https://doi.org/10.20965/jaciii.2009.p0667

Paper:

Time Horizon Generalization in Reinforcement Learning: Generalizing Multiple Q-Tables in Q-Learning Agents

Yasuyo Hatcho*, Kiyohiko Hattori*, and Keiki Takadama*,**

*The University of Electro-Communications,
1-5-1, Chofugaoka, Chofu, Tokyo 182-8585, Japan
**PRESTO, Japan Science and Technology Agency (JST)
4-1-8 Honcho Kawaguchi, Saitama 332-0012, Japan
Emails: hatcho@cas.hc.uec.ac.jp, {hattori, keiki }@hc.uec.ac.jp
[Received April 24, 2009; accepted June 19, 2009]

This paper focuses on *generalization* in reinforcement learning from the *time horizon* viewpoint, exploring the method that generalizes multiple Q-tables in the multiagent reinforcement learning domain. For this purpose, we propose *time horizon generalization* for reinforcement learning, which consists of (1) Q-table selection method and (2) Q-table merge timing method, enabling agents to (1) select which Q-tables can be generalized from among many Q-tables and (2) determine when the selected Q-tables should be generalized. Intensive simulation on the bargaining game as sequential interaction game have revealed the following implications: (1) both Q-table selection and merging timing methods help replicate the subject experimental results without ad-hoc parameter setting; and (2) such replication succeeds by agents using the proposed methods with smaller numbers of Q-tables.

Keywords: generalization, time horizon, sequential interaction, reinforcement learning

1. Introduction

In the Human-Agent Interaction (HAI) domain, *reinforcement learning* [1] aids agents in adapting to users by understanding user intent. One example is AIBO, a robot dog that learns user preferences. What should be noted here is that learning of an artifact (i.e., an agent or robot) is mostly done in *one* interaction. AIBO receives a reward immediately by raising a front paw when a user stretches a hand toward AIBO saying "Ote," ("shake" hands), simply represented by a one state-action pair in the reinforcement learning framework. *Sequential* interactions are required, however, to enrich the communication between the artifact and user, e.g., by showing more complex behavior (i.e., sequential behavior), of AIBO through extensive interaction.

Reinforcement learning agents should therefore, have sufficient numbers of Q-tables to cover *all* situations in total iterations, which requires a humongous learning time to correctly estimate expected rewards for all state-action pairs. In contrast, users memorize generalized state-action pairs rather than all state-action pairs. In Shogi, Japanese chess, for example, players memorize generalized state-action pairs of early, middle, and last game stages rather than all state-action pairs of the first move, second move, ... etc. Such knowledge generalization eliminates the need to memorize all situations. Here, we call knowledge generalization as time horizon generalization that generalizes multiple Q-tables from the time horizon.

Focusing on *time horizon generalization* in the reinforcement learning context unlike the *state generalization* [2] or *function approximation* [3] addressed in conventional reinforcement learning, we propose two methods for the time horizon generalization: (1) *Q-table selection method* enabling agents to select Q-tables to be generalized among many Q-tables; and (2) *Q-table merge timing method* enabling agents to determine when the selected Q-tables should be generalized. These two approaches generalize multiple Q-tables, in contrast to conventional Q-value generalization within a single Q-table, strictly speaking, the state-action space.

To explore the effectiveness of our proposed generalization methods, this paper applies them to the sequential bargaining game [4] resembling Shogi, and investigates whether reinforcement learning agents could use them to replicate the subject experiment results. We compare simulation and the subject experiment result, rather than evaluating performance as in conventional research, because (1) player behavior (i.e., strategy) changes over time as in Shogi, which is essential for evaluating agent ability regarding behavior change (i.e., whether agents acquire such behavior change over time and show complex behavior through interactions with users); and (2) since previous research [5] found that Q-learning agents [6] with sufficient numbers of Q-tables can replicate the subject experiment results in the same example, this paper investigates whether the same replication can be done even in time horizon generalization by roughly categorizing behavior change.

This paper is organized as follows. Section 2 explains generalization in reinforcement learning, and Section 3 describes our proposed generalization methods for the reinforcement learning. Section 4 details the bargaining

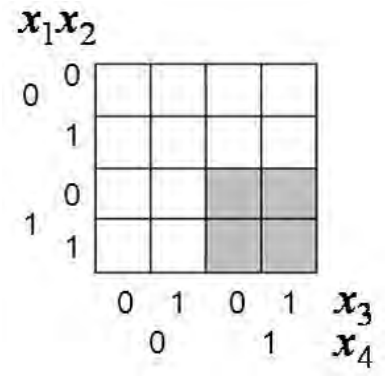


Fig. 1. State generalization.

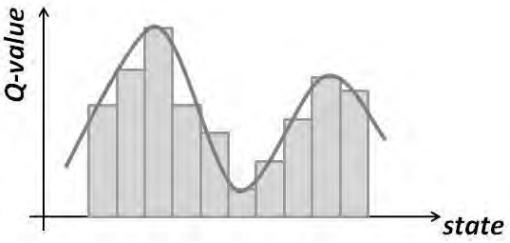


Fig. 2. Function approximation.

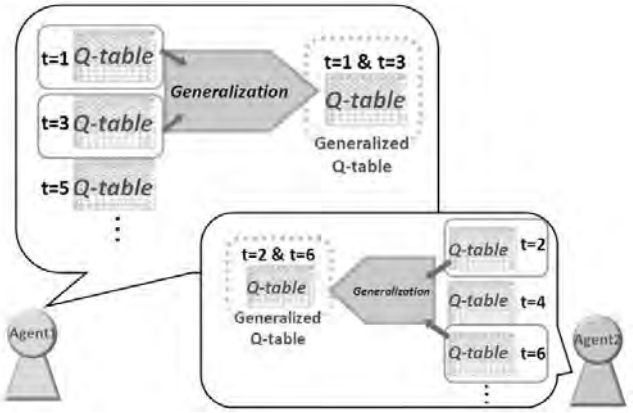


Fig. 3. Generalization in this paper.

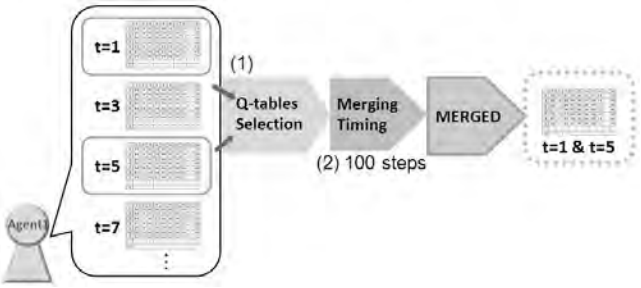


Fig. 4. A sequence of knowledge generalization.

game, Section 5 presents the computer simulation, and Section 6 discusses time horizon generalization in reinforcement learning. Finally, our conclusions are given in Section 7.

2. Generalization in Reinforcement Learning

This section gives an overview of *generalization* in reinforcement learning and our proposed *generalization*.

- **State Generalization**
  In the Learning Classifier System (LCS) [7, 8], consisting of numerous *if-then* rules (i.e., state-action pairs called *classifiers*) state generalization has been addressed by employing the concept of "*don't care*" for *if* (i.e., state) which is usually represented by "0", "1", and "♯". LCS tries to create rules that generalize the state by using "♯", *don't care* "0" or "1". **Fig. 1** shows an example of a simple problem classified by 4 bits (i.e., $x_1x_2x_3x_4$) where light gray squares are by the general state of 1♯♯1, i.e., x_1=1 and x_4=1, instead of specific states 1001, 1011, 1101 and 1111. In a related issue, state generalization was proposed in reinforcement learning with Support Vector Machines [9].

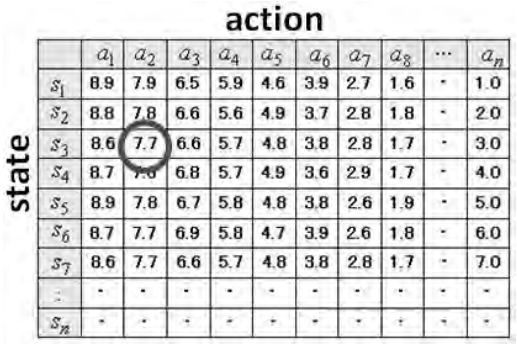
- **Function Approximation**
  One major *generalization* in reinforcement learning addresses an approximation of the value function such as shown in **Fig. 2** in a continuous problem. Justin and Andrew proposed the *Grow-Support algorithm* appropriately appropriating the value function in continuous environments [3].

- **Time Horizon Generalization**
  In contrast to the above two concepts of generalization, this paper focuses on the generalization of multiple Q-tables from the *time horizon*. Given an example in which agents have multiple Q-tables as shown in **Fig. 3** for multiple interactions, appropriate Q-tables are merged in the time horizon (e.g., Q-tables of $t = 1$ and 3 are merged in agent 1 while Q-tables of $t = 2$ and 6 are merged) to generalize them. As shown in **Fig. 3**, agent 1 has Q-tables of $t = 1, 3, 5, \ldots$ (i.e., not continuous such as $t = 1, 2, 3, \ldots$), because at the beginning of a Shogi game, for example, agent 1 determines the first move from Q-table of $t = 1$, after which, agent 2 determines the second move from Q-table of $t = 2$, and agent 1 determines the third move from Q-table of $t = 3$.

3. Generalizing Q-Tables in Agents

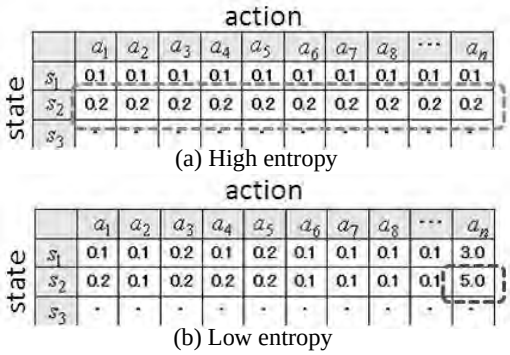
To enable agents to generalize Q-tables just as users can generalize knowledge, this paper proposes methods merging the Q-tables that agents have. We call this process as Q-table generalization in which two slightly different Q-tables are merged as one generalized Q-table covering both. **Fig. 4**, for example, shows that Q-tables of $t = 1$ and 5 are merged as one Q-table, where Q-values in the same state and action between Q-tables are averaged.

The merged Q-table determines the action of the agent before merging and is also updated during learning. The critical points of merging are to (1) select appropriate Q-



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	⋯	a_n
s_1	8.9	7.9	6.5	5.9	4.6	3.9	2.7	1.6	-	1.0
s_2	8.8	7.8	6.6	5.6	4.9	3.7	2.8	1.8	-	2.0
s_3	8.6	7.7	6.6	5.7	4.8	3.8	2.8	1.7	-	3.0
s_4	8.7	7.6	6.8	5.7	4.9	3.6	2.9	1.7	-	4.0
s_5	8.9	7.8	6.7	5.8	4.8	3.8	2.6	1.9	-	5.0
s_6	8.7	7.7	6.9	5.8	4.7	3.9	2.6	1.8	-	6.0
s_7	8.6	7.7	6.6	5.7	4.8	3.8	2.8	1.7	-	7.0
⋮	-	-	-	-	-	-	-	-	-	-
s_n	-	-	-	-	-	-	-	-	-	-

Fig. 5. Q-table.



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	⋯	a_n
s_1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
s_2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
s_3	-	-	-	-	-	-	-	-	-	-

(a) High entropy

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	⋯	a_n
s_1	0.1	0.1	0.2	0.1	0.2	0.1	0.1	0.1	0.1	3.0
s_2	0.2	0.1	0.2	0.2	0.2	0.1	0.1	0.1	0.1	5.0
s_3	-	-	-	-	-	-	-	-	-	-

(b) Low entropy

Fig. 6. Q-values entropy.

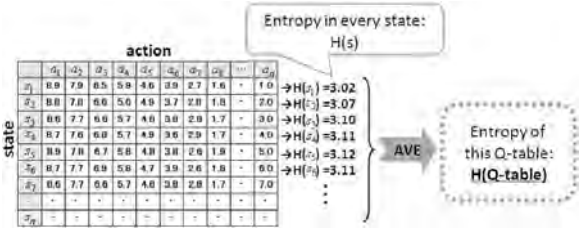


Fig. 7. Q-table entropy.

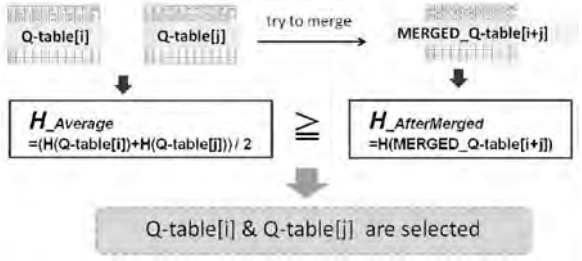


Fig. 8. Q-table selection.

tables to be generalized from among many Q-tables (e.g., Q-tables of $t = 1$ and 5), and (2) determine when the selected Q-tables should be generalized (e.g., 100 steps, i.e., one step indicates one game) as discussed in Section 3.2.

3.1. Basic Agent Model

We consider a basic reinforcement learning agent model such as the Q-learning agent [6] having multiple Q-tables consisting of many Q-values as shown **Figs. 3** and **4**. The Q-table consists of a matrix of the number of situations by the number of actions as shown in **Fig. 5**, where the vertical and horizontal axes indicate the states and actions, respectively. Q-value, $Q(s,a)$, in each Q-table indicates values (i.e., estimation of the expected reward) that the agent will acquire when conducting action "a" in state "s". $Q(s_3,a_2)$ in the circle shown in **Fig. 5**, for example, means that the agent will have reward 7.7 if taking action a_2 when in state s_3. When the agent uses Q-learning, the Q-value is updated as follows, where, α, r, S', and a' are the learning rate, reward, the next state, and the next action, respectively:

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

Q(s',a') is the Q-value of the next step (i.e., not in the same Q-table). In **Fig. 3**, for example, the Q-value in the Q-table of $t = 1$ is updated with the Q-value in the Q-table of $t = 3$.

3.2. Time Horizon Generalization

3.2.1. Q-table Selection Method

Q-tables must be selected appropriately because merging inappropriate Q-tables causes incorrect Q-table generalization, meaning that not similar Q-tables are merged. To select appropriate Q-tables for merging, our proposal uses *entropy* to calculate information uncertainty (i.e., Q-table uncertainty).

Note that Q-table uncertainty is high when most Q-values are similar, making it difficult to select a single best action, and vice versa. **Fig. 6** shows examples of Q-values with (a) high and (b) low entropy. $Q(s_2,a_1)$, $Q(s_2,a_2)$, … $Q(s_2,a_n)$ have similar values of almost 0.2, which derives high entropy in state s_2, as shown in **Fig. 6(a)**. $Q(s_2,a_n)$ ($= 5.0$) is the highest among the other Q-values in state s_2 which derives low entropy as shown in **Fig. 6(b)**.

Basically, our proposed method selects Q-tables when merged Q-table uncertainty is low or almost the same compared to non-merged Q-tables. Each pair of Q-tables is therefore processed as follow:

1. A probability of every state and action pair is calculated by Eq. (2), where n is the number of actions. Instead of roulette selection, Boltzmann Distribution is used.

$$p(a \mid s) = \frac{Q(s,a)}{\sum_{k=1}^{n} Q(s,a_k)} \quad \ldots \quad (2)$$

2. Entropy of each state (s), $H(s)$, is calculated by Eq. (3) as shown in **Fig. 7** (i.e., *H(s=1)*, *H(s=2)*, etc).

$$H(s) = -\Sigma_{k=1}^{n} p(a_k \mid s) \log p(a_k \mid s) \quad \ldots \quad (3)$$

3. Entropy of Q-table[i], *H(*Q-table[i]*)*, is calculated by averaging all entropy of states ($H(s)$) as shown in **Fig. 7**, i.e., *H(*Q-table[i]*)*=$(\sum_{i=1}^{n} H(s=i))/n$. Entropy of Q-table[j], *H(*Q-table[j]*)*, is calculated the same as *H(*Q-table[i]*)*.

4. $H_{Average}$ is calculated by averaging *H(*Q-table[i]*)* and *H(*Q-table[j]*)* (*=(H(*Q-table[i]*)+H(*Q-table[j]*))/2*) as shown at left in **Fig. 8**.

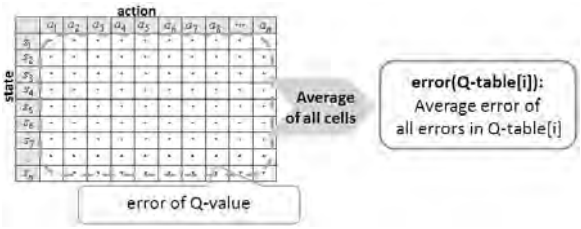


Fig. 9. Error of Q-table[i].

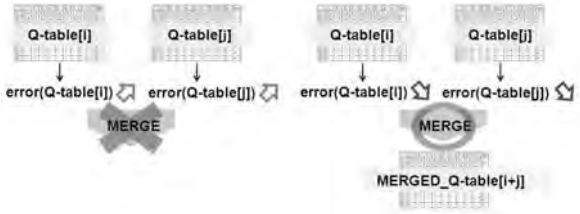


Fig. 10. Q-table merge timing.

5. Entropy of the temporally merged Q-table[i+j], merged from Q-table[i] and Q-table[j] by averaging Q-values in the same state and action between Q-tables, $H_{AfterMerged}$ is calculated the same as *H(*Q-table[i]*)* or *H(*Q-table[j]*)* (i.e., calculated by averaging all entropy of states in merged Q-table[i+j]) as shown at the right in **Fig. 8**.

6. Q-table[i] and Q-table[j] are selected for merging when $H_{AfterMerged}$ is smaller than or almost equal to $H_{Average}$ as shown in **Fig. 8**.

3.2.2. Q-table Merge Timing

Determining the timing of Q-table generalization is also important because it is non-sense to merge not fully learned Q-tables whose Q-values are not enough to be updated. This indicates that early Q-table generalization probably causes incorrect Q-table merging. To determine when the selected Q-tables should be generalized, our proposed method employs the error of Q-value ($error(s,a)$), indicating the gap between the next step $Q(s',a')$ and the current $Q(s,a)$ and updated as follows Eq. (4):

$$error(s,a) \leftarrow error(s,a) + \alpha(|\, r + \gamma max_{a' \in A'} Q(s',a') - Q(s,a) \,| - error(s,a)). \quad (4)$$

Basically, the selected Q-tables are merged when error decreases because (a) $error(s,a)$ increases in early learning when the initial value of $error(s,a)$ typically set near 0 is far from acquired rewards and (b) $error(s,a)$ decreases in later learning phase when Q-value approaches expected rewards.

The selected Q-table[i] and Q-table[j] are therefore processed as follows:

1. *Error(*Q-table[i]*)* (i.e., error of the i-th Q-table) is calculated by averaging *error(s,a)* of all state and action pairs as shown in **Fig. 9**. *Error(*Q-table[j]*)* is calculated the same way as *Error(*Q-table[i]*)*.

Fig. 11. An example of the sequential bargaining game.

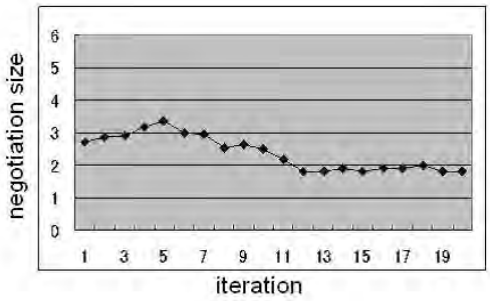


Fig. 12. Negotiation size of the subject experiments [5].

2. Q-table[i] and Q-table[j] are merged when *error(*Q-table[i]*)* and *error(*Q-table[j]*)* decrease as shown at right in **Fig. 10**, meaning they are not merged when both *error(*Q-table[i]*)* and *error(*Q-table[j]*)* increase as shown at the left in **Fig. 10**.

Note that the selected Q-tables are merged when *either* of *errors(*Q-table[t]*)* decreases because one Q-table is mostly converged at the correct expected reward, which helps derive the correct convergence of other Q-tables.

4. Problem Description

This paper employs the *bargaining game* as an example of sequential interaction. The bargaining game [10] is studied, in which two or more players try to reach to mutually beneficial agreements through negotiations, in the context of bargaining theory [4] in game theory [11]. This game has been proposed for investigating when and what types of offers of individual players could be accepted other players.

Take the example in **Fig. 11**, substituting "agent" for "player" from now on.

The two agents must decide the ratio of reward R (=10) through negotiations. Agent 1, for example, starts by offering "30%" to agent 2, who counter-offers "10%" by refusing the original offer from agent 1. Through such negotiations, agent 2 finally offers "40%" and agent 1 accepts, meaning agent 1 and 2 accrue 4 and 6 rewards, respectively. If the game exceeds the maximum number of negotiation size (MAXSTEP), which is the common knowledge of agents, neither agent accrues reward.

Note that the agent making the last offer (the last offerer) is advantageous because both agents know the maximum number of negotiation size in the game as common

knowledge. The other agent must accept the last offer to acquire some reward, even if the last offer is minimal offer, (i.e., if the agent does not accept it, no reward at all accrues). If the first offerer is agent 1 and the maximum number of negotiation size is 6, for example, agent 2 is better positioned to receive the larger reward by making an advantageous offer in the last negotiation step. Due to such a situation the equilibrium payoff of the theoretical approach is 1:9, however, the payoff of the subject experiments is mostly 5:5 [5]. The difference between equilibrium and experimental payoffs cause negotiation instability.

Figure 12 shows the subject experiment result obtained in previous research [5], where vertical and horizontal axes indicate negotiation size and iterations (i.e., the number of games, specifically, one game ends when a player accepts the offer of the opponent player or the negotiation size exceeds at the maximum number of negotiation size). In this experiment, the maximum number of negotiations set to 6 and the first offerer is fixed.

This figure suggests that the negotiation size increases in early iterations and decreases later because of the following reasons: (1) the negotiation size increases in the first several iterations because neither human player guesses strategy of the other player, which prompts them to explore a larger payoff by mutually competing, requiring further negotiations (i.e., more negotiations required to explore a larger payoff); and (2) the negotiation size decreases in later iterations because both human players find a mutually agreeable payoff through knowing strategies of the other, which decreases their motivation to negotiate again (i.e., a few negotiation size is enough to determine their payoffs) [12].

We call this tendency in changing human thinking *increasing and decreasing trend*. Such a trend as behavior changes is essential for agents to facilitate complex iterations with a user as mentioned in Section 1, so this paper investigates whether agents replicate this trend found in the subject experiment, precisely, even by the small number of Q-tables generalized by our proposal.

5. Simulation

5.1. Simulation Cases

To explore the effectiveness of the time horizon generalization methods in Section 4, this paper investigates whether agents employing the proposed methods can replicate the subject experiment by comparing results of the agents employing our proposals to those of agents employing the following heuristic method:

$$\sum_{s\in S,\ a\in A} COUNTIF(\mid Q_i(s,a) - Q_j(s,a) \mid \leq D) \geq \mid S\times A \mid /2. \quad (5)$$

S and A indicate a set of all states and actions of one Q-table. This heuristic method (i) calculates the difference in all state and action pairs of *Q(s,a)* between Q-table[i] and Q-table[j] (i.e.,$|Q_i(s,a) - Q_j(s,a)|$); (ii) calculates the number of Q-values whose difference is less than or equal to D, i.e., the threshold; and (iii) selects Q-table[i] and Q-table[j] when such a number exceeds or equals $\mid S\times A \mid /2$, i.e., the half size of all state and action pairs. $COUNTIF(x)$ counts the number when x is true. When D is set to 3, for example, x (5) is true for $Q_i(2,3)$=3 and $Q_j(2,3)$=4.5 because $|Q_i(2,3) - Q_j(2,3)|$=1.5<3. When D is set to 1, however, it is not true for the same case because $|Q_i(2,3) - Q_j(2,3)|$=1.5>1.

action

state	10%	20%	30%	40%	50%	60%	70%	80%	90%	Acc
1st	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
10%	8.8	7.8	6.6	5.6	4.9	3.7	2.8	1.8	0.8	1.0
20%	8.6	7.7	6.6	5.7	4.8	3.8	2.8	1.7	0.9	2.0
30%	8.7	7.6	6.8	5.7	4.9	3.6	2.9	1.7	0.8	3.0
40%	8.9	7.8	6.7	5.8	4.8	3.8	2.6	1.9	0.9	4.0
50%	8.7	7.7	6.9	5.8	4.7	3.9	2.6	1.8	0.8	5.0
60%	8.6	7.7	6.6	5.7	4.8	3.8	2.8	1.7	0.8	6.0
70%	·	·	·	·	·	·	·	·	·	7.0
80%	·	·	·	·	·	·	·	·	·	8.0
90%	·	·	·	·	·	·	·	·	·	9.0

Fig. 13. Bargaining game Q-table.

The following cases were conducted as comparative simulation referring the subject experiment results [5] as shown in **Fig. 12**:

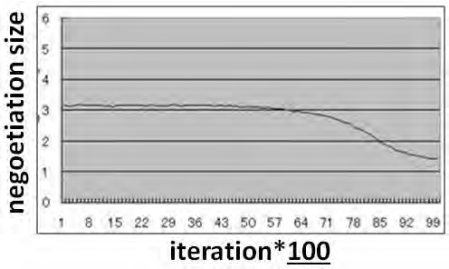
- **(a)** Q-tables generalized in 2000 iterations heuristically using D=3 (almost similar Q-tables), in Eq. (5).
- **(b)** Q-tables generalized in 2000 iterations heuristically using D=1 (very similar Q-tables), in Eq. (5).
- **(c)** Q-tables generalized in 500 iterations heuristically using D=1 (very similar Q-tables) in Eq. (5).
- **(d)** Q-tables generalized by both the proposed Q-table selection and merge timing methods.

Each agent can offer 10%, 20%, ..., 90% to the other agent alternately three times (i.e., the maximum number of negotiation size is 6). Total reward is R=10. Q-table of consists, as shown in **Fig. 13**, of each agent having states indicating offers of "10%", "20%", ..., and "90%" from the other agent (vertical axis), and actions indicating offers of "10%", "20%", ..., and "90%" to the other agent (horizontal axis). Acceptance of the other agent's offer is represented by "*Acc*". The state of the beginning of the game is represented by "*1st*" in which the agent who makes the first offer is at the beginning of every game.

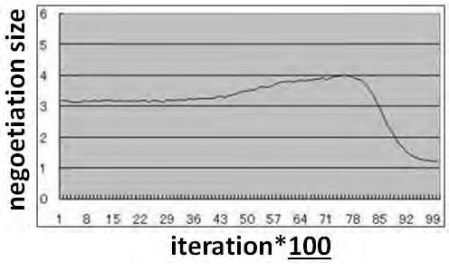
The result of the negotiation size in each case shown in the next subsection is averaged over 100 runs.

5.2. Simulation Results

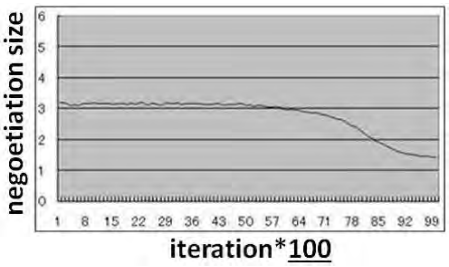
Figure 14 shows simulation results for **(a)-(d)**, where vertical and horizontal axes indicate the negotiation size and iterations, respectively. Negotiation size decreases without increasing in **Fig. 14(a)**, while negotiation size



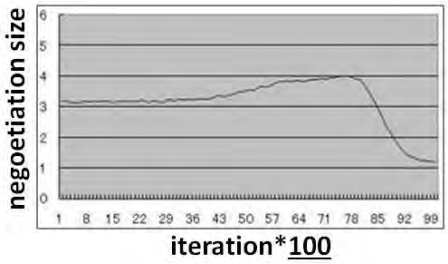
(a) Heuristic method using D=3 merging Q-tables in 2000 iterations



(b) Heuristic method using D=1 merging Q-tables in 2000 iterations

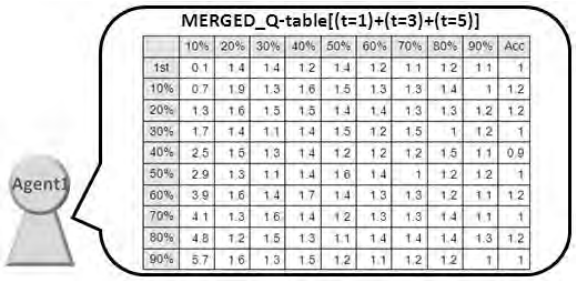


(c) Heuristic method using D=1 merging Q-tables in 500 iterations



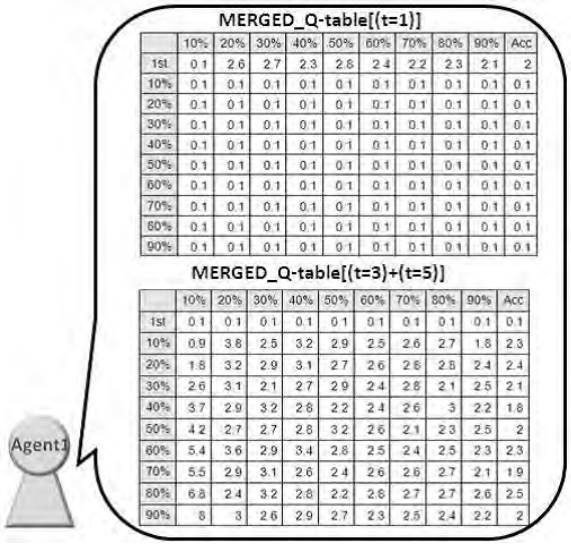
(d) Proposed method selecting and merging Q-tables

Fig. 14. Simulation Results.



	10%	20%	30%	40%	50%	60%	70%	80%	90%	Acc
1st	0.1	1.4	1.4	1.2	1.4	1.2	1.1	1.2	1.1	1
10%	0.7	1.9	1.3	1.6	1.5	1.3	1.3	1.4	1	1.2
20%	1.3	1.6	1.5	1.5	1.4	1.4	1.3	1.3	1.2	1.2
30%	1.7	1.4	1.1	1.4	1.5	1.2	1.5	1	1.2	1
40%	2.5	1.5	1.3	1.4	1.2	1.2	1.2	1.5	1.1	0.9
50%	2.9	1.3	1.1	1.4	1.6	1.4	1	1.2	1.2	1
60%	3.9	1.6	1.4	1.7	1.4	1.3	1.3	1.2	1.1	1.2
70%	4.1	1.3	1.6	1.4	1.2	1.3	1.3	1.4	1.1	1
80%	4.8	1.2	1.5	1.3	1.1	1.4	1.4	1.4	1.3	1.2
90%	5.7	1.6	1.3	1.5	1.2	1.1	1.2	1.2	1	1

(a) Heuristic method using D=3 with merging Q-tables in 2000 iterations



	10%	20%	30%	40%	50%	60%	70%	80%	90%	Acc
1st	0.1	2.6	2.7	2.3	2.6	2.4	2.2	2.3	2.1	2
10%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
20%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
30%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
40%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
50%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
60%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
70%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
80%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
90%	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

	10%	20%	30%	40%	50%	60%	70%	80%	90%	Acc
1st	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
10%	0.9	3.8	2.5	3.2	2.9	2.5	2.6	2.7	1.8	2.3
20%	1.8	3.2	2.9	3.1	2.7	2.6	2.8	2.8	2.4	2.4
30%	2.6	3.1	2.1	2.7	2.9	2.4	2.8	2.1	2.5	2.1
40%	3.7	2.9	3.2	2.8	2.2	2.4	2.6	3	2.2	1.8
50%	4.2	2.7	2.7	2.8	3.2	2.6	2.1	2.3	2.5	2
60%	5.4	3.6	2.9	3.4	2.8	2.5	2.4	2.5	2.3	2.3
70%	5.5	2.9	3.1	2.6	2.4	2.6	2.6	2.7	2.1	1.9
80%	6.8	2.4	3.2	2.8	2.2	2.8	2.7	2.7	2.6	2.5
90%	8	3	2.6	2.9	2.7	2.3	2.5	2.4	2.2	2

(d) Proposed methods selecting and merging Q-tables

Fig. 15. Merged Q-tables details.

increases in early iterations and decreases in later iterations in **Fig. 14(b)**. The result of **(b)** shows the same trend in the subject experiment results as shown in **Fig. 12**, indicating that appropriate Q-table generalization was done when the very similar Q-table (D=1) merged. Even for the very similar Q-tables ($D = 1$), however, appropriate Q-table generalization was not done when Q-tables merged in 500 iterations as shown in **Fig. 14(c)**. Compared to cases **(a), (b)**, and **(c)**, **Fig. 14(d)** shows the same trend in the subject experiment results even when Q-tables are merged in 1900 iterations, automatically determined by the proposed method.

6. Discussion

Figure 14(d), showed that agents who select Q-tables by proposed Q-table selection method and merge them at the proposed Q-table merge timing method can replicate the subject experiment results as shown in **Fig. 12** because (1) Q-tables selected by the proposed method are similar, which can remain the feature of Q-tables (i.e., the strategy of negotiation) after they are merged, and (2) merge timing calculated by the proposed method is appropriate (i.e., Q-values approach the correct expected reward) preventing merging at incorrect timing.

Q-table[$t = 1$] is not merged with other Q-tables, even though Q-tables [$t = 3$] and [$t = 5$] are, as shown below in **Fig. 15**, indicating that only similar Q-tables are merged. Such separate Q-table merging is achieved only by providing enough time to update Q-values that show different values, indicating that appropriate Q-table merge timing is critical in addition to appropriate Q-table selection.

Figure 14(b) showed that agents who select Q-tables heuristically ($D = 1$) and merge them in 2000 iterations also replicate the subject experiment results for the same reason as in (d). $D = 1$, for which the difference between Q-tables is less than 1, helps selecting the similar Q-table, and 2000 iterations are enough to update different Q-values. What should be noted here, however, is that it is difficult to find appropriate values for D (1 for this problem) and merge timing (2000 iterations for this problem). Since such values are directly problem-dependent, agents employing our proposed methods replicate the subject experiment results without ad-hoc parameter setting.

Compared to these agents, both agents who select Q-tables heuristically using $D = 3$ and merge them at 2000 iterations as shown in **Fig. 14(a)** and agents who select Q-tables heuristically using $D = 1$ and merge them at 500 iterations as shown in **Fig. 14(c)**, on the other hand, cannot replicate the subject experiment results. This is because all Q-tables are merged into one Q-table that destroys Q-tables features (i.e., the strategy of negotiation) after they are merged as shown in **Fig. 15**. This is caused by the following reasons: (1) incorrect Q-tables are selected heuristically using $D = 3$ in (a) because $D = 3$ allows merging of not very similar Q-tables; and (2) merge timing at 500 iterations in (c) is inappropriate, i.e., Q-values have not yet approached the correct expected reward, raising the already high possibility of enabling incorrect Q-table merging.

Q-table selection and merge timing are therefore critical for replicating the subject experiment result without ad-hoc parameter setting. This is very significant because the same result cannot be obtained if inappropriate Q-tables are merged or if merge timing is faster than Q-values convergence.

In this simulation, each agent has three Q-tables. Our proposals do not depend on the number of Q-tables, however, enabling them to perform well in larger-scale problems requiring more Q-tables that agents have.

7. Conclusions

This paper focused on generalization in reinforcement learning from the time horizon, and has explored the methods that generalize multiple Q-tables in multiagent reinforcement learning. For this purpose, we proposed the time horizon generalization for reinforcement learning, consisting of (1) Q-table selection method and (2) the Q-table merge timing method, enabling agents to (1) select which Q-tables to be generalized among many Q-tables and (2) determine when the selected Q-tables should be generalized. Intensive simulations in the bargaining game as sequential interaction game have revealed the following implications: (1) both Q-tables selection and merging timing methods help replicate the subject experiment results without ad-hoc parameter setting; and (2) such replication succeeds using the proposed agents with smaller numbers of Q-tables.

The following issues should be pursued in the near future remained to be resolved: (1) investigating how the agent with our time horizon generalization method interacts with users; (2) applying our time horizon generalization method to other engineering problem; and (3) generalizing knowledge in Q-value units instead of with a time horizon.

References:

[1] R.S. Sutton and A.G. Bart, "Reinforcement Learning -An Introduction-," The MIT Press, 1998.

[2] S.W. Wilson, "Classifier Fitness Based on Accuracy," Evolutionary Computation, Vol.3, No.2, pp. 149-175, 1995.

[3] A. B. Justin and W. M. Andrew, "Generalization in Reinforcement Learning: Safely Approximating the Value Function," In Proc. of Neural Information Processings Systems 7, 1995.

[4] A. Muthoo, "Bargaining Theory with Applications," Cambridge University Press, 1999.

[5] T. Kawai, Y. Koyama, and K. Takadama, "Modeling Sequential Bargaining Game Agents Towards Human-like Behaviors: Comparing Experimental and Simulation Results," The First World Congress of the Int. Federation for Systems Research (IFSR'05), pp. 164-166, 2005.

[6] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," Machine Learning, Vol.8, pp. 55-68, 1992.

[7] J. H. Holland, and J. Reitman, "Cognitive Systems Based on Adaptive Algorithms," in D. A. Waterman, and F. Hayes-Roth, (Eds.), Pattern Directed Inference Systems, Academic Press, pp. 313-329, 1978.

[8] J. H. Holland, "The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-based System," Escaping Brittleness, Machine Learning, Vol.2, pp. 593-623, 1986.

[9] R. Goto and K. Matsuo, "State Generalization Method with Support Vector Machines in Reinforcement Learning," Trans. of the Institute of Electronics, Information and Communication Engineers. D-I, pp. 897-905, 2003 (in Japanese).

[10] A. Rubinstein, "Perfect Equilibrium in a Bargaining Model," Econometrica, Vol.50, No.1, pp. 97-109, 1982.

[11] M. J. Osborne and A. Rubinstein, "A Course in Game Theory," MIT Press, 1994.

[12] K. Takadama, T. Kawai, and T. Koyama, "Micro- and Macro-Level Validation in Agent-Based Simulation: Reproduction of Human-Like Behaviors and Thinking in a Sequential Bargaining Game," J. of Artificial Societies and Social Simulation (JASSS), Vol.11, No.2, 2008.

Name:
Yasuyo Hatcho

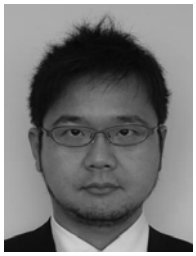
Affiliation:
The University of Electro-Communications

Address:
1-5-1 Chofugaoka, Chofu, Tokyo

Brief Biographical History:
2008 Received M.E. Degree at the University of Electro-Communications
2008- Master Course Student at the University of Electro-Communications

Main Works:
• Her research interests reinforcement learning and social simulation.

Membership in Academic Societies:
• The Institute of Electronics, Information and Communication Engineering (IEICE)



Name:
Kiyohiko Hattori

Affiliation:
The University of Electro-Communications

Address:
1-5-1 Chofugaoka, Chofu, Tokyo

Brief Biographical History:
2006 Doctor of Engineering Degree at the Tokyo Institute of Technology
2006- Assistant Professor at the University of Electro-Communications

Main Works:
• His research interests include multiagent system, distributed system, wireless communication, artificial intelligence, and autonomy.

Name:
Keiki Takadama

Affiliation:
*The University of Electro-Communications
**PRESTO, Japan Science and Technology Agency (JST)

Address:
*1-5-1 Chofugaoka, Chofu, Tokyo, Japan
**4-1-8 Honcho Kawaguchi, Saitama, Japan

Brief Biographical History:
1998 Doctor of Engineering Degree at the University of Tokyo
1998-2002 Joined Advanced Telecommunications Research Institute (ATR) International as the a visiting researcher
2002-2006 Tokyo Institute of Technology as a lecturer
2006- Associate Professor at the University of Electro-Communications

Main Works:
• His research interests include multiagent system, distributed artificial intelligence, autonomous system, reinforcement learning, learning classifier system, and emergent computation.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE), member
• AI- and Informatics-related Academic Societies, member