T. Sudo, K. Goto, Y. Nojima, and H. Ishibuchi, "Further analysis on strange evolution behavior of 7-bit binary string strategies in iterated prisoner's dilemma game," *Proc. of 2016 IEEE Congress on Evolutionary Computation (Accepted).*

Further Analysis on Strange Evolution Behavior of 7-bit Binary String Strategies in Iterated Prisoner's Dilemma Game

Takahiko Sudo, Kazushi Goto, Yusuke Nojima, and Hisao Ishibuchi
Department of Computer Science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
{takahiko.sudo@ci., kazushi.goto@ci., nojima@, hisaoi@}cs.osakafu-u.ac.jp

***Abstract*—Evolution of cooperation has been actively studied in the evolutionary computation (EC) community mainly for the iterated prisoner's dilemma (IPD) game. One of the frequently examined settings is a noisy environment where a player chooses a different action from the suggested one by its strategy with a pre-specified error probability. The use of the error probability in the IPD game usually makes the evolution of cooperation very difficult. This is because occasional defection by error disturbs mutual cooperation. In our former study, we examined the effect of the error probability on the evolution of cooperation among players with binary string strategies under various settings of memory length. Then we found that a higher average payoff was obtained by increasing the error probability from zero to a small value only when we used 7-bit binary string strategies with a memory about the opponent's previous two actions. That is, a small error probability helped the evolution of cooperation only under this particular setting. This behavior was not observed in the other settings of memory usage. In this paper, we further examine this behavior through computational experiments for a wide variety of settings of various factors. Experimental results show that this behavior is observed only under special settings of the following factors: the number of players (i.e., population size), memory length, memory usage, and a crossover operator.**

Keywords—Iterated prisoner's dilemma (IPD); evolutionary games; game strategies; evolution of cooperation; error probability.

I. INTRODUCTION

The prisoner's dilemma (PD) game is one of the most well-known and frequently discussed non-zero-sum games. A player and its opponent are supposed to decide whether to cooperate or defect simultaneously. Its iterated version is called the iterated prisoner's dilemma (IPD) game. The IPD game has frequently been studied in many areas [1-3]. A player decides the next action based on its own strategy with a memory about the player's and/or the opponent's previous actions. It has been reported that the evolution of cooperation depends on a variety of factors [4]-[7]. One important factor is the memory length and the memory usage [7].

In our previous study [8], we examined the evolution of cooperative strategies in a noisy setting where a player chose a different action from the suggested one by its strategy with a pre-specified error probability. In general, the error in action selection makes the evolution of cooperation difficult because a player (i.e., player's strategy) cannot distinguish it from the defection by the opponent's strategy. We investigated the relation between the error probability and the evolution of binary string strategies with a different memory length about the opponent's previous actions. Only when we used a 7-bit strategy with a memory about the opponent's previous two actions, a clear increase in the average payoff was observed by the increase of the error probability from zero to a small positive value. This behavior was not observed by the other binary string strategies. We also analyzed this strange behavior of the 7-bit strategy by observing a single run in detail.

In [8], we also examined the effect of the number of players on the above-mentioned behavior of the 7-bit strategy. This behavior was not clearly observed when the number of players was less than 300. However, other settings were not examined in [8] in detail. For example, strategies with a memory about the player's own actions were not examined. Only one-point crossover was used as crossover operator in [8]. In this paper, we examine the effects of various factors on the emergence of the strange behavior observed in [8]. In addition, we also examine another noisy setting where errors occur in a player's memory about the previous actions in the IPD game. We show the effects of a wide variety of settings on the strange behavior of the 7-bit strategy by carefully observing each individual run. We focus on the following four parameters:

(a) the number of players,
(b) the usage of memory resources,
(c) the choice of the crossover operators,
(d) the error in a player's memory.

This paper is organized as follows. In Section II, we explain our IPD model where we use a standard payoff matrix. Next we explain the strange behavior of 7-bit binary string strategies in Section III. Section IV explains how our binary string strategies decide the player's action and describes a genetic algorithm for the evolution of those strategies. We report experimental results under various settings of the above-mentioned four factors in Section V. Finally, we conclude this paper in Section VI.

II. ITERATED PRISONER'S DILEMMA GAME

The PD game is a well-known two-player game. In PD games, a player and its opponent decide either "C: Cooperation" or "D: Defection", simultaneously. They receive a payoff from the payoff matrix in Table I. The defection is a rational action selection for both of the player and the opponent. This is because a higher payoff is obtained by defecting than by cooperating independent of the opponent's action selection in Table I. When the opponent chooses cooperation, the player receives payoff 5 by defecting and payoff 3 by cooperating. When the opponent chooses defection, the player receives payoff 1 by defecting and payoff 0 by cooperating. However, the average payoff 1 by mutual defection is the worst for them. The average payoff 3 by mutual cooperation is the best for them. This is the dilemma in Table I.

TABLE I. PAYOFF MATRIX OF OUR IPD GAME

Player's Action	Opponent's Action	
	C: Cooperation	D: Defection
C: Cooperation	Player Payoff: 3 Opponent Payoff: 3	Player Payoff: 0 Opponent Payoff: 5
D: Defection	Player Payoff: 5 Opponent Payoff: 0	Player Payoff: 1 Opponent Payoff: 1

In an IPD game, the player and the opponent iterate a PD game for the pre-specified number of rounds. From Table I, an average payoff close to 3.0 means that mutual cooperation is achieved for almost all rounds. On the other hand, an average payoff close to 1.0 means that mutual defection is achieved for almost all rounds.

III. STRANGE BEHAVIOR OF 7-BIT STRATEGIES

Errors in action selection have been examined in many studies [8, 9]. In a noisy setting, a player chooses the different action from the suggested one by its strategy with the pre-specified error probability. We illustrate the difficulty of mutual cooperation in a noisy setting by using the Tit-for-tat (TFT) strategy which cooperates in the first round and then chooses the opponent's previous action. When the TFT strategy plays the IPD game in the noise-free setting where the error probability is zero, the two TFT strategies play cooperation in all rounds of the IPD game (i.e., 100% mutual cooperation). However, in a noisy setting, (D, C) and (C, D) are repeated after one player defects by error.

In our previous study [8], we examined the relation between the error probability and the evolution of six types of binary string strategies with a memory about the opponent's previous actions. Fig. 1 shows the average payoff of 7-bit strategies with a memory about the opponent's previous two actions (i.e., memory length 2) at the 1000th generation over 50 runs. This result was obtained under the following setting: the number of players was 1600, one-point crossover was used to generate offspring, and a spatial structure was not used to define a player's neighbors. The average payoff with the error probability zero was 2.35. The highest average payoff 2.86 was obtained with a small error probability. The difference between these two settings was statistically significant (Mann-Whitney U test, $p < 0.001$). In the other five binary string strategies with memory length 1, 3, 3, 5 and 6 (string length 3, 15, 31, 63 and 127), the highest average payoff was always obtained when the error probability was specified as zero. Therefore, the relation between the evolution of 7-bit binary string strategies and the error probability looks strange.

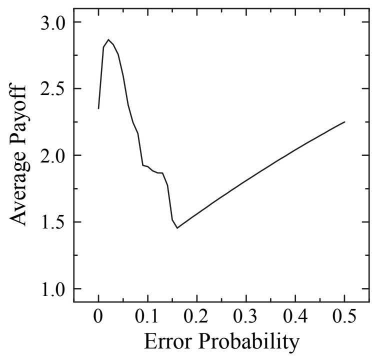


Fig. 1 Average payoff of 7-bit strategies at the 1000th generation of the non-spatial IPD model where all the players are neighbors of each player (Fig. 3 (b) in [8]).

We analyzed this strange behavior of 7-bit strategies in detail. We investigated the difference between a single run with the error probability 0 and 0.02. We reported all strategies evolved to choose defection and cooperation periodically with the error probability 0. These strategies led to the average payoff 2.0. Once all strategies converge to such a strategy, it is difficult to evolve toward the mutual cooperation (or the mutual defection). The same situation was also observed with the error probability 0.02. That is, all strategies repeated defection and cooperation in an IPD game. However, the small error probability often changed the situation toward the mutual cooperation only when 7-bit strategies was used. It seems from those experimental results that a small error probability encourages the evolution of cooperation.

IV. STRATEGY REPRESENTATION AND UPDATE

A. Binary String Strategies

The choice of the representation scheme is the important factor in evolutionary IPD game strategies. A variety of representation schemes were studied as the player's strategies [5, 6]. In this paper, we use binary string strategies to determine the player's action. Table II shows a 7-bit binary string strategy with a memory about the opponent's previous two actions. "C" and "D" in Table II mean cooperation and defection, respectively. The first bit x_1 is used to determine the player's action in the first round. x_2 and x_3 are used to determine the player's action in the second round. The other four bits $x_4x_5x_6x_7$ are used to determine the player's actions in the subsequent rounds. For example, when the opponent's action in the first round is "D", x_2 is used to determine the player's action in the second round. When the opponent's action is "D" in the first round and "C" in the second round, x_5 is used to determine the player's action in the third round.

TABLE II. BINARY STRATEGY OF LENGTH 7

Opponent's action at the (t-2)th round	-	-	-	D	D	C	C
Opponent's action at the (t-1)th round	-	D	C	D	C	D	C
Player's action at the t-th round	x_1	x_2	x_3	x_4	x_5	x_6	x_7

B. Strategy Update

In our computational experiments, each player has a binary string strategy. Each strategy is initialized by randomly assigning 0 (i.e., "D") or 1 (i.e., "C") to each bit of the strategy.

Each player randomly selects four opponents without replacement from all players except for the player itself. The player plays IPD games against each of the selected opponents. The PD game is iterated between the player and the opponent for 100 rounds. The fitness of the player is the average payoff per round obtained by the IPD game against all the opponents.

Each player selects two parents by binary tournament selection with replacement from all players including the player itself. A crossover is applied to the strategies of the selected parents to generate offspring with probability 1.0. Then, bit-flip mutation is applied to each bit of the offspring with the probability $1/(LN)$ where L is the bit length of the strategy and N is the number of players. A single bit of all players is flipped by mutation on average. After all players generate their new strategies, they replace their strategies with the new one simultaneously.

The evolution of strategies is terminated after the pre-specified number of generations. The number of generations is specified as 1000 in our computational experiments.

V. Computational Experiments

We examine the effect of the choice of parameters. We use the following settings [8] in our computational experiments.

[Setting of the IPD game]
- The number of players (N): 100, 1024, 1600,
- Spatial structure: Non-spatial,
- Error probability: 0.00, 0.01, 0.02, ..., 0.20,
- The number of opponents for each player: 4,
- The number of rounds in the IPD game: 100.

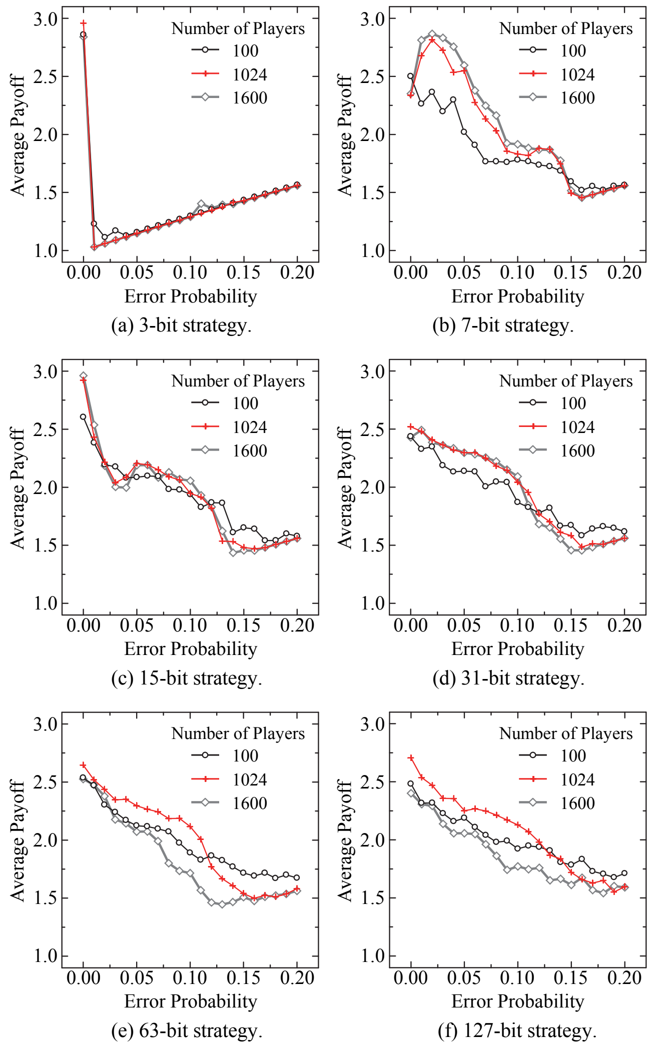
[Setting of the genetic algorithm for strategy evolution]
- Coding of strategies: binary strings,
- String length (L): 3, 7, 15, 31, 63, 127,
- Parent selection: binary tournament selection,
- Crossover: one-point crossover, two-point crossover, uniform crossover,
- Mutation: bit-flip mutation with the probability: $1/(LN)$ (L: string length, N: the number of players),
- Number of runs for each setting: 50 runs.

In [8], we examined the evolution of cooperation under the non-spatial setting of players as well as the grid-world setting where the neighbors of each player were defined by the two-dimensional grid-world. The strange evolution behavior was not observed under the grid-world setting. Moreover, high error probabilities discouraged cooperation even if we used 7-bit strategies. In this paper, we only consider the non-spatial structure and low error probabilities (i.e., not more than 0.20).

A. The Effect of the Number of Players

In [8], we briefly investigated the effect of the number of players on the behavior of 7-bit binary string strategies. The strange behavior was observed when we used more than 200-300 players. In this subsection, we compare the evolution of cooperation by specifying the number of players as 100, 1024 and 1600.

First, we compare the behavior of 7-bit strategies with other strategies used in [8]. Fig. 2 shows the results of the average payoff at the 1000th generation over 50 runs. When we used 1024 and 1600 players, we observed the strange behavior. However, when we used 100 players, the average payoff in the noise-free setting is higher than that in the noisy settings (Fig. 2 (b)). Moreover, no strange behavior was observed when we used binary string strategies of the length 3, 15, 31, 63 and 127 independent of the number of players (Fig. 2 (a) and (c)-(f)). Further investigation is needed to explain why the small player size disturbs the strange evolution behavior of the 7-bit strategy.



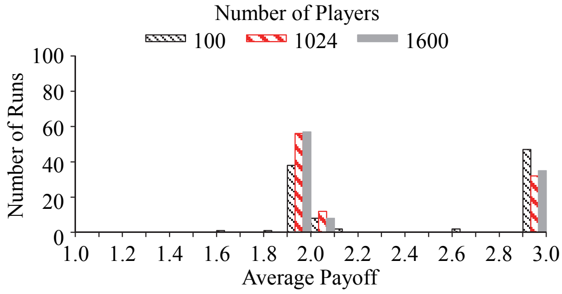
(a) 3-bit strategy. (b) 7-bit strategy. (c) 15-bit strategy. (d) 31-bit strategy. (e) 63-bit strategy. (f) 127-bit strategy.

Fig. 2 Effects of the player size with a different string length.

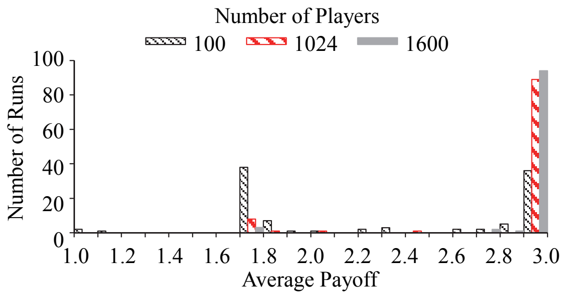
Fig. 3. shows the histograms of the average payoff at 1000th generation obtained by each of 100 runs for the two settings of the error probability: 0 and 0.02. In Fig. 3 (a), the average payoff of each run is divided into two groups: One is close to 2.0 and the other is close to 3.0. The average payoff of

the former group is about 1.8 in the case of the error probability 0.02 in Fig. 3 (b).

In Fig. 3 (a), the average payoff close to 3.0 was obtained from 47, 32 and 35 runs when the number of players was 100, 1024 and 1600, respectively. In Fig. 3 (b), the average payoff close to 3.0 was obtained from 36, 89 and 94 runs when the number of players was 100, 1024 and 1600, respectively. When we used 1024 and 1600 players, the number of runs in which the average payoff about 3.0 was obtained was much larger in the case of the error probability 0.02 than the case of the error probability 0. However, in our computational experiments with 100 players, the number of those runs was less than 50 in both cases of the error probability: 0 and 0.02.



(a) Error probability 0.



(b) Error probability 0.02.

Fig. 3. Histograms of the average payoff with 100, 1024 and 1600 players.

Let us analyze the evolution of cooperation of 100 players in a single run with the error probability 0.02 in detail. In Fig. 4, the average payoff continued to be close to 3.0 for about 700 generations. While the average payoff rapidly decreased around the 5th, 220th, 330th and 550th generations, it rapidly recovered from those drops. However, after the drop around the 700th generation, the average payoff converged to 1.8.

Fig. 5 shows the relation between the average payoff and the percentage of some representative strategies around the 675th generation. The average payoff continued to be close to 3.0 and many players had the strategy 0100111 or the strategy 0101111 until about the 640th generation in Fig. 5. The strategy 0100111 cooperates unless the opponent defects twice in a row after the third round in the IPD game. The strategy 0101111 cooperates independent of the opponent's action after the third round. The probability of the defection twice in a row by error is 0.02 x 0.02 = 0.0004. That is, almost all rounds in the IPD game were mutual cooperation since almost all players had one of the two strategies. After the 640th generation, the percentage of the strategy 0101111 increased as that of the strategy 0100111 decreased. The strategy 0101111 can obtain high payoff from a cooperator (e.g., strategy 0100111) while it is exploited by a defector (e.g., strategy 0000000).

After the increase in the percentage of the strategy 0101111, the average payoff dropped sharply to about 1.2 around the 675th generation. At the same time, the strategy 0100000 rapidly increased. This strategy defects in at least 99 rounds in the IPD game with 100 rounds. Since this strategy exploits the strategy 0101111, the percentage of this strategy rapidly increased around the 675th generation. When the number of players was 1600, the percentage of the strategies whose last four bits were 1111 was at most about 14% as reported in [8]. Since the number of these strategies is small, the evolution of defection (i.e., rapid increase of defective strategies such as 0100000) is difficult in this situation. When the number of players is small, the diversity of strategies is small (i.e., there exist at most 100 different strategies among 100 players). So, a strategy used by a small number of players may have a larger influence and can spread over the entire population more easily than the case with a large number of players.

After the 675th generation, the percentages of the strategies 0100010 and 0110010 increased. It is likely that 0110010 was generated by mutation from 0100010. These strategies choose defection and cooperation alternately in a synchronized manner and receive the average payoff 2.0 from the IPD game between them in the noise-free setting. Those strategies were observed in [8] when we used 1600 players.

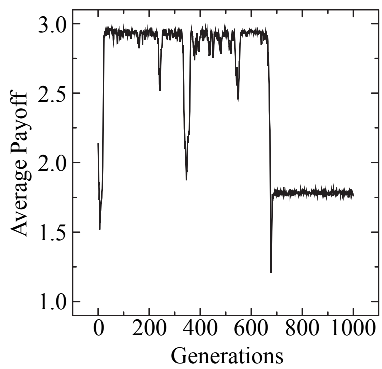


Fig. 4. The average payoff by a single run of 100 players with the error probability 0.02.

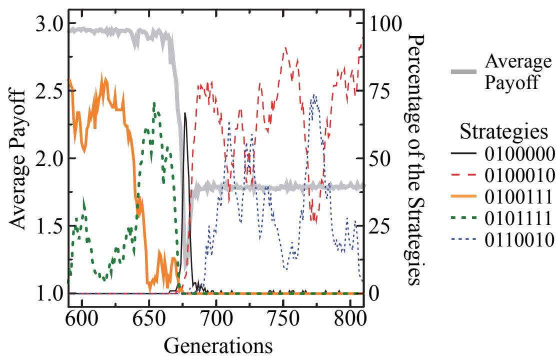


Fig. 5. The relation between the average payoff and the percentage of strategies around the rapid drop of the average payoff in Fig. 4.

Since similar results were obtained from the two settings of the number of players (i.e., 1024 and 1600) in Fig. 2 and Fig. 3, we report experimental results from the setting with 1024 players in the following subsections.

B. The Choice of the Usage of Memory Resources

We examine the effect of the different memory usage. The memory length of the strategy has a large effect on the evolution of cooperative strategies. The usage of memory resources is also an important factor [7]. For example, the following two strategies have the same memory length but different memory usage: one with a memory about the opponent's previous three actions, and the other with a memory about the player's previous single action and the opponent's previous two actions. A 7-bit binary string strategy decides the action based on the opponent's previous two actions. In addition to this type of 7-bit binary string strategies, there exist the following two types of strategies with the memory length two. One is 5-bit strategies with a memory about the player's and opponent's actions in the previous single round. The other is 7-bit strategies with a memory about the player's previous two actions. The action selection in the IPD game with no memory about the opponent's actions seems to be the same as in a non-iterated PD game. Since the defection is a rational action, such a strategy usually converges to the mutual defection. So, we do not use those strategies with no memory about the opponent's actions in this paper.

In this subsection, we denote the type of binary string strategies by their memory length about the opponent's actions and the player's own actions. This is because some different types of strategies have the same bit length. Strategies with a memory about the opponent's m actions and the player's n actions are represented as "OmPn" strategies. For example, "O2P0" are 7-bit strategies based on the opponent's previous two actions, and "O1P1" are 5-bit strategies based on the opponent's and player's actions in the previous single round.

Fig. 6 shows the result of the average payoff at the 1000th generation over 50 runs. We compare O2P0 (7-bit) strategies and O1P1 strategies with the memory length two (Fig. 6 (a)). The strange behavior was not observed by O1P1 strategies. We also compare the strategies with the memory length three (Fig. 6 (b)). O2P1 strategies have a memory about the opponent's previous two actions. The memory length about opponent's actions is the same as O2P0 strategies. However, the strange behavior was not observed. The average payoff of O1P2 strategies was increased by the small increase of the error probability. However, the difference of the average payoff was small (i.e., the average payoff 2.67 with the error probability 0 and the average payoff 2.76 with the error probability 0.03). Fig. 6 (c) shows the results of strategies with the memory length four. The strange evolution behavior was not obtained by them including O2P2 strategies with a memory about the opponent's previous two actions. From Fig. 6, we can see that strategies with the memory length two in total or the memory about the opponent's previous two actions did not always show the strange behavior. The strange behavior was clearly observed only when we used 7-bit strategies with the memory length two and the memory about the opponent's previous two actions.

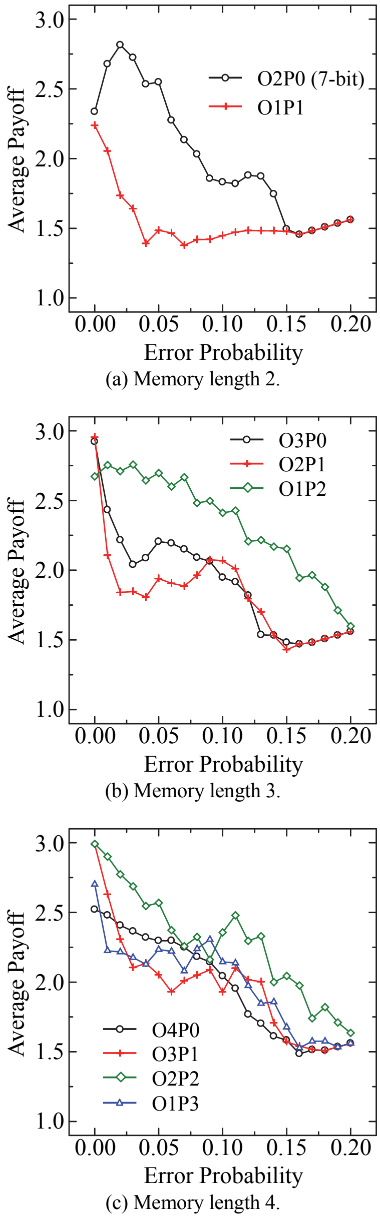


Fig. 6. Average payoff by the strategies with the memory length two, three, and four.

C. The Effect of the Choice of the Crossover Operators

In this subsection, we focus on the effect of the choice of a crossover operator. In [8], the one-point crossover was used. For examining the effect of the use of the one-point crossover, we examined the use of the following three crossover operators: one-point, two-point and uniform crossover operators. Experimental results are summarized in Fig. 7 where the average payoff at the 1000th generation over 50 runs is shown for each crossover operator and various settings of the error probability between 0 and 0.2.

In Fig. 7, no clear difference was observed among the three crossover operators when the error probability was larger than 0.02. The average payoff decreased with the increase of the error probability from 0.02 to 0.15. However, the average payoff increased with the increase of the error probability in [0.15, 0.20]. This is because the choice of cooperation by error increased the average payoff. When the error probability was 0, the average payoff 2.34 by the one-point crossover was much smaller than that by the other crossover operators (i.e., the average payoff 2.76 by the two-point crossover and the average payoff 2.65 by the uniform crossover). When we used the two-point and uniform crossover operator, we observed small increases in the average payoff by slightly increasing the error probability from 0. The highest average payoff of the two-point crossover was 2.77 with the error probability 0.02 and the highest average payoff of the uniform crossover was 2.80 with the error probability 0.01. The strange evolution behavior was not clearly observed by the two-point crossover and the uniform crossover.

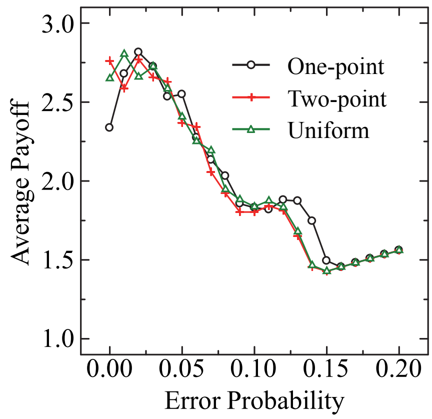


Fig. 7. Average payoff of the 7-bit strategy by using one-point crossover, two-point crossover and uniform crossover.

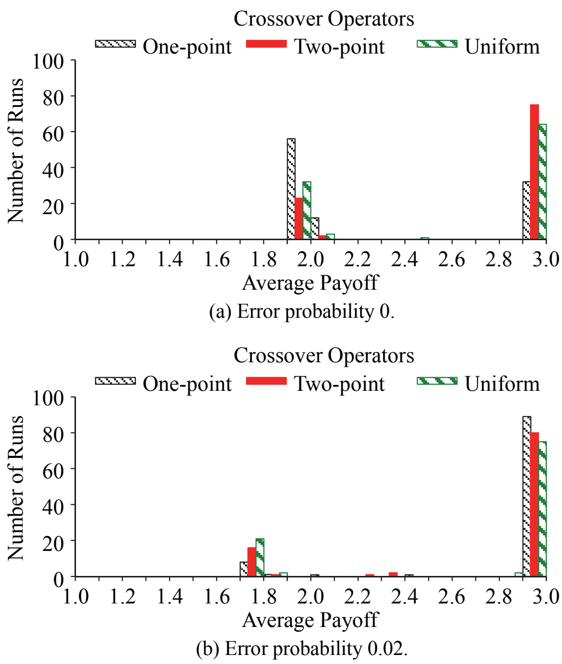


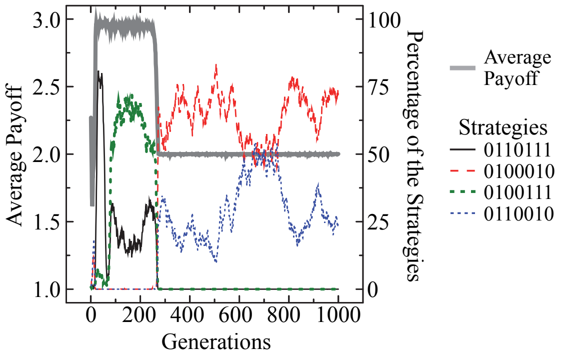
Fig. 8. Histograms of the average payoff of the one-point crossover, the two-point the crossover and uniform crossover.

Fig. 8 shows the histogram of the average payoff at the 1000th generation obtained by each of 100 runs. The average payoff of each run is divided into two groups: One is close to 2.0 and the other is close to 3.0. In Fig. 3 and Fig. 8, the average payoff of each run was always about 2.0 or 3.0 in independent of the parameter settings. However, the number of runs with the average payoff 3.0 depends on the parameter settings. In Fig. 8 (a) with the error probability 0, the average payoff close to 3.0 was obtained from 32, 75 and 64 runs with the one-point, two-point and uniform crossover, respectively. This observation suggests that the one-point crossover is likely to prevent the convergence to the mutual cooperation with the average payoff 3.0. However, in Fig. 8 (b) with the error probability 0.02, the average payoff close to 3.0 was obtained from 89, 80 and 75 runs with the one-point, two-point and uniform crossover, respectively. Many runs converged to the mutual cooperation independent of the choice of a crossover operator. Figs. 7 and 8 suggest that the one-point crossover may have a large effect on the strange behavior.

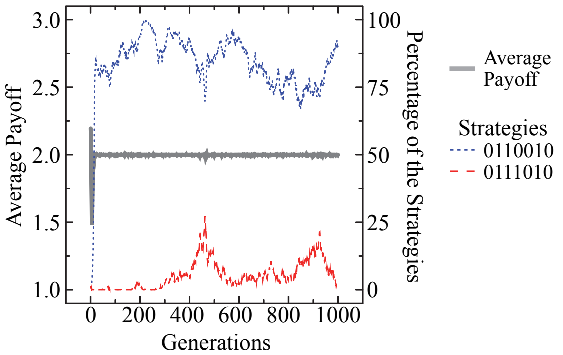
Further we carefully examined each individual run. Fig. 9 shows experimental results of a single run with the average payoff 2.0 at the 1000th generation and the error probability 0. Fig. 9 (a) is the results with the one-point crossover while Fig. 9 (b) is the result with the uniform crossover. In each graph in Fig. 9, the total percentage of the strategy 0110010 and the strategy 0100010 continued to be almost 100% after the average payoff converged to 2.0. In [8], we explained how these two strategies work in the strange behavior.

Fig. 10 shows experimental results of a single run with the average payoff 3.0 at the 1000th generation and the error probability 0. In Fig. 10 (a), the percentages of strategies with 0 in their first bit were high. That is, many players choose defection in the first round of the IPD game. However, in Fig. 10 (b), the percentages of the strategies with 1 in their first bit were high. That is, many players choose cooperation in the first round of the IPD game. In Fig. 10 (a) and Fig. 10 (b), the average payoff sharply dropped (i.e., moved toward the mutual defection) in an early stage of evolution. After that, strategies quickly converged to the mutual cooperation. During the quick convergence to the mutual cooperation, the binary string with the first bit 0 seems to remain by the one-point crossover. If the first bit is 0 and the second, fifth, and seventh bits are 1 (i.e., 01**1*1: "*" means "don't care"), the mutual cooperation is achieved in 99 rounds in the IPD game among those strategies. Those strategies were obtained in Fig 10 (a) (e.g., 0110111). The first bit 0 (i.e., the defection in the first round) seems to make the evolution of cooperation unstable.

Fig. 11 and Fig. 12 show experimental results of a single run with the error probability 0.02. In Fig. 11 with the average payoff about 1.8 at the 1000th generation, almost 100% of strategies was 0110010 at the 1000th generation while the strategy 0110111 had a high percentage before the sharp drop of the average payoff from about 3.0 to about 1.8. However, in Fig. 12 with the average payoff about 3.0 at the 1000th generation, the percentage of the strategy 0110111 continued to be high in almost all generations. Whereas there was some sharp drops of the average payoff (and the percentage of the strategy 0110111) in Fig. 12, the average payoff (and the percentage of the strategy 0110111) quickly recovered.

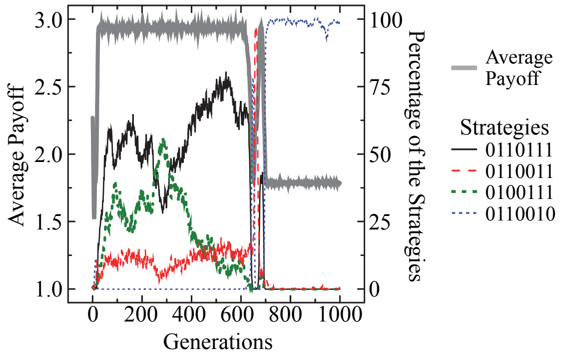


(a) One-point crossover.

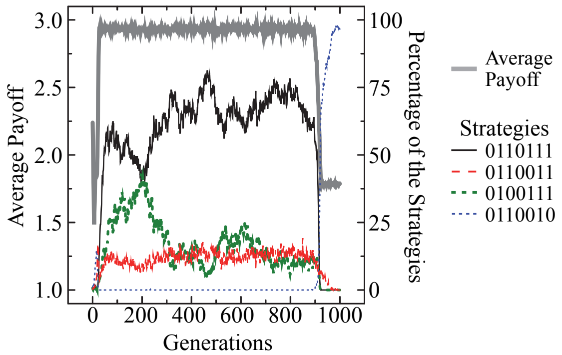


(b) Uniform crossover.

Fig. 9. Average payoff 2.0 with error probability 0.

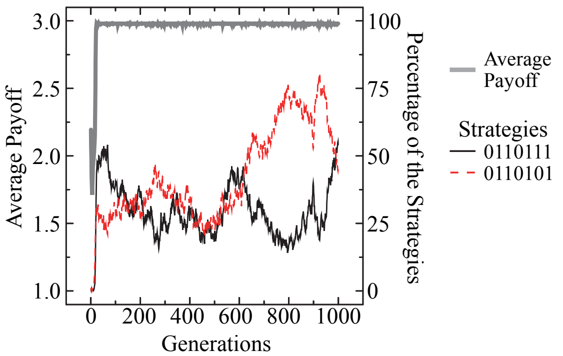


(a) One-point crossover.

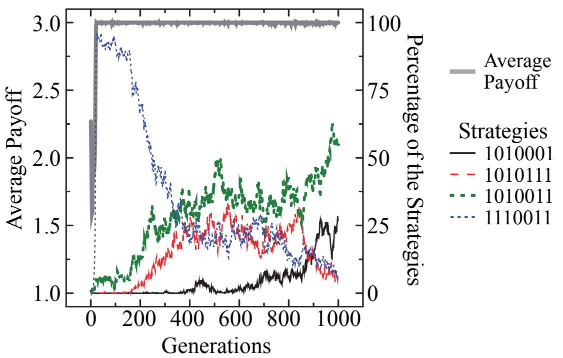


(b) Uniform crossover.

Fig. 11. Average payoff 1.8 with error probability 0.02.

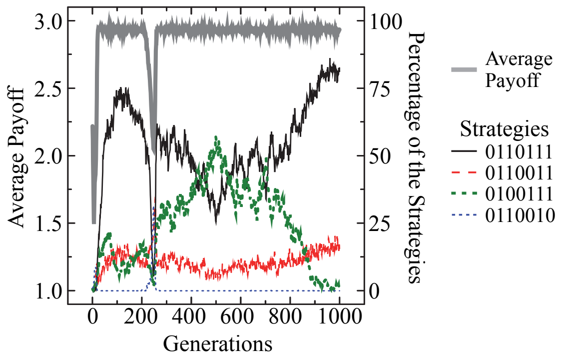


(a) One-point crossover.

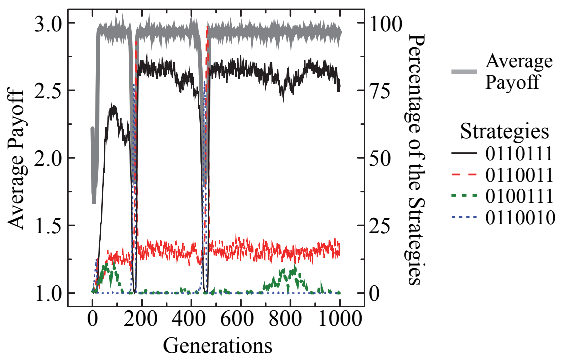


(b) Uniform crossover.

Fig. 10. Average payoff 3.0 with error probability 0.



(a) One-point crossover.



(b) Uniform crossover.

Fig. 12. Average payoff 3.0 with error probability 0.02.

D. The Error in a Memory about Actions

In [8], we used the error in action selection. The error in a player's memory was not examined. In our real environment, we do not recall all the past events correctly. That is, our memory has some uncertainly. In this subsection, we use the error in a memory about the player's and/or the opponent's previous actions in the IPD game. That is, a player's memory has a different action from the actually selected one in the IPD game with the error probability.

Fig. 13 shows the average payoff at the 1000th generation over 50 runs by using the error in a player's memory. When we used the one-point crossover, the strange behavior of the 7-bit binary string strategy was observed. The difference in the average payoff between the error probability 0.0 and 0.01 was statistically significant (Mann-Whitney U test, p = 0.038). However, when we used the two-point crossover and the uniform crossover, it was not observed. We cannot observe any large differences between the error in action selection and the error in a player's memory with respect to the strange behavior of 7-bit binary string strategies.

However, we can observe a clear difference between two settings in Fig. 13. When the error probability was higher than 0.1, clearly lower average payoff was obtained from the error in action selection than the error in a player's memory. Especially when the error probability was higher than 0.15, the perfect mutual defection with the average payoff 1.0 was obtained from the error in a player's memory in all the three settings of crossover. However, the average payoff about 1.5 was obtained in the case of the error in action selection. This is because a high error probability in action selection means an occasional choice of cooperation even if the strategy always chooses defection.

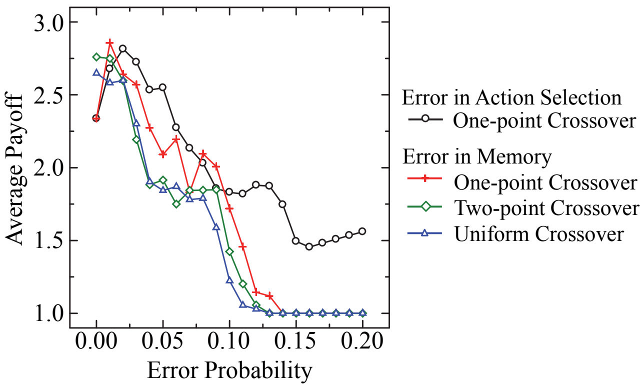


Fig. 13. Average payoff by using the error in a player's memory.

VI. Conclusions

In this paper, we examined the effects of the parameter settings on the strange behavior of 7-bit binary string strategies. The strange behavior means a clear increase in the average payoff by the increase of the error probability from zero to a mall positive value. Computational experiments demonstrated that it strongly depended on the parameter settings.

The number of players was one of the important factors for the strange behavior. When we used 100 players, the strange behavior was not observed. This may be because the decrease in the number of players decreases the diversity of strategies. The memory length and the memory usage in strategies were also important factors. The strange behavior was observed only when we used strategies with the memory length two for the opponent's previous two actions whereas we examined a wide variety of combinations of the memory length and the memory usage.

The choice of a crossover operator showed a large effect on the evolution of strategies. The strange behavior was observed only when we used the one-point crossover operator. Further experiments are needed to explain why the one-point crossover operator discouraged the evolution of cooperation in the noise-free setting.

Finally, we examined the error in a player's memory. The strange behavior was observed in this setting as in the case of the error in action selection. When the error probability was higher than 0.15, the perfect mutual defection with the average payoff 1.0 was observed in the case of the error in a player's memory whereas the average payoff was about 1.5 in the case of the error in action selection.

In summary, the strange behavior was observed only in the following settings: 7-bit binary strings were used to represent player's strategies, the number of player was relatively large (i.e., more than 200-300), the one-point crossover operator was used to generate new strategies, and a small error probability (e.g., 0.02) in action selection and a player's memory was compared with the setting of the error probability zero.

References

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

[2] D. B. Fogel, "Evolving behaviors in the iterated prisoner's dilemma," *Evolutionary Computation*, vol. 1, no. 1, pp. 77-97, 1993.

[3] G. Kendall, X. Yao, and S. Y. Chong (eds.), *The Iterated Prisoners' Dilemma: 20 Years*, World Scientific, Singapore, 2007.

[4] H. Ishibuchi, K. Hoshino, and Y. Nojima, "Evolution of strategies in a spatial IPD game with a number of different representation schemes," *Proc. of 2012 IEEE Congress on Evolutionary Computation*, pp. 808-815, Brisbane, June 10-15, 2012.

[5] H. Ishibuchi, K. Takahashi, K. Hoshino, J. Maeda, and Y. Nojima, "Effects of configuration of agents with different strategy representations on the evolution of cooperative behavior in a spatial IPD game," *Proc. of 2011 IEEE Conference on Computational Intelligence and Games*, pp. 313-320, Seoul, August 31 - September 3, 2011.

[6] H. Ishibuchi, H. Ohyanagi, and Y. Nojima, "Evolution of strategies with different representation schemes in a spatial iterated prisoner's dilemma game," *IEEE Trans. on Computational Intelligence and AI in Games*, vol. 3, no. 1, pp. 67-82, March 2011.

[7] D. Ashlock and E. Y. Kim, "The impact of varying resources available to iterated prisoner's dilemma agents," *Proc. of 2013 IEEE Symposium on Foundations of Computational Intelligence*, pp. 60-67, Singapore, 2013.

[8] T. Sudo, K. Goto, Y. Nojima, and H. Ishibuchi, "Strange evolution behavior of 7-bit binary string strategies in iterated prisoner's dilemma game," *Proc. of 2015 IEEE Congress on Evolutionary Computation*, pp. 3346-3353, Japan, 2015.

[9] S. Y. Chong and X. Yao, "Behavioral diversity, choices and noise in the iterated prisoner's dilemma," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 6, pp. 540-551, 2005.