Reinforcement Learning with Symbiotic Relationships for Multiagent Environments

Shingo Mabu, Masanao Obayashi and Takashi Kuremoto
Graduate School of Science and Engineering, Yamaguchi University, Tokiwadai 2-16-1
Ube, Yamaguchi , 755-8611, Japan
E-mail: mabu@yamaguchi-u.ac.jp, m.obayas@yamaguchi-u.ac.jp, wu@yamaguchi-u.ac.jp

Abstract

Multiagent systems, where many agents work together to achieve their objectives, and cooperative behaviors between agents need to be realized, have been widely studied In this paper, a new reinforcement learning framework considering the concept of “Symbiosis” in order to represent complicated relationships between agents and analyze the emerging behavior is proposed. In addition, distributed state-action value tables are designed to efficiently solve the multiagent problems with large number of state-action pairs. From the simulation results, it is clarified that the proposed method shows better performance comparing to the conventional reinforcement learning without considering symbiosis.

Keywords: reinforcement learning, symbiosis, multiagent system, cooperative behavior,

1. Introduction

There are many situations where interests of some parties are coincided or conflicted[1], for example, human relationships, cooperation or competition between companies, and even international relationships. Recently, the globalization is rapidly progressing, thus, the relationships between persons and organizations have become very complicated networks. On the other hand, information systems have been intelligent and working cooperatively with each other, for example, cloud networks, car navigation systems and automation by robots. Research on complex networks began around 1998[2] and it has attracted attentions recently as an important research for analyzing phenomena in social systems. Therefore, a model that can predict problems caused by the complex networks and propose the optimal solutions for the problems will be useful for realizing safe and secure social systems. In addition, if the best relationships between parties can be found, it will contribute to the development of the whole society.

In this paper, a novel reinforcement learning framework that introduces a concept of "Symbiosis" in order to build Win-Win relationships between parties even if each party is pursuing the maximization of its own profits. Symbiosis can be defined as a relationship where two or more organisms live in close association with each other[3], and several computational models based on the symbiosis in the ecosystem have been studied[4-7].

In the proposed reinforcement learning method, multiagent environments are considered where there are several agents (persons and organizations) that have cooperative, competitive or self-satisfied relationships,

Published by Atlantis Press

and such relationships are defined as "symbiotic vectors". The symbiotic vectors can represent six basic symbiotic relationships, i.e., mutualism, harm, predation, altruism, self-improvement and self-deterioration. The symbiotic vectors are used to calculate rewards given to each agent when updating Q values in reinforcement learning. The symbiotic vectors represent not only the target direction of self-benefit, but that of the other agents working in the same environment. As a result, the proposed method with symbiotic vectors can build action rules for controlling benefit of several agents. Therefore, the proposed method can predict the emergent phenomena under the current symbiotic relationships by implementing simulations.

This paper is organized as follows. In section 2, Q learning algorithm with distributed state-action value table is introduced for efficiently solving the multiagent problems with a large number of state-action pairs. Section 3 explains the proposed learning algorithm using symbiotic vectors. Section 4 describes the simulation environments and results. Finally, section 5 is devoted to conclusions.

2. Q learning with distributed state-action value tables

In the standard reinforcement learning, the number of state-action pairs increases exponentially as the numbers of inputs, objects to be perceived, and possible actions increase, that is called “The curse of dimensionality” [8]. In the case of multiagent systems, the problem becomes more serious and cannot be avoided. Here we design a “distributed state-action value table (Q table)” to overcome the explosion of the dimensionality problem for the reinforcement learning of multiagent systems..

2.1. *Representation of distributed Q tables*

Suppose that a set of inputs (sensors) of agents is I, and a set of possible actions is A. Then I is manually divided into several subsets, i.e., $I_1, I_2, \ldots, I_n$ ($I = \{I_1, I_2, \ldots I_n\}$), depending on the problems. For example, in the self-sufficient garbage collecting problem used in the simulations of this paper, there are mainly three tasks (objectives) which have to be achieved by agents, thus, I is divided into three subsets, i.e., I_1, I_2, I_3. As a result, three sub-Q-tables are created based on I_1, I_2, I_3, respectively (Fig. 1).

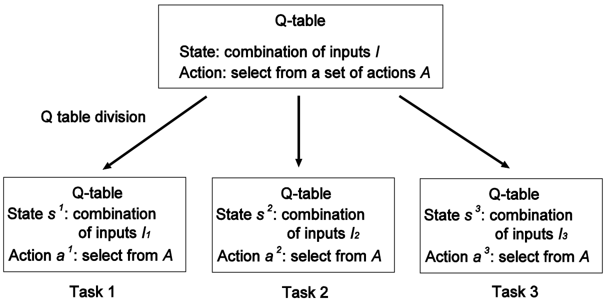
Q-table
State: combination of inputs I
Action: select from a set of actions A
Q table division
Q-table
State s^1: combination of inputs I_1
Action a^1: select from A
Q-table
State s^2: combination of inputs I_2
Action a^2: select from A
Q-table
State s^3: combination of inputs I_3
Action a^3: select from A
Task 1
Task 2
Task 3

Fig. 1. Q-table division

2.2. *State transition and learning*

In this subsection, the procedure of deciding an action is explained based on an example shown in Fig. 1 (three sub-Q-tables are used).

The procedure of deciding an action is as follows.

1) When inputs are given from an environment, each sub-Q-table independently determines the current state s^n (*n* is the sub-Q-table number. $n \in \{1,2,3\}$).

2) Three actions a^n are independently selected by each sub-Q-table using greedy policy[9].

3) Compare the three Q-values of a^n, and the sub-Q-table selecting the action with the maximum Q-value is defined as n^{winner} (winner-Q-table), and its current state is defined as s^{winner}.

4) The action selected by winner-Q-table is executed with the probability of ε, or a random action is executed with the probability of 1-ε. This executed action is defined as a^{winner}.

The update of Q value is executed by Eq. (1).

$$Q\left(n_t^{winner}, s_t^{winner}, a_t^{winner}\right) \leftarrow Q\left(n_t^{winner}, s_t^{winner}, a_t^{winner}\right) + \alpha\left[r + \gamma \max_{n,a^n} Q\left(n, s_{t+1}^n, a^n\right) - Q\left(n_t^{winner}, s_t^{winner}, a_t^{winner}\right)\right], \tag{1}$$

where, n_t^{winner}, s_t^{winner} and a_t^{winner} show the number, state and action of the winner sub-Q-table at time *t*, respectively. *n* is a sub-Q-table number, s^n_{t+1} is the state of sub-Q-table *n* at time *t+1*, and a^n is a possible action in sub-Q-table *n*. *r* is a reward, α ($0 < \alpha \leq 1.0$) is a learning rate, and γ ($0 \leq \gamma \leq 1.0$) is a discount rate

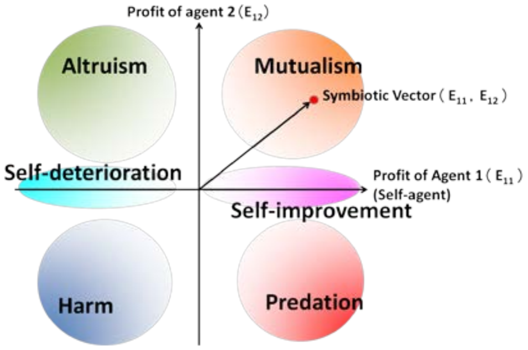
Profit of agent 2 (E$_{12}$)
Altruism
Mutualism
Symbiotic Vector (E$_{11}$, E$_{12}$)
Self-deterioration
Profit of Agent 1 (E$_{11}$)
(Self-agent)
Self-improvement
Harm
Predation

Fig. 2. Symbiotic vector and six symbiotic relationships for agent 1 (An example of two dimensions)

3. Reinforcement Learning with Symbiosis

This section introduces a symbiotic vector and how to apply the symbiotic vector to reinforcement learning.

3.1. *Symbiotic vectors*

Standard reinforcement learning aims to maximize rewards that the self-agent obtains, however, in the proposed method, not only the rewards for the self-agent, but also the rewards for other agents are consider to execute reinforcement learning. In addition, six symbiotic relationships are considered to build the action strategies, that is, Predation, Mutualism, Altruism, Harm, Self-improvement and Self-deterioration. Fig. 2 shows symbiotic strategy for "agent 1", where one axis shows the weight (E$_{11}$) on the benefit of agent 1 (self-agent), the other axis shows the weight (E$_{12}$) on the benefit of agent 2. In other words, E$_{11}$ shows the symbiotic strategy of agent 1 for agent 1, and E$_{12}$ shows the symbiotic strategy of agent 1 for agent 2. Therefore, if agent 1 aims to maximize rewards for both agents, it will take "Mutualism" strategy where symbiotic vector $\boldsymbol{v_1}$=(E$_{11}$, E$_{12}$)=(1.0, 1.0) ($-1.0 \le E_{11}, E_{12} \le 1.0$). Fig. 3 shows a symbiotic relationship between two agents, where one agent takes mutualism strategy and the other agent takes predation strategy. In this case, it can be considered that the symbiotic vector of agent 1 is $\boldsymbol{v_1}$=(E$_{11}$, E$_{12}$)=(1.0, 1.0), and that of agent 2 is $\boldsymbol{v_2}$=(E$_{21}$, E$_{22}$)= (-1.0, 1.0). Intermediate values between -1.0 and 1.0 can be also used to define a symbiotic relationship. For example, a symbiotic vector $\boldsymbol{v}$=(1.0, 0.1) shows a weak mutualism that consider the other agent's benefit a little. Therefore, the symbiotic vector flexibly represents any degree of symbiotic relationships, and moreover, it can be extended to the relationships between many agents.

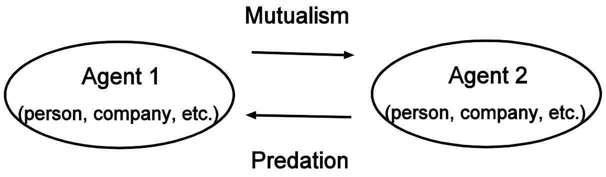

Fig. 3. Symbiotic relationships between two agents

3.2. *Reinforcement learning with Symbiotic vectors*

This subsection explains how to update Q values considering a symbiotic vector. Here, suppose there are *p* agents (agent #1 – #p), where the symbiotic vector of agent *k* ($1 \le k \le p$) is $\boldsymbol{v_k}$=(E_{k1}, E_{k2},..., E_{kp}). After agent *k* takes an action and finds rewards for all the agents (r^1, r^2, ..., r^P), the modified reward used for updating Q value of agent *k* in (1) is calculated as follows.

$$r_k = \sum_{l=1}^{p} E_{kl} r^l \qquad (2)$$

Eq. (2) calculates the sum of the weighted rewards of all the agents #1 – #p. For instance, when agent 1 takes mutualism strategy $\boldsymbol{v_1}$=(E$_{11}$, E$_{12}$)=(1.0, 1.0) and agent 2 takes predation strategy $\boldsymbol{v_2}$=(E$_{21}$, E$_{22}$)= (-1.0, 1.0), Eq. (2) for agent 1 and 2 can be represented by Eq. (3) and (4), respectively.

$$r_1 = 1.0 \times r^1 + 1.0 \times r^2 \qquad (3)$$

$$r_2 = -1.0 \times r^1 + 1.0 \times r^2 \qquad (4)$$

4. Simulations

4.1. *A simulation environment*

Self-sufficient garbage collecting problem[10] is used for the performance evaluation of the proposed method. Fig. 4 shows the simulation environment used in this paper, where there are two agents, 11 trashes, one charging station, and two trash collecting places. The aim of this problem is to collect many trashes in the environment and take them to the collecting places assigned to each

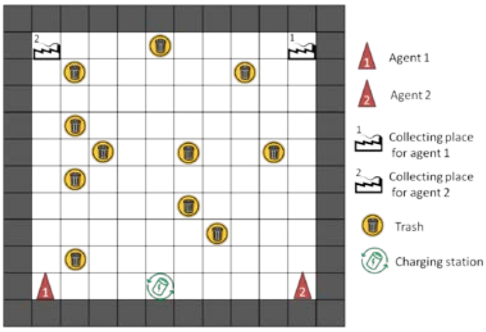
2
1
1
2
Agent 1
Agent 2
1 Collecting place for agent 1
2 Collecting place for agent 2
Trash
Charging station

Fig. 4. A simulation environment

Table 1. Inputs and actions

#	Input contents	Input value
1	forward cell	nothing, obstacle, collecting place for agent 1, collecting place for agent 2, trash, charging station
2	backward cell	the same as "forward cell"
3	right cell	the same as "forward cell"
4	left cell	the same as "forward cell"
5	direction of nearest trash	forward, backward, right, left
6	direction of charging station	forward, backward, right, left
7	direction of collecting place for agent 1	forward, backward, right, left
8	direction of collecting place for agent 2	forward, backward, right, left
9	the number of holding trashes	0, 1, 2 (max 2)
10	current energy level	low (less than 30), high (more than 70), middle (other values)
	Actions	
1	go forward	
2	turn right	
3	turn left	
4	no action	

Table 2. Sub-Q-table setting

table #	Main task	Input # used in each sub-Q-table
1	for agent 1's benefit	1,2,3,4,5,7,9
2	for agent 2's benefit	1,2,3,4,5,8,9
3	Charging energy	1,2,3,4,6,10

agent, i.e., agent k has to take trashes to the collecting place for agent k to obtain reward. In addition, each agent has a limited energy to move, thus the agents should check the remaining energy and go to the charging station before running out of the energy. Table 1 shows the inputs and possible actions that the agents can use. The initial energy is 100 (full charge), and when an agent goes forward, energy is used by three, and when it turns right or left, energy is used by one. The energy can be recharged gradually if the agent stays at the charging station. The total time for one episode is 100 steps. Reward 100 is given to agent k when an agent takes one trash to the colleting place for agent k, 10 is given when an agent collects a trash, and $0.1\times(\text{charged energy})$ is given when an agent stays at the charging station.

4.2. *Conditions of Q-learning with Distributed Q-tables*

Table 2 shows the setting of distributed Q-tables, where three sub-Q-tables are prepared for dealing with tasks for agent 1's benefit, for agent 2's benefit, and for charging energy, respectively. Each agent has its own Q-table (three sub-Q-tables), i.e., the reinforcement learning of the two agents are carried out independently. The learning parameters are set as learning rate α=0.1, discount rate γ=0.9, and ε=0.05.

4.3. *Simulation results*

Comparison between the proposed method with mutualism strategy and conventional Q learning

In this subsection, to confirm the basic effects of the symbiotic relationship, the proposed method with mutualism strategy, i.e., $\boldsymbol{v_1}=(E_{11}, E_{12})=(1.0, 1.0)$ $\boldsymbol{v_2}=(E_{21}, E_{22})=(1.0, 1.0)$, is compared with the conventional Q-learning, i.e., both agents take Self-improvement strategy, i.e., $\boldsymbol{v_1}=(E_{11}, E_{12})=(1.0, 0.0)$ $\boldsymbol{v_2}=(E_{21}, E_{22})=(0.0, 1.0)$.

Fig. 5 shows the number of trashes taken to the collecting places for agent 1 and that for agent 2, respectively, obtained by the conventional method. Each line is the average over 20 independent simulations. As the episode goes on, the number of trashes increases, and in 5000th episode, 1.5 trashes are taken to the collecting place for agent 1 and 2.25 trashes are for agent 2. Fig 6 shows the results of the proposed method, where 2.1 trashes are taken for agent 1 and 3.9 trashes are for agent 2. It should be noted that the number of collected trashes for agent 1 increases without

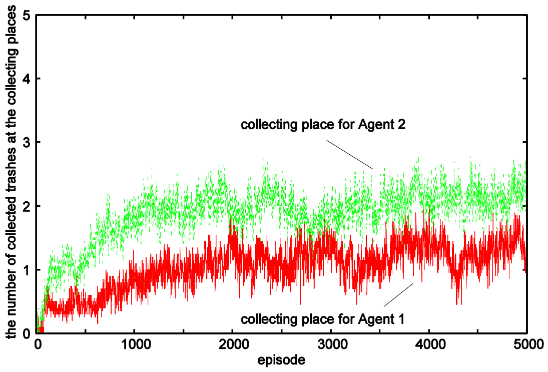

Fig. 5. The number of collected trashes at collecting places for Agent 1 and Agent 2 (Conventional method)

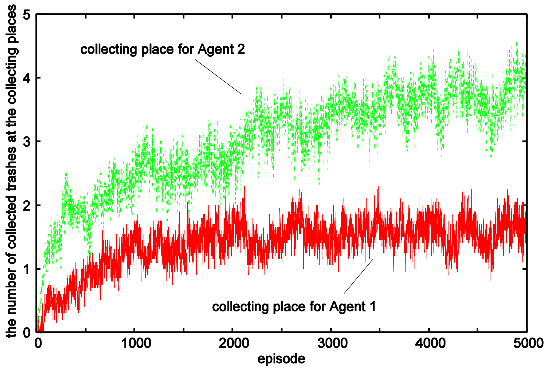

Fig. 6. The number of collected trashes at collecting places for Agent 1 and Agent 2 (Proposed method with Mutualism)

decreasing that for agent 2 (even increasing).Therefore, it is clarified that the proposed method with mutualism strategy can obtain cooperation behavior and show better performance than the conventional method. Next, the contribution of agent 1 and 2 to taking trashes to the collecting places is analyzed. Table 3 shows the data on the number of trashes taken to the collecting places in the final episode. As described before, the proposed method takes 2.10 trashes to the collecting place for agent 1 on average, where the contribution of agent 1, i.e., the number of trashes that agent 1 takes to the collecting place for agent 1, is 1.20, and the contribution of agent 2, i.e., the number of trashes that agent 2 takes to the collecting place for agent 1, is 0.90. Therefore, we can find that both agent 1 and 2 contribute to the benefit of agent 1. Q learning takes 1.50 trashes to the collecting place for agent 1 on average, where the contribution of agent 1 is 1.50 and that of agent 2 is zero, which means that only agent 1 contributes to the agent 1's benefit. Next, the number of trashes taken to the collecting place for agent 2 is analyzed. The proposed method takes 3.90 trashes on average, where the contribution of agent 1 is 1.95 and that of agent 2 is 0.90. Therefore, we can also find that both agents contribute to the benefit of agent 2. Q learning takes 2.25 trashes, where the contribution of agent 1 is 0.25 and that of agent 2 is 2.00, which shows that agent 1 contributes to the benefit of agent 2 only a little, and agent 2 contributes to its own benefit only.

Simulation results of other symbiotic relationships

The simulations with other symbiotic relationships are carried out to see the emergent phenomena. First, the result of the proposed method where both agents take predation strategy, i.e., $\boldsymbol{v_1}$=(E_{11}, E_{12})=(1.0, -1.0) $\boldsymbol{v_2}$=(E_{21}, E_{22})=(-1.0, 1.0), is shown in Fig. 7. It can be seen from the comparison between Figs. 6 and 7 that the predation strategy cannot obtain higher benefit than mutualism strategy in this simulation. Therefore, we can say that the highly competitive relationship would not make a progress of both agents. Second, the result of another set of strategies is shown in Fig. 8, where agent 1 takes self-improvement strategy: $\boldsymbol{v_1}$=(E_{11}, E_{12})=(1.0, 0.0), and agent 2 takes altruism strategy: $\boldsymbol{v_2}$=(E_{21}, E_{22})=(1.0, -0.8). In this case, agent 1 obtains higher benefit than agent 2, which is different result from the results of the other sets of symbiotic relationships which always show the higher benefit of agent 2 than agent 1. Altruism strategy of agent 2 contributes to the benefit of agent 1 even sacrificing its own benefit. As shown in the above results, the proposed method can execute simulations and obtain the emergent results under certain symbiotic relationships. In the future, we will

Table 3. Data on the number of trashes taken to the collecting places in the last episode

	The number of trashes taken for agent 1			The number of trashes taken for agent 2		
	Total	Contribution of agent 1	Contribution of agent 2	Total	Contribution of agent 1	Contribution of agent 2
Proposed method	2.10	1.20	0.90	3.90	1.95	1.95
Q learning	1.50	1.50	0.00	2.25	0.25	2.00

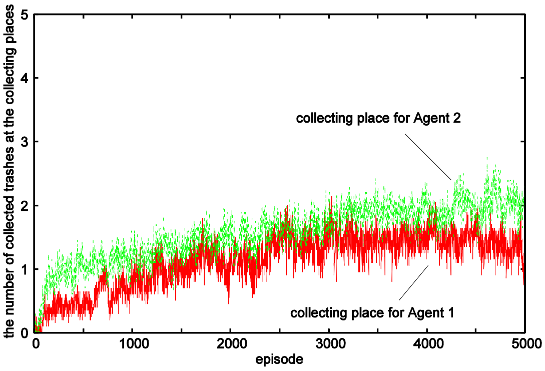

Fig. 7. The number of collected trashes at collecting places for Agent 1 and Agent 2 (Predation vs. Predation)

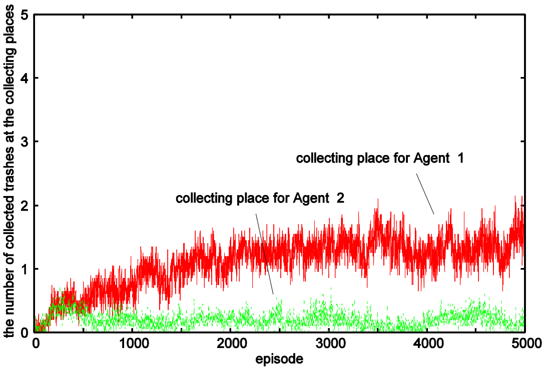

Fig. 8. The number of collected trashes at collecting places for Agent 1 and Agent 2 (Altruism vs Self-improvement)

consider to execute simulations with more complicated multilateral relationships which exist in the real society.

5. Conclusions

In this paper, a novel reinforcement learning framework based on the concept of symbiotic relationships was proposed, where symbiotic vector was introduced to represent various kinds of relationships. Additionally, distributed Q-tables were designed for the reinforcement learning in multiagent systems. In the simulations, the effectiveness of the proposed method with mutualism strategy, and the results of some other symbiotic relationships were shown. By considering not only the benefit of the self-agent, but also that of the other agents, cooperative behaviors emerged in the mutualism strategy. In the future, other combinations of symbiotic relationships will be considered to analyze the emerging behaviors, and moreover, multilateral relationships will be also considered to build simulation models dealing with real situations of cooperation or competition between agents.

References

1. Ken Binmore, *Game Teory: A Very Short Introduction* (Oxford Uniersity Press, 2007)
2. D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks, *Nature*, **393** (1998) 440-442.
3. L. Margulis, *The Symbiotic Planet*, Contact, (1999).
4. T. Eguchi, K. Hirasawa, J. Hu, and N. Ota, A Study of Evolutionary Multiagent Models Based on Symbiosis, *IEEE Trans. on Systems, Man, and Cybernetics, part B*, (**36**) 1, (2006).
5. J. Y. Goulermas and P. Liatsis, Hybrid symbiotic genetic optimization for robust edge-based stereo correspondence, *Pattern Recognition*, (**34**), (2001) 2477–2496.
6. J. Y. Goulermas and P. Liatsis, A collective-based adaptive symbiotic model for surface reconstruction in area-based stereo, *IEEE Trans. on Evolutionary Computation*, (**7**) 5, (2003), 482–502.
7. C. P. Pieters, Symbiotic networks, in *Proc. of the IEEE Congress on Evolutionary Computation*, (2003) 921–927.
8. R. E. Bellman, *Dynamic Programming*, (Princeton University Press, 1957).
9. R. S. Sutton, A. G. Batro, *Reinforcement Learning - An Introduction -*, (1998).
10. R. Pfeifer, and C. Scheier. *Understanding intelligence*, (MIT press, 1999).