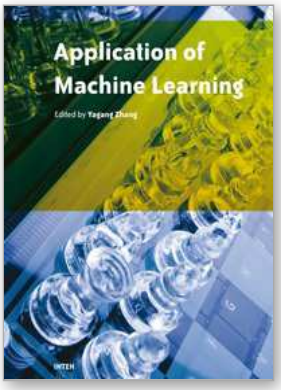
Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, MIT Press, ISBN: 978-0-262-19398-6, Cambridge, UK.

Uchibe, E.; Asada, M. & Hosoda, K. (2002). State Space Construction for Cooperative Behavior Acquisition in the Environments Including Multiple Learning Robots, *Journal of the Robotics Society of Japan*, Vol.20, No.3, pp.281-289, ISSN: 0289-1824 (in Japanese).

Yamamura, M.; Miyazaki, K. & Kobayashi, S. (1995). A survey on learning for agents, *Journal of Japanese Society for Artificial Intelligence*, Vol.10, No.5, pp.683-689, ISSN: 1346-0714 (in Japanese).

Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning, *Machine Learning*, Vol.8, pp.279-292, ISSN: 0885-6125.

Weber, C.; Elshaw, M. & Mayer, N. M. (2008). *Reinforcement Learning, Theory and Application*, I-Tech Education and Publishing, ISBN: 978-3-902613-14-1, Vienna, Austria.



Application of Machine Learning
Edited by Yagang Zhang

ISBN 978-953-307-035-3
Hard cover, 280 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

The goal of this book is to present the latest applications of machine learning, which mainly include: speech recognition, traffic and fault classification, surface quality prediction in laser machining, network security and bioinformatics, enterprise credit risk evaluation, and so on. This book will be of interest to industrial engineers and scientists as well as academics who wish to pursue machine learning. The book is intended for both graduate and postgraduate students in fields such as computer science, cybernetics, system sciences, engineering, statistics, and social sciences, and as a reference for software professionals and practitioners. The wide scope of the book provides them with a good introduction to many application researches of machine learning, and it is also the source of useful bibliographical information.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821