Cooperative Behavior in an Iterated Game with a Change of the Payoff Value

Shigeo Matsubara and Makoto Yokoo
NTT Communication Science Laboratories
2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
{matubara, yokoo}@cslab.kecl.ntt.jp

Abstract

We have formalized a finite iterated game with change. The formalization extends a traditional framework, e.g., prisoner's dilemma, by incorporating an influence on the payoff matrix at some future point through the execution of an action at the present time. This enables us to explain why cooperative behavior emerges in human interactions, even though from a myopic view, cooperative behavior does not seem to be profitable. Furthermore, we propose a new method for selecting an action in such a framework. The method overcomes the drawbacks of previous methods. Our proposed method can yield cooperative behavior and is not time-consuming. We analyze the properties of our method by using a simple model. Finally, we compare previous methods and our method by evaluating some example problems in terms of efficiency, stability and simplicity.

Introduction

One difficulty of decision making in multiagent environments is that the payoff of an agent depends on actions executed by other agents, as well as the agent itself. One way of dealing with such a situation is to apply the game theory; it has been used in the Distributed AI field as well [Rosenschein and Zlotkin, 1994][Genesereth *et al.*, 1986]. In situations where we are going to apply agent technology to real world problems, selecting an action by an agent is not done only once. Furthermore, executing an action at the present time will change a subsequent situation. Therefore, we formalize situations as an iterated game with change, and discuss how to select appropriate actions.

Our motivation in this research is to explain why cooperative behavior emerges in human interactions, even though from a myopic view, cooperative behavior does not seem to be profitable, e.g., volunteering to answer novice questions in Internet news.

Such a situation, in particular, a situation called the prisoner's dilemma, has been extensively studied in the field of game theory and multiagent systems. Surprisingly, if we assume that agents have complete rationality, cooperative behavior does not emerge. Since this does not match our intuition for human behavior in the real world, there has been various approaches to show how cooperative behavior can emerge in such situations. For example, [Mor and Rosenschein, 1995] showed that if the agents have limited rationality, they can cooperate in the iterated prisoner's dilemma game.

In this paper, we show that by slightly modifying a problem setting, such that executing an action at the present time will change a subsequent situation, agents with complete or limited rationality can cooperate. This is provided that the agents can take into account the changes in the subsequent situations.

We compare three methods for selecting an action, under the condition that self-action and the action of another agent change the payoff values in subsequent situations. The first and second methods are in a traditional framework, and the third one is proposed by us. The first method separately solves each constituent game. It is simple, but cannot yield cooperative behavior. The second method executes backward inference. It is based on a subgame perfect equilibrium, which relates to the stability of an action sequence. The solution obtained by this method has the desirable properties in that it belongs to a subgame perfect equilibrium and involves cooperative actions. However, this method is time-consuming. If there is a deadline in decision making, this becomes a serious problem. In order to overcome these drawbacks, we propose a new method that incorporates an influence of changes in payoff values into the evaluation of each action. It is not time-consuming and is able to yield cooperative behavior. We examine the properties of this method by analyzing a simple model. Furthermore, we clarify the properties of these three methods by utilizing some example problems from the points of efficiency, stability and simplicity.

In Section 2, we formalize a finite iterated game with change. In Section 3, we show three methods for solving the problem. In Section 4, we compare the three methods by using some example problems.

Iterated Game with Change

In this section, we make clear what problem we will tackle. Formalization of the problem is based on the game theory. We point out the necessity for incorporating the influence on subsequent games, which is done by executing actions in the present game, into a traditional framework.

Formalization of a Finite Iterated Game without Change

One of our goals is to find an effective way for an autonomous agent to select an appropriate action under the condition that the agent's payoff depends on both his action and the action of another agent. The difficulty of the problem is that the agent cannot control the behavior of another agent. We have to consider what action is concluded only using the fact that all agents are rational.

In this paper, we limit an interaction between agents to a two-agent relation. If more than two agents exist in the world, we treat their interactions as the accumulation of independent two-agent relations. A game is defined by the tuple of agents, actions and payoff matrices. A set of agents is represented as follows.

$$Agent = \{agent_1, agent_2\}$$

A set of actions is represented as follows.

$$Action = \{Action_1, Action_2\}$$
$$Action_1 = \{a_1, a_2, \cdots, a_k\}$$
$$Action_2 = \{b_1, b_2, \cdots, b_l\}$$

$agent_1$ can execute one of the actions a_i, and $agent_2$ can execute one of the actions b_j. Each agent gets paid off and the value is determined by the pair of actions a_i and b_j. A *payoff* consists of $f_1(a_i, b_j)$ and $f_2(a_i, b_j)$. When $agent_1$ executes a_i and $agent_2$ executes b_j, $agent_1$ receives $f_1(a_i, b_j)$ and $agent_2$ receives $f_2(a_i, b_j)$.

We assume the following.

- Each agent is rational, i.e., each agent executes an action expected to maximize his payoff.
- There is no agreement by which an agent has to select an action. This is called a non-cooperative game.
- Each agent knows the other agent's payoff value, i.e., payoff values are common knowledge.

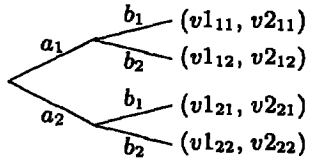
Here, we define a two-player game, G.

Definition: A two-player game G is represented as the tuple of *Agent, Action, and Payoff.* It is the series of getting a task, selecting an action and getting paid off.

A game is represented by a bimatrix (a payoff matrix) or an extensive form. An example of a bimatrix is shown in Figure 1(a). In Figure 1(a), lower left values $v1_{ij}$ are for $agent_1$ and upper right values $v2_{ij}$ are for $agent_2$. The extensive form of the game is shown in Figurel(b).

	b_1	b_2
a_1	$v1_{11}$ $v2_{11}$	$v1_{12}$ $v2_{12}$
a_2	$v1_{21}$ $v2_{21}$	$v1_{22}$ $v2_{22}$

(a) A Payoff Matrix



(b) An Extensive Form

Figure 1: Game Representations

In general, action selection is not done only once. We consider the case where a task is periodically given to each agent. When the task is given, each agent selects an action for the task and gets paid off. This process is repeated N times. N is given at the beginning of the whole game.

We assume the following.

- Each agent has a complete memory of past games.

Here, we define a two-player finite iterated game Γ.

Definition: A two-player finite iterated game Γ is represented by the tuple of *Agent, Action, Payoff, History, and N.* It is a series for the two-player game, G. *History* is a memory of the past games, and N stands for the number of iterations.

Formalization of a Finite Iterated Game with Change

The traditional framework mainly deals with Γ on the condition that the payoff matrix in G does not change while playing Γ. However, this is not always appropriate. Executing an action causes a change in state, and it effects the payoff values. For instance, let's suppose that tasks for transporting goods are given to two agents several times. When an agent encounters an obstacle during transportation, he can either go around the obstacle and leave it or remove it and go straight. If the first detector goes around the obstacle, the other agent has to pay an extra cost for the detour when he passes through there. On the other hand, if the first detector removes the obstacle, the detour cost no longer exists. Therefore, if it is possible to predict an influence by executing an action, it is necessary to deal with the change of payoff values in order to examine the cooperative behavior. In this paper, we only assume the change of payoff values; other factors do not change.

We define a two-player iterated game with change Γc as follows.

Definition: A two-player iterated game with change Γc is represented as the tuple of *Agent, Action, Payoff, History, N, and Change Rules of Payoff.* It is a two-player iterated game with changes in payoff matrix values.

The *Change Rules of Payoff* is represented by the difference between payoff values in the next game and in the present game. A description of effects by actions is another of the *Change Rules of Payoff.* By calculating the new state and calculating the worth and cost of attaining a task in the state, we can obtain new payoff values.

It is true that we can formalize a problem without an explicit description of change by looking at the entire game in advance. However, it is not realistic to enumerate and maintain all sequences of actions if the number of actions and/or iterations increases. Therefore, we need an explicit description of change.

Our claim is that cooperative behavior yields from existence of a relation between the present game and subsequent games. In the next section, we will illustrate our claim.

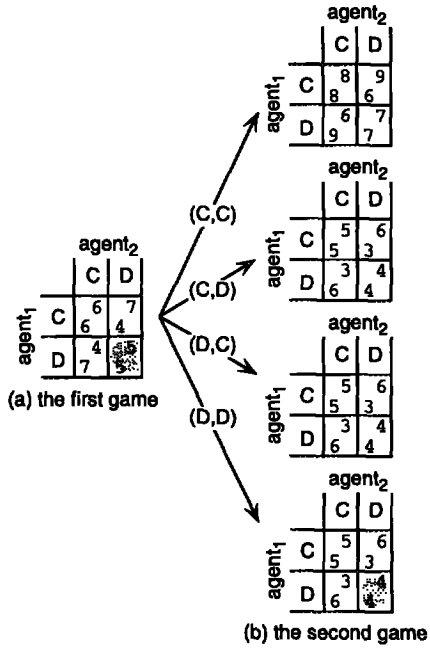


Figure 2: An Example Problem

An Action Selection

In this section, we describe how to select an appropriate action. First, we explain the concept of the Nash equilibrium point. Then, we show three methods for selecting an action.

Solutions in a Two-player Game

What action should an agent select? The concept of the Nash equilibrium point suggests one solution for this question. A Nash equilibrium point is defined as follows.

Definition: If $f_1(a_i^*, b_j^*) = \max_{a_i \in Action_1} f_1(a_i, b_j^*)$ and $f_2(a_i^*, b_j^*) = \max_{b_j \in Action_2} f_2(a_i^*, b_j)$, then (a_i^*, b_j^*) is a Nash equilibrium point.

If $agent_2$ selects an action b_j^* belonging to a Nash equilibrium point, $agent_1$ has no incentive to deviate from selecting an action a_i^* also belonging to it. A Nash equilibrium point is not necessarily Pareto optimal, but it is stable. If both agents are rational, they will select actions belonging to a Nash equilibrium point.

If there is more than one Nash equilibrium point, both agents select the equilibrium point that jointly dominates the other equilibrium points. If there is not such an equilibrium point, more constraints are needed to select a unique solution.

Think Only of the Present

In a two-player iterated game with change Γc, a simple way to select an action is to separately deal with each game.

Think Only of the Present Method (TPM):
Find Nash equilibrium points for the present game, G. Any change of payoff values is not included in the evaluation of the action. If a unique one exists, execute the action belonging to the equilibrium point. If there is more than one equilibrium point, select the dominant equilibrium point. If a dominant one does not exist, select one at random and execute it. If there is no equilibrium point, execute an arbitrary action.

One drawback of this method is missing an action that has a lower payoff in the present, but a higher payoff in the future. Let's assume the iterated game shown in Figure 2. Action C stands for a cooperative action and action D stands for a non-cooperative action. The number of iterations is two. Figure 2(a) shows the first game and Figure 2(b) shows the second game, which is obtained by using *Change Rules of Payoff* and corresponds to each pair of actions in the first game. These types of games are called a prisoner's dilemma, where $v1_{21} > v1_{11} > v1_{22} > v1_{12}$, $v2_{12} > v2_{11} > v2_{22} > v2_{21}$, $2 \times v1_{11} > v1_{21} + v1_{12}$, and $2 \times v2_{11} > v2_{12} + v2_{21}$. TPM gives us the solution (D,D) in the first game and (D,D) in the second game, i.e., both agents always behave non-cooperatively. The total payoff for each is 9. It turns out that in the next subsection, there is another solution belonging to an equilibrium point that dominates the solution obtained by TPM.

Backward Inference

How can we find a sequence of actions that brings a higher payoff in the future? Such a sequence is obtained by backward inference. Backward inference

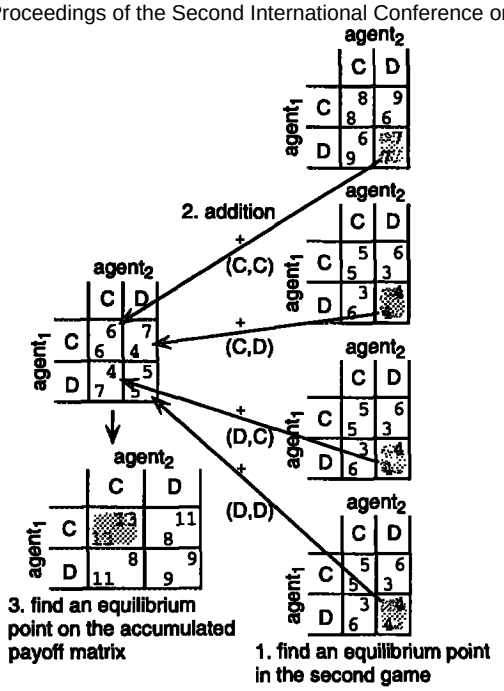


Figure 3: Process for Finding An Action Sequence by BIM

gives us a solution belonging to a subgame perfect equilibrium point. A subgame perfect equilibrium point is a pair of action sequences whereby each agent does not intend at any point to change the latter part of the sequence after an action. If a solution for an n-times iterated game belongs to a subgame perfect equilibrium point, the solution for the last $(n-1)$ part of the game has to belong to a subgame perfect equilibrium point. A different backward inference is used in [Kraus *et al.*, 1995] for other problems.

The backward inference method is as follows.

Backward Inference Method (BIM): Calculate the payoff matrix in the last game by using *Change Rules of Payoff*. Find a Nash equilibrium point for each last game. By adding the value of each equilibrium point in the last game to the payoff values of the $(N-1)$th preceeding game, get an accumulated payoff matrix in the $(N-1)$th game. Find a Nash equilibrium point on the accumulated payoff matrix. Repeat the procedure until the calculation reaches the first game. When this occurs, select the action belonging to a Nash equilibrium point in the payoff matrix accumulated from the last to first games. Select the action already obtained in the succeeding game. When there is no Nash equilibrium point or there is more than one equilibrium point in the game, deal with it like TPM.

We go back to the above example problem. Figure 3 shows the calculation process by BIM. BIM first finds equilibrium points in the second game. By adding each payoff value belonging to an equilibrium point to each corresponding value in the payoff matrix of the first game, we get an accumulated payoff matrix. It contains the sum of payoff values obtained from the whole game. BIM finds equilibrium points on that payoff matrix. BIM gives us the solution (C,C) in the first game and (D,D) in the second game. The total payoff value for each is 12. Each agent does not intend to execute a non-cooperative action in the first game because it brings a higher payoff in the first game, but results in a lower payoff, 11, at the end of the second game. The solution obtained by BIM belongs to a subgame perfect equilibrium point.

In a finite iterated game, the game theory states that rationality does not necessarily yield cooperative behavior. In our example, rationality yields cooperative behavior. That is, the expectation of a higher payoff in a future game yields cooperative behavior. Therefore, in order to understand cooperation existing in the real world, we have to deal with the relations between the present game and future games.

A drawback of this method is that its calculation is time-consuming. The computation cost of this method is $O(k \times l)^{N-1}$, whereas the computation cost of TPM is $O(N)$. This becomes serious if an agent has to execute an action before a deadline.

Look Ahead

We propose a new method in order to overcome the drawbacks of TPM and BIM. Our requests for the method are that an agent can select his action while taking changes of the games into account and that the method is not time-consuming. We define the utility u_1, u_2, and an agent selects an action according to the utility.

Definition: Utility for $agent_1$: $u_1(a_i, b_j) = (1-w) \times v1_{ij} + w \times diff_1(a_i, b_j)$

where w is the weighting factor. $diff_1(a_i, b_j)$ is the average difference between the payoff value in the succeeding game and that in the present game. It is defined as follows:

$$diff_1(a_i, b_j) = \sum_{i,j}(v1'_{ij} - v1_{ij})/(l \times k)$$

where $v1_{ij}$ is a payoff value in the present game and $v1'_{ij}$ is a payoff value in the game after executing a pair of actions a_i, b_j.

The utility for $agent_2$ is defined in the same way.

Look Ahead Method (LAM): Calculate the utility u and obtain a utility matrix in the present game. Find a Nash equilibrium point on the utility matrix. Select an action belonging to the equilibrium point. When there is no Nash equilibrium point or there is more than one equilibrium point in the game, deal with it like TPM.

The aim of this method is to incorporate the influence of changes in the payoff matrix into the action evaluation criterion.

TPM	BIM	LAM
$O(N)$	$O((k \times l)^{N-1})$	$O(N)$

Table 1: Computational Costs of the Three Methods

We go back to the above example problem. When $w = 0.6$, a utility for (C,C) is calculated as follows.

$$u_1(C,C) = (1 - 0.6) \times 6 + 0.6 \times 2 = 3.6$$

When $w = 0.6$, LAM gives us the solution (C,C) in both the first and second games. LAM treats the second game as the game with change. The total payoff value for each is 14. This solution does not belong to an equilibrium point. However, if both agents know that the other agent is using LAM, they find that executing a non-cooperative action in the first game decreases each total payoff. Thus, an agent prefers cooperative behavior in this situation.

Another way for selecting actions by incorporating future payoff into the evaluation is reinforcement learning, such as Q-learning [Watkins and Dayan, 1992]. It is effective if an agent is unable to model the world and the other agent. However, if the agent knows how to change the world and that the other agent is rational, LAM is effective with respect to speed and memory capacity. LAM does not need a learning period and LAM does not need to maintain Q-values.

The computational costs of the three methods are shown in Table 1. If there are some constraints on how to change the payoff values, BIM's computational cost may be reduced as follows. (1) If there is a limit on the amount of accumulated changes, some branches of an expanding tree of payoff matrices are cut by checking the maximum or minimum values accumulated in remaining games. (2) If a way of changing the payoff values does not depend on the previous action sequence, we can obtain some criteria for selecting actions without exhaustive calculation. However, for the worst case, BIM's computational cost is still exponential to the number of iterations.

Evaluation by using a simple model

By using a simple model, we examine when cooperative behavior emerges. Suppose two agents play the game shown in Figure 4. $v1^n_{CC}$ is the payoff value of agent1 for (C,C) in the nth game. We assume each payoff matrix is symmetrical, i.e., $v1^n_{CC} = v2^n_{CC}, v1^n_{CD} = v2^n_{DC}, v1^n_{DC} = v2^n_{CD}, v1^n_{DD} = v2^n_{DD}$. $\alpha_n, \beta_n, \gamma_n$ are the change of payoff values from the nth game to the $(n+1)$th game for (C,C), $\{(C,D),(D,C)\}$, and (D,D), respectively.

When BIM's calculation reaches the n from the last game ($n = N$), holding the following inequalities are requirements for (C,C) to be selected.

$$(v1^n_{CC} + A^{n+1}_{1,CC}) > (v1^n_{DC} + A^{n+1}_{1,DC}) \quad (1)$$

$$(v1^n_{CC} + A^{n+1}_{1,CC}) > (v1^n_{DD} + A^{n+1}_{1,DD}) \quad (2)$$

	C	D
C	$v1^n_{CC}$ $v2^n_{CC}$	$v1^n_{CD}$ $v2^n_{CD}$
D	$v1^n_{DC}$ $v2^n_{DC}$	$v1^n_{DD}$ $v2^n_{DD}$

Figure 4: A Payoff Matrix in the nth Game

$A^{n+1}_{1,CC}$ is an accumulated payoff value for $agent_1$ from the $(n+1)$th game to the Nth game, as (C,C) is executed in the nth game.

For simplicity, we assume that a way of changing payoff values does not depend on the previous action sequence. As such, for whatever action is selected in the nth game, an action sequence after the $(n+1)$th game obtained by BIM is uniquely determined, since the relations among payoff values in each game do not change. Therefore, we obtain the following equations.

$$A^{n+1}_{1,CC} - A^{n+1}_{1,DC} = (N-n)\alpha_n - (N-n)\beta_n$$
$$A^{n+1}_{1,CC} - A^{n+1}_{1,DD} = (N-n)\alpha_n - (N-n)\gamma_n$$

Since we assume that all payoff values in each game change equally, we obtain the following equations.

$$v1^n_{CC} - v1^n_{DC} = v1^1_{CC} - v1^1_{DC}$$
$$v1^n_{CC} - v1^n_{DD} = v1^1_{CC} - v1^1_{DD}$$

The following inequalities are obtained from (1),(2) by using the above four equations.

$$(N-n)(\alpha_n - \beta_n) > v1^1_{DC} - v1^1_{CC} \quad (3)$$

$$(N-n)(\alpha_n - \gamma_n) > v1^1_{DD} - v1^1_{CC} \quad (4)$$

These inequalities elucidate the following about emergence of cooperative behavior.

- As remaining games become longer, the more cooperative actions emerge.
- As a future payoff through a cooperative action becomes larger, the more cooperative actions emerge.
- As an immediate payoff from a non-cooperative action becomes smaller, the more cooperative actions emerge.

Next, we examine the relation between the correctness of action selection and the weighting factor, w.

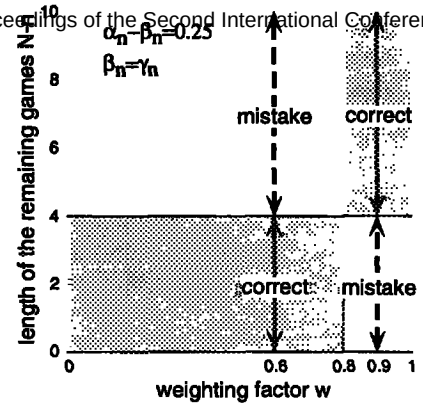
In the nth game, LAM evaluates the following inequalities.

$$(1-w)v1^n_{CC} + w\alpha_n > (1-w)v1^n_{DC} + w\beta_n$$
$$(1-w)v1^n_{CC} + w\alpha_n > (1-w)v1^n_{DD} + w\gamma_n$$

We obtain the following.

$$(\alpha_n - \beta_n - v1^1_{CC} + v1^1_{DC})w > v1^1_{DC} - v1^1_{CC} \quad (5)$$

$$(\alpha_n - \gamma_n - v1^1_{CC} + v1^1_{DD})w > v1^1_{DD} - v1^1_{CC} \quad (6)$$



LAM can select correct actions in shaded regions, but LAM makes a mistake in non-shaded regions.

Figure 5: Relation between the weighting factor and the length of the remaining games

	C	D
C	6, 6	7, 4
D	4, 7	5, 5

(a_i, b_j)	$diff(a_i, b_j)$
(C, C)	$(+0.2, +0.2)$
(C, D)	$(-0.1, -0.1)$
(D, C)	$(-0.1, -0.1)$
(D, D)	$(0.0, 0.0)$

The maximum accumulated value of change is 2.0, and the minimum accumulated value of change is -2.0.

Figure 6: A Prisoner's Dilemma with Changes in the Payoff Values

In order for the solution obtained by LAM to equal the solution obtained by BIM, it is necessary to either simultaneously maintain or not maintain the above inequalities (3),(4),(5), and (6).

The relation between w and $N-n$ is shown in Figure 5 for $\alpha_n - \beta_n = 0.25, \beta_n = \gamma_n, v1^1_{DC} - v1^1_{CC} = v1^1_{CC} - v1^1_{DD} = 1$. If $w = 0.6$, LAM makes a mistake in selecting an action when $N-n > 4$ because it underestimates the future payoff. If $w = 0.9$, LAM makes a mistake when $N - n \leq 4$ because it overestimates the future payoff. The appropriateness of w in each game decides whether or not LAM is able to select the correct actions.

Case Analysis

In this section, we apply the three methods to three problems and evaluate the properties of these methods.

Case 1

We show that incorporating the influence of game changes into the evaluation of actions yields cooperative behavior. Figure 6 shows the problem. There are two agents in the environment. A task is given to each agent 14 times. The payoff values in Figure 6 are initial values, which are calculated from the worth of the task and the cost. Each agent has two ways of attaining the task. One is to cooperate with the other agent, and the other is to be non-cooperative. Figure 6 also represents the change of payoff values. For instance, if both agents execute cooperative actions, the payoff value increases by 0.2 for each. The upper and lower limits of accumulated changes are 2.0 and -2.0, respectively.

Number of iterations	TPM	BIM	LAM
8	0.07	0.22	0.07
10	0.07	2.46	0.07
12	0.07	39.52	0.07
14	0.07	639.51	0.07

Table 2: CPU Times for the Three Methods (unit: [s])

We evaluated the performance of each method by the average payoff value. In TPM, the agent executed no cooperative actions. In BIM, the agent executed cooperative actions between the first game and the 9th game and non-cooperative actions in the remaining games. In LAM ($w = 0.95$), the agent always executed cooperative actions.

Figures 7(a),(b),(c) show results when the other agent uses TPM, BIM, and LAM, respectively. The x-axis represents the progress of the game and the y-axis represents the average payoff value. Each point in Figure 7 shows the average payoff value at that time. Figure 7(b) tells us that the action sequence obtained by BIM finally dominates the other two methods. It belongs to the subgame perfect equilibrium point, i.e., if the agent knows that the other agent is using BIM, a deviation from the sequence obtained by BIM decreases his average payoff value. Figure 7(c) shows that the action sequence obtained by LAM finally dominates that of TPM. It does not belong to an equilibrium point. However, if an agent knows that the other agent is using LAM and the value of weighting factor w, he understands that he should use LAM instead of TPM.

The drawback of TPM is that it does not yield cooperative behavior. Therefore, TPM brings a considerably lower average payoff in comparison with BIM and LAM. The drawback of BIM is that it is time-consuming. CPU times are shown in Table 2. If the number of iterations increases, BIM cannot solve the problem in a reasonable time. In this situation, LAM becomes useful than BIM.

Case 2

We will solve a problem that is the same as in Case 1, except for change rules of payoff values. The payoff value increases by 0.4 for (C,C), (C,D), and (D,C) and decreases by 0.4 for (D,D). In this game, cooper-

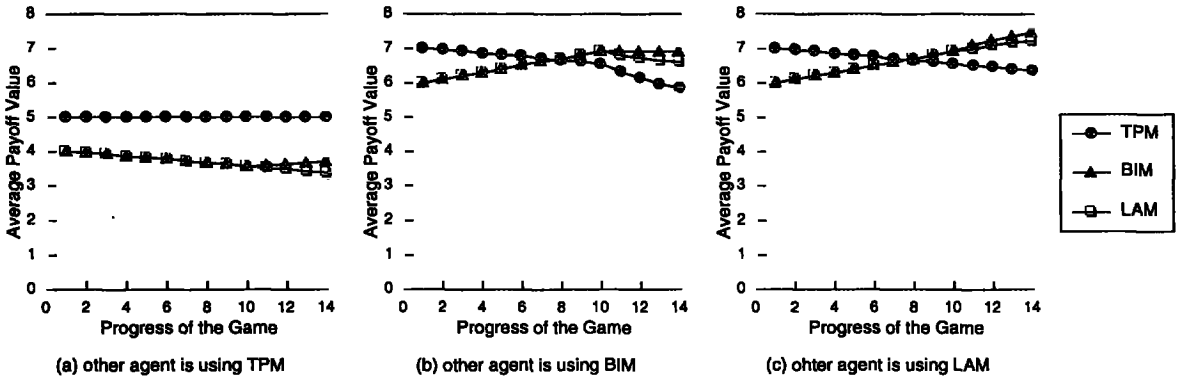


Figure 7: Results by the three methods: 1

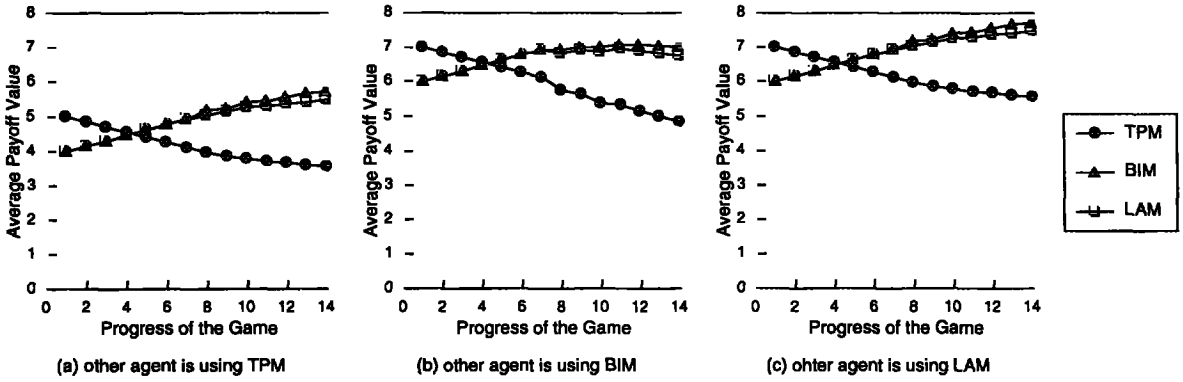


Figure 8: Results by the three methods: 2

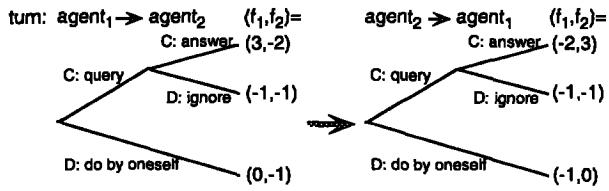
ative behavior, regardless of the method used by the other agent, always brings a favorable situation in the subsequent game.

Figure 8 shows the results. Figure 8 (a) shows that the average payoff value by BIM finally exceeds that by TPM. In this game, even if the agent does not know the method used by the other agent, he is motivated to execute a cooperative action. An action sequence obtained by BIM in Case 2 has stronger stability than that in Case 1.

Case 3

The third problem is shown in Figure 9. Each agent alternatively takes the initiative. An agent who has the initiative determines whether he will solve the task alone or ask the other agent for some information useful for solving the problem. When the other agent is asked for information, he determines whether he will give his information to the agent or ignore the request. Each agent solves a task seven times.

In LAM, as w increases, the more cooperative an agent's behavior becomes. Figure 10 shows the results for when the agent who had the initiative in odd num-



(a_i, b_j)	$diff(a_i, b_j)$
(C, C)	$(0.0, +0.4)$
(C, D)	$(0.0, -0.4)$
$(D, *)$	$(0.0, -0.4)$

(a_i, b_j)	$diff(a_i, b_j)$
(C, C)	$(+0.4, 0.0)$
(C, D)	$(-0.4, 0.0)$
$(*, D)$	$(-0.4, 0.0)$

The maximum accumulated value of difference is 0.8, and the minimum accumulated value of difference is -0.8.

Figure 9: An Alternate Query and Answer Problem

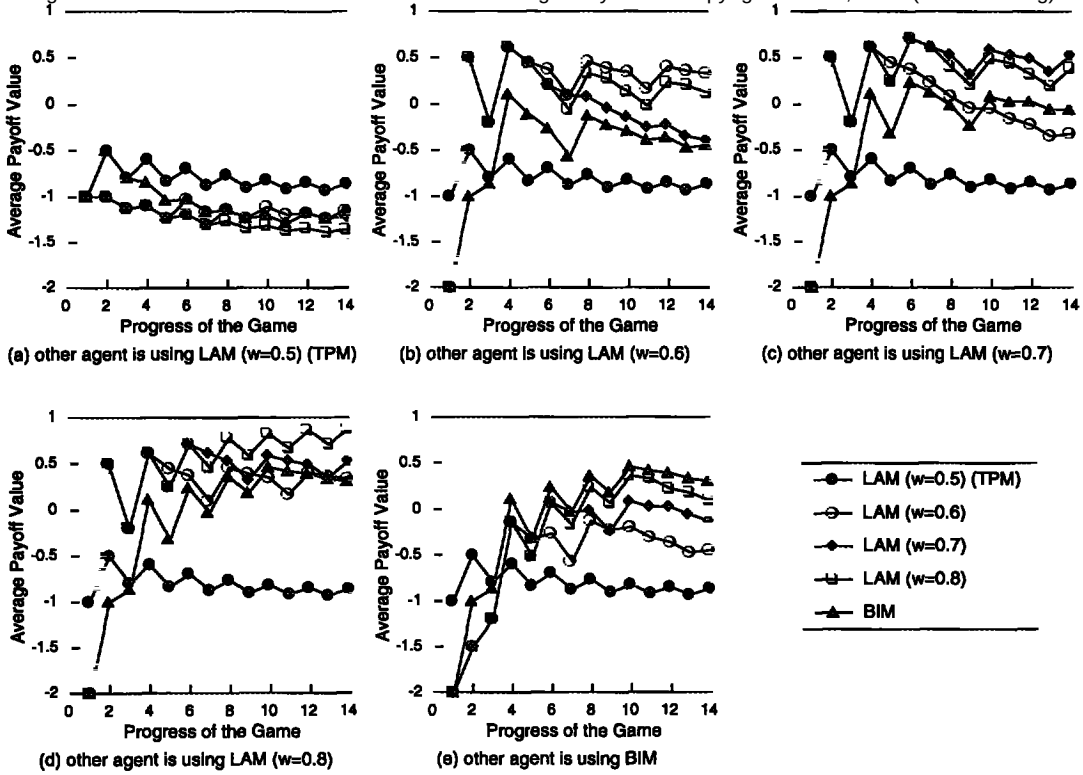


Figure 10: Results by the three methods: 3

ber games uses TPM, BIM, and LAM with $w = 0.5$, 0.6, 0.7, and 0.8. The average payoff value shown in Figure 10 is for the agent who had the initiative in the even number games. LAM with $w = 0.5$ does not yield as cooperative behavior as TPM. LAM with $w = 0.8$ does not yield non-cooperative behavior. We can change which action the agent selects by tuning the weighting factor, w. Each figure tells us that using the same method as the other agent brings the highest average payoff. Knowing the method used by the other agent brings much profit, so telling the other agent the method used by oneself brings much profit.

Conclusions

We have shown the formalization of a finite iterated game with changes, and the influence regarding changes of payoff values in an iterated game have been incorporated into the traditional framework. The formalization enables us to answer the question, "Why does cooperative behavior emerge in human interactions, even though from a myopic view, cooperative behavior does not seem to be profitable?" The answer is that the expectation of a greater payoff at some future point yields cooperative behavior. Furthermore, we have proposed a new method for selecting an action in such a framework. It is called the Look Ahead Method (LAM), and it overcomes the drawbacks of previous methods. Our proposed method can yield cooperative behavior and is not time-consuming.

Our future work is to investigate agents' behavior in iterated games with more complicated patterns of Change Rules of Payoff.

Acknowledgments

The authors wish to thank Ko-ichi Matsuda and Nobuyasu Osato for their support during this work at NTT Laboratories.

References

Genesereth, M. R., Ginsberg, M. L., and Rosenschein, J. S., "Cooperation without Communication," *AAAI-86*, pp. 51-57, 1986.

Kraus, S., Wilkenfeld, J., and Zlotkin, G., "Multiagent negotiation under time constraints," *Artificial Intelligence*, 75, pp. 297-345, 1995.

Mor, Y. and Rosenschein, J. S., "Time and the Prisoner's Dilemma," *ICMAS-95*, pp. 276-282, 1995.

Rosenschein, J. S., and Zlotkin, G., "Rules of Encounter," The MIT Press, 1994.

Watkins, C. J. C. H. and Dayan, P., "Technical Note: Q-Learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.