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, Australia, June, 10-15, 2012.

Evolution of Strategies in a Spatial IPD Game with a Number of Different Representation Schemes

Hisao Ishibuchi, Koichiro Hoshino, and Yusuke Nojima
Department of Computer Science and Intelligent Systems
Osaka Prefecture University
Sakai, Osaka, Japan
{hisaoi@, kouichirou.hoshino@ci., nojima@}cs.osakafu-u.ac.jp

***Abstract*—We examine the evolution of strategies for a spatial IPD (Iterated Prisoner's Dilemma) game, which are encoded using different representation schemes. Each agent at a cell in a two-dimensional grid-world has its own representation scheme for encoding its strategy. In general, strategies with different representation schemes cannot be recombined. Thus a population of agents can be viewed as a mixture of different species (i.e., an ecology with different species). When the size of a neighborhood structure is small and/or the number of representation schemes is large, it is likely that some agents have no neighbors with the same representation scheme. We discuss the handling of those agents because they cannot generate their new strategies through recombination. In computational experiments, we use four types of strings (i.e., four representation schemes). Agents in our spatial IPD game are randomly divided into four groups of the same size. One string type is assigned to each group. Recombination is performed between strings of neighboring agents with the same string type. With respect to the IPD game, we compare two settings with each other. In one setting, the IPD game is played between any pair of neighboring agents regardless of their string types. In the other setting, it is played only between neighboring agents with the same string type. Using these two settings, we examine the effect of the IPD game between agents with different representation schemes on strategy evolution. We also examine the effect of the number of different representation schemes in a population (i.e., the number of species) on strategy evolution.**

Keywords - Iterated prisoner's dilemma games, spatial IPD games, evolutionary games, evolution of game strategies, encoding, representation schemes

I. INTRODUCTION

Recently Ashlock et al. [1]-[3] clearly demonstrated that the choice of a representation scheme for encoding IPD (Iterated Prisoner's Dilemma) game strategies has large effects on the evolution of cooperative behavior. Totally different results were obtained from different representation schemes in their studies. Spatial structures of agents also have large effects on the evolution of cooperative behavior. In our former study on a spatial IPD game [4], we examined the effect of using two neighborhood structures: One is for local parent selection for recombination in strategy evolution and the other is for local opponent selection in the IPD game. It was demonstrated in [4] that the use of a small neighborhood structure for opponent selection had a large positive effect on the evolution of cooperative behavior.

In most studies on the evolution of IPD game strategies, the same representation scheme was assigned to all agents. Even when different representation schemes were examined through computational experiments, a single representation scheme was assigned to all agents in each run (while a different scheme was used in a different run). In contrast to those studies, two representation schemes were simultaneously used in each run in our former studies [5], [6]. Agents were divided into two sub-populations with different representation schemes. As a result, a population of agents was a mixture of two species. The main observation in [5], [6] was that similar results were obtained from different representation schemes when they were used in a single run. When those representation schemes were used separately in different runs, totally different results were obtained as in [1]-[3].

In our former studies [5], [6], we examined a mixture of two representation schemes. In this paper, we examine a case with four representation schemes. Agents in a two-dimensional grid-world are divided into four sub-populations. A different representation scheme is assigned to a different sub-population. When we use many representation schemes, it is likely that some agents have no neighbors with the same representation scheme. We cannot generate new strategies for those agents by recombination since no neighbors have the same representation scheme. Only mutation can be used to generate new strategies. It is also impossible for those agents to find their opponents with the same representation scheme among their neighbors when they play the IPD game.

This difficulty did not become clear in [5], [6] because we used only two representation schemes. We also allowed each agent to play the IPD game against itself. This means that each agent had at least one opponent (i.e., that agent itself) with the same representation scheme. Thus computational experiments could be performed in [5], [6] under the following setting: The IPD game was played only between neighbors with the same representation scheme. However, the IPD game of each agent against itself is somewhat strange since we do not play any games against ourselves in our everyday life. Moreover, the IPD game of each agent against itself has a strong bias towards cooperation when a small neighborhood structure is used for local opponent selection [7]. In this paper, we do not allow any agent to play the IPD game against itself in a population with four different representation schemes where some agents have no neighboring opponents with the same representation scheme.

In order to avoid the above-mentioned difficulty (i.e., some agents have no qualified mates for recombination or qualified opponents for the IPD game), we use a following simple trick in our computational experiments: When an agent has no neighbors with the same representation scheme, we use a larger neighborhood structure for that agent. In this manner, the following situation is always maintained in our computational experiments: Each agent has at least one neighboring agent with the same representation scheme.

This paper is organized as follows. In Section II, we briefly explain our spatial IPD game with two neighborhood structures (i.e., one for local opponent selection in the IPD game and the other for local parent selection in strategy evolution). We also explain four representation schemes of IPD game strategies in Section II. In Section III, we explain the setting of our computational experiments where IPD game strategies are evolved. We examine various specifications with respect to the assignment of representation schemes to agents. In Section IV, we report experimental results. We discuss the effect of using different representation schemes in a single population on the evolution of IPD game strategies. Finally we conclude this paper in Section V.

II. Spatial IPD Game and Representation Schemes

A. IPD Game

We use a standard payoff matrix of the prisoner's dilemma game in Table I. For example, if both the agent and the opponent cooperate, they receive three points (i.e., "Agent: 3" and "Opponent: 3" in Table I). The agent receives the highest payoff of five by defecting when the opponent cooperates (i.e., "Agent: 5" and "Opponent: 0" in Table I). The prisoner's dilemma game is iterated for 100 rounds in our computational experiments in this paper.

TABLE I. Standard Payoff Matrix of Prisoner's Dilemma Game.

Agent's Action	Opponent's Action	
	C: Cooperate	D: Defect
C: Cooperate	Agent: 3 Opponent: 3	Agent: 0 Opponent: 5
D: Defect	Agent: 5 Opponent: 0	Agent: 1 Opponent: 1

B. IPD Game Strategies

An agent's strategy determines its next action based on a finite memory about previous actions. In this paper, we use the following four types of strings (i.e., four representation schemes) to represent IPD game strategies:

Binary strings of length 3,
Real number strings of length 3,
Binary strings of length 7,
Real number strings of length 7.

In binary strings, "0" and "1" represent "D: Defect" and "C: Cooperate", respectively. Binary strings of length 3 determine the next action based on the opponent's previous action. In Table II, we show a binary string of length 3 "101" called TFT (tit-for-tat). The TFT in Table II encoded as "101" cooperates at the first round and then cooperates at each round only when the opponent cooperated in the previous round.

TABLE II. TFT by the Binary String of Length 3: "101".

Agent's first action: Cooperate		1
Opponent's previous action	Suggested action	
D: Defect	D: Defect	0
C: Cooperate	C: Cooperate	1

Each bit value of binary strings of length 3 can be viewed as the probability of cooperation. For example, the first bit value "1" of "101" in Table II can be viewed as the probability of cooperation at the first round. Using this interpretation, we can generalize binary strings to real number strings. For example, "0.2 0.1 0.9" cooperates with the probability 0.2 in the first round, cooperates with the probability 0.1 in response to the opponent's defection, and cooperates with the probability 0.9 in response to the opponent's cooperation.

Binary strings of length 7 determine the next action based on the opponent's previous two actions. In Table III, the TFT strategy "101" in Table II is shown as a binary string of length 7 "1010101". The first bit "1" specifies the action in the first round. The next two bits "01" determine the action in the second round depending on the opponent's action in the first round. The remaining four bits "0101" determine the action in each of the other rounds depending on the opponent's previous two actions. Another example is shown in Table IV where the binary string "1110111" defects only when the opponent defects in the previous two rounds. Binary strings of length 7 can be easily generalized to real number strings of length 7.

TABLE III. TFT by the Binary String of Length 7: "1010101".

Agent's first action: Cooperate		1
Opponent's previous action	Suggested action	
D: Defect	D: Defect	0
C: Cooperate	C: Cooperate	1
Opponent's previous two actions	Suggested action	
D: Defect and D: Defect	D: Defect	0
D: Defect and C: Cooperate	C: Cooperate	1
C: Cooperate and D: Defect	D: Defect	0
C: Cooperate and C: Cooperate	C: Cooperate	1

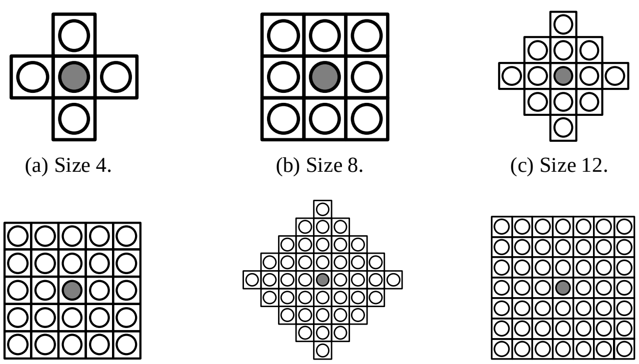
TABLE IV. Another Binary String of Length 7 "1110111".

Agent's first action: Cooperate		1
Opponent's previous action	Suggested action	
D: Defect	C: Cooperate	1
C: Cooperate	C: Cooperate	1
Opponent's previous two actions	Suggested action	
D: Defect and D: Defect	D: Defect	0
D: Defect and C: Cooperate	C: Cooperate	1
C: Cooperate and D: Defect	C: Cooperate	1
C: Cooperate and C: Cooperate	C: Cooperate	1

C. Spatial IPD Game with Two Neighborhood Structures

We use an 11×11 grid-world with the torus structure. The number of agents is 121, which is the same as the number of cells in the 11×11 grid-world. In our spatial IPD game, opponents of each agent are selected from its neighbors. Let us denote the set of neighbors of Agent *i* by $N_{\mathrm{IPD}}(i)$ for local opponent selection. That is, $N_{\mathrm{IPD}}(i)$ is a neighborhood structure for local opponent selection. As we have already explained, any agent is not allowed to play the IPD game against itself. This means that $N_{\mathrm{IPD}}(i)$ does not include Agent *i* itself.

In Fig. 1, we show six neighborhood structures used as $N_{\mathrm{IPD}}(i)$ in this paper. The six neighborhood structures in Fig. 1 include 4, 8, 12, 24, 40 and 48 neighbors, respectively. We also examine an extreme specification of $N_{\mathrm{IPD}}(i)$ for Agent *i* where all the other 120 agents in the 11×11 grid-world are included in $N_{\mathrm{IPD}}(i)$. That is, we examine the seven specifications of $N_{\mathrm{IPD}}(i)$ including 4, 8, 12, 24, 40, 48 and 120 neighbors.



(a) Size 4. (b) Size 8. (c) Size 12.

(d) Size 24. (e) Size 40. (f) Size 48.

Figure 1. Six neighborhood structures used as $N_{\mathrm{IPD}}(i)$.

In our computational experiments, all neighbors in $N_{\mathrm{IPD}}(i)$ are selected as opponents of Agent *i* when $N_{\mathrm{IPD}}(i)$ includes five or less neighbors as in Fig. 1 (a). When $N_{\mathrm{IPD}}(i)$ includes more than five neighbors as in Fig. 1 (b)-(f), five neighbors are randomly selected from $N_{\mathrm{IPD}}(i)$ as opponents of Agent *i* (i.e., random sampling of five neighbors from $N_{\mathrm{IPD}}(i)$ without replacement). The fitness value of the strategy of Agent *i* is calculated as the average payoff from the IPD game against all the selected opponents.

A new strategy for each agent is generated from strategies of its neighbors through selection, crossover and mutation. Parent strategies are selected for each agent from its neighbors. Let $N_{\mathrm{GA}}(i)$ be a set of neighbors of Agent *i* for local parent selection. That is, $N_{\mathrm{GA}}(i)$ is a neighborhood structure for local parent selection. A new strategy of Agent *i* can be generated from its current strategy. Thus $N_{\mathrm{GA}}(i)$ includes Agent *i* itself. In our computational experiments, we examine six specifications of $N_{\mathrm{GA}}(i)$ corresponding to the six neighborhood structures for $N_{\mathrm{IPD}}(i)$ in Fig. 1 (note that $N_{\mathrm{GA}}(i)$ includes Agent *i* while $N_{\mathrm{IPD}}(i)$ does not include Agent *i*). We also examine an extreme specification of $N_{\mathrm{GA}}(i)$, which includes all the 121 agents in the 11×11 grid-world.

III. COMPUTATIONAL EXPERIMENTS

A. Evolution of IPD Game Strategies

In each trial of strategy evolution (i.e., in each run in our computational experiments), one of the four types of strings is assigned to each agent. The assigned type of strings is never changed during strategy evolution in each trial. An initial strategy is randomly generated for each agent. For binary strings, each bit is randomly specified as 0 or 1 with the same probability. Real numbers are randomly specified using the uniform distribution over the unit interval [0, 1]. The fitness value of each agent is calculated as the average payoff obtained from the IPD game with 100 rounds against opponents in its local opponent selection neighborhood.

After the fitness calculation for all agents is completed, two parents are selected for each agent from its local parent selection neighborhood using binary tournament selection with replacement. A new strategy is generated through crossover and mutation from the selected two parents for each agent. For binary strings, we use one-point crossover and bit-flip mutation. For real number strings, we use blend crossover (BLX-α [8]) with $\alpha = 0.25$ and uniform mutation. If a real number becomes more than 1 (or less than 0) by the crossover operator, it is repaired to be 1 (or 0) before the mutation. The same crossover probability 1.0 and the same mutation probability $1/(5 \times 121)$ are used for binary and real number strategies.

The current strategy of each agent is replaced with a newly generated one for that agent. The fitness evaluation and the generation update are iterated for 1000 generations.

B. Expansion of Neighborhood Structures

When we use many representation schemes, it is almost always the case that some agents have no neighbors with the same representation scheme in their local parent selection neighborhood. This means that those agents have no mates for recombination. That is, recombination is never used to generate new strategies for those agents. In this case, we expand the local parent selection neighborhood of those agents.

Let us assume that the local parent selection neighborhood $N_{\mathrm{GA}}(i)$ of Agent *i* includes no neighbors with the same representation scheme as Agent *i*. In this case, we expand $N_{\mathrm{GA}}(i)$ in the following order: Fig. 1 (a) => Fig. 1 (b) => ... => Fig. 1 (f). For example, if $N_{\mathrm{GA}}(i)$ with eight neighbors in Fig. 1 (b) includes no neighbors with the same representation scheme as Agent *i*, $N_{\mathrm{GA}}(i)$ is expanded from Fig. 1 (b) to Fig. 1 (c). If

$N_{GA}(i)$ in Fig. 1 (c) still has no neighbors with the same representation scheme as Agent i, it is further expanded from Fig. 1 (c) to Fig. 1 (d). In this manner, $N_{GA}(i)$ of each agent is expanded until $N_{GA}(i)$ includes at least one neighbor with the same representation scheme as Agent i. As a result, each agent may have its own local parent selection neighborhood structure of a different size in our computational experiments.

With respect to the IPD game, we examine the following two settings. In one setting, the IPD game is played between two neighbors regardless of their representation schemes. In the other setting, the IPD game is played between neighbors only when they have the same representation scheme. That is, Agent i can play the IPD game only against its neighbors with strategies of the same string type. In this setting, the fitness value of Agent i cannot be evaluated through the IPD game if it has no neighbors with the same representation scheme in $N_{IPD}(i)$. In this case, we expand the local opponent selection neighborhood $N_{IPD}(i)$ in the same manner as $N_{GA}(i)$. That is, if $N_{IPD}(i)$ has no neighbors with the same representation scheme as Agent i under the latter setting (i.e., the IPD game is played only between neighbors with the same representation scheme), $N_{IPD}(i)$ is expanded in the following order: Fig. 1 (a) => Fig. 1 (b) => ... => Fig. 1 (f). As a result, each agent may have its own local opponent selection neighborhood structure of a different size in our computational experiments. It should be noted that local opponent selection neighborhood is never expanded under the former setting (i.e., when the IPD game can be played between any neighbors regardless of their representation schemes).

C. Settings of Computational Experiments

As we have already mentioned, we use the following four representation schemes:

Four Representation Schemes

Binary strings of length 3,
Real number strings of length 3,
Binary strings of length 7,
Real number strings of length 7.

We examined the following three cases with respect to the assignment of representation schemes to agents:

Homogeneous Case with a Single Representation Scheme:
A single representation scheme is assigned to all of the 121 agents in our spatial IPD game with the 11×11 grid-world. Since we have the four representation schemes, four settings are examined. In each setting, one of the four representation schemes is assigned to all agents.

Heterogeneous Case with Two Representation Schemes:
Two types of strings of the same length are used by agents as their representation schemes. We examine two settings with respect to string length: length 3 and length 7. In one setting, binary or real number strings of length 3 are assigned to each agent as its representation scheme. More specifically, we first randomly choose 60 agents from the 11×11 grid-world. Then we assign binary strings of length 3 to the selected 60 agents (and real number strings of length 3 to the remaining 61 agents) as their representation schemes. In each trial of strategy evolution, the random selection of 60 agents is updated. In the other setting, binary or real number strings of length 7 are assigned to each agent in the same manner as in the case of length 3 strings. For each setting with respect to string length, we examine the two settings with respect to the IPD game (as we have just explained in the previous subsection: with/without the IPD game between different representation schemes).

Heterogeneous Case with Four Representation Schemes:
One of the four types of strings is assigned to each agent as its representation scheme. More specifically, first 121 agents in the 11×11 grid-world are randomly divided into four sub-populations with 30, 30, 30 and 31 agents. One of the four types of strings is assigned to all agents in each sub-population. In each trial of strategy evolution, the random partition into four sub-populations is updated. The two settings with respect to the IPD game (i.e., with/without the IPD game between different representation schemes) are examined for such a heterogeneous population with the four types of strings.

In our computational experiments, average results are calculated over 500 trials of strategy evolution for each setting (i.e., over 500 runs of a cellular genetic algorithm for strategy evolution). In each run, we use the following specifications:

Parameters for the IPD Game:

Grid-world: 11×11 grid-world with the torus structure,
Number of agents: 121,
Neighborhood structure $N_{IPD}(i)$ for opponent selection:
Seven structures (Fig. 1 and the grid-world itself),
Number of opponents: Maximum five neighbors,
Number of rounds in the IPD game: 100 rounds.

Parameters for Strategy Evolution:

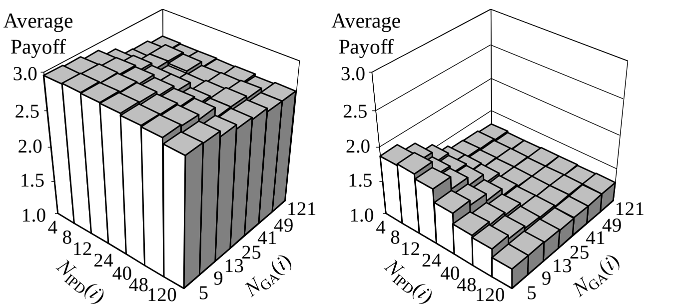
Population size: 121 in the 11×11 grid-world,
Representation schemes:
Binary and real number strings of length 3 and 7,
Initial population: Randomly generated strings,
Neighborhood structure $N_{GA}(i)$ for parent selection:
Seven structures (Fig. 1 and the grid-world itself),
Parent selection:
Binary tournament selection with replacement,
Crossover: One-point crossover for binary strings, and
BLX-α with $\alpha = 0.25$ for real number strings,
Mutation: Bit-flip mutation for binary strings, and
uniform mutation for real number strings,
Crossover probability: 1.0,
Mutation probability: $1/(5 \times 121)$,
Constraint handling for real number strings:
Repair to 1 (if larger than 1) or 0 (if smaller than 0),
Termination condition: 1000 generations.

As we have already explained, $N_{IPD}(i)$ of Agent i for local opponent selection does not include Agent i itself while $N_{GA}(i)$ for local parent selection includes Agent i.

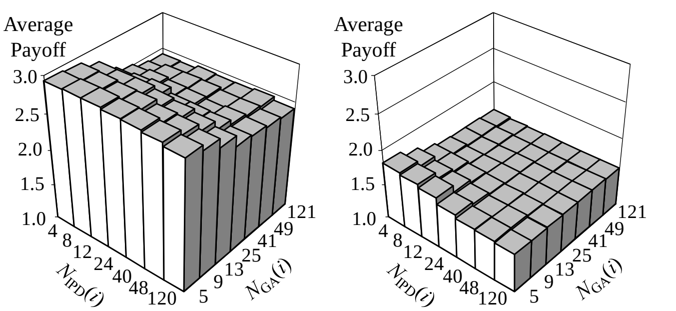
IV. Experimental Results

A. Results using a Single Representation Scheme

First we report experimental results using a homogeneous population with a single representation scheme. One of the four representation schemes is assigned to all agents. The average payoff over all agents is calculated through 1000 generations of 500 runs. This calculation is performed for each of the 7×7 combinations of the two neighborhood structures: $N_{\mathrm{IPD}}(i)$ for local opponent selection and $N_{\mathrm{GA}}(i)$ for local parent selection. Computational experiments are performed for each of the four representation schemes. Experimental results are summarized in Fig. 2. The vertical axis is the average payoff while the base plane shows the 7×7 combinations of the two neighborhood structures. In Fig. 3, we show the average payoff at the 1000th generation (while Fig. 2 shows the average payoff over 1000 generations). We can see that the choice of a representation scheme has a large effect on the average payoff. That is, the evolution of cooperative behavior strongly depends on the choice of a representation scheme for encoding IPD game strategies. The highest average results are obtained from binary strings of length 3 among the four types of strings. We observe no large differences between the average results over 1000 generations and those at the 1000th generation. In the following, we report only the average results over 1000 generations.

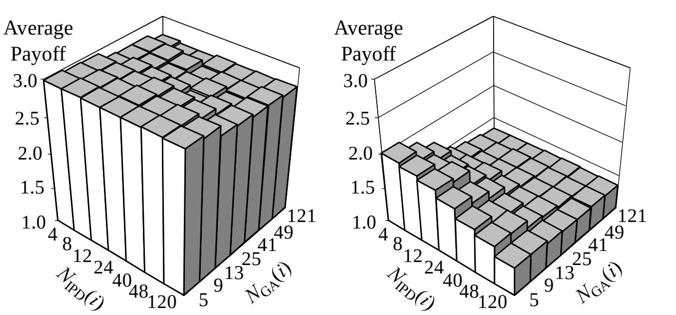


(a) Binary strings of length 3. (b) Real number strings of length 3.

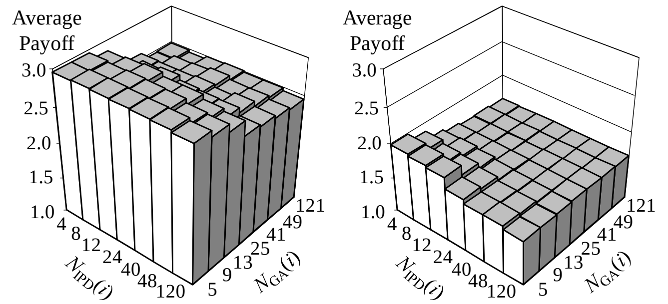


(c) Binary strings of length 7. (d) Real number strings of length 7.

Figure 2. Average payoff over 1000 generations from a homogeneous population with a single representation scheme.



(a) Binary strings of length 3. (b) Real number strings of length 3.

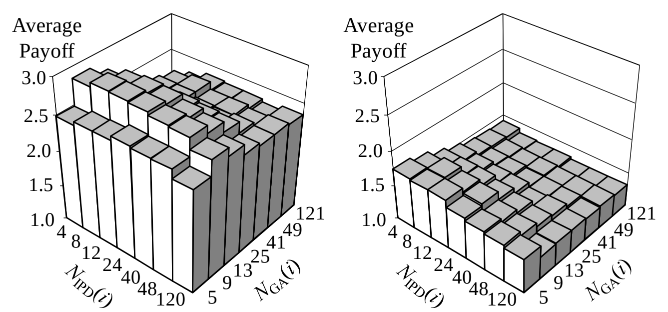


(c) Binary strings of length 7. (d) Real number strings of length 7.

Figure 3. Average payoff at the 1000th generation from a homogeneous population with a single representation scheme.

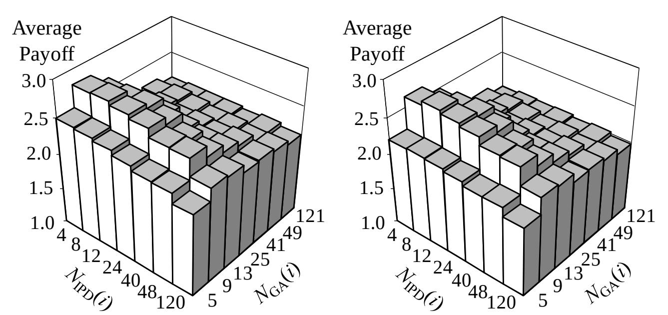
B. Results using Two Representation Schemes

Next we report experimental results by a mixture of two types of strings of the same length. Fig. 4 and Fig. 5 show the average payoff over 1000 generations from a mixture of binary and real number strings of length 3. The IPD game is not played between agents with different representation schemes in Fig. 4 while it is played regardless of representation schemes in Fig. 5. That is, there exists no interaction between binary and real number strings in Fig. 4 while these two types of strings interact with each other through the IPD game in Fig. 5.



(a) Binary strings of length 3. (b) Real number strings of length 3.

Figure 4. Results from a mixture of binary and real number strings of length 3 (without the IPD game between binary and real number strings of length 3).



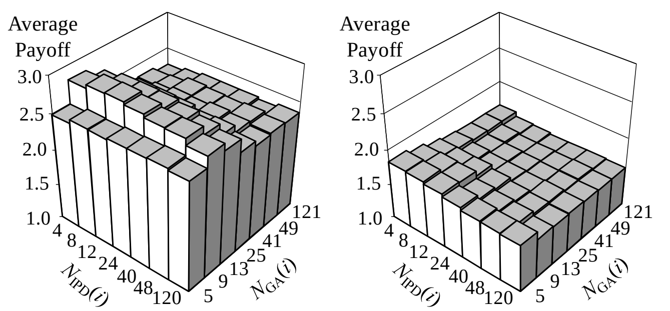
(a) Binary strings of length 3. (b) Real number strings of length 3.

Figure 5. Results from a mixture of binary and real number strings of length 3 (with the IPD game between binary and real number strings of length 3).

In Fig. 4, strategies with one type of stings are evolved independently from those with the other type of strings. Thus the experimental results are totally different between Fig. 4 (a) and Fig. 4 (b). The average payoff in Fig. 4 (a) is decreased from Fig. 2 (a). This is because the population size and the number of qualified mates in $N_{GA}(i)$ in Fig. 4 (a) are actually a half of those in Fig. 2 (a). The negative effect of the decrease in the number of qualified mates in $N_{GA}(i)$ is prominent when the size of $N_{GA}(i)$ is five (i.e., the left-most seven bars in Fig. 4 (a)).

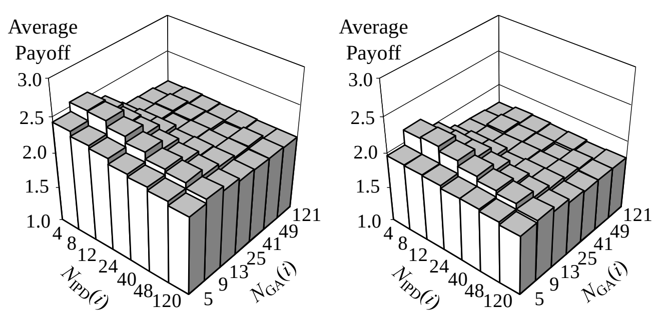
In Fig. 5, the two types of strings are interacted with each other through the IPD game. This interaction leads to similar results in Fig. 5 (a) and Fig. 5 (b). From Fig. 4 and Fig. 5, we can see that the interaction through the IPD game leads to a large increase in the average payoff of real number strings from Fig. 4 (b) to Fig. 5 (b) at the cost of small decrease in the average payoff of binary strings from Fig. 4 (a) to Fig. 5 (a).

As in Fig. 4 and Fig. 5, we show experimental results by a mixture of binary and real number strings of length 7 in Fig. 6 and Fig. 7. Similar results are obtained in Fig. 7 (a) and Fig. 7 (b) from the two types of length 7 strings when they are interacted with each other through the IPD game. In the case of no interaction between them, Fig. 6 (a) and Fig. 6 (b) are totally different from each other. We can also see that the experimental results by length 3 strings in Fig. 4 and Fig. 5 are similar to those by length 7 strings in Fig. 6 and Fig. 7. A slightly higher average payoff is obtained from length 3 strings in Fig. 5 than length 7 strings in Fig. 7.



(a) Binary strings of length 7. (b) Real number strings of length 7.

Figure 6. Results from a mixture of binary and real number strings of length 7 (without the IPD game between binary and real number strings of length 7).



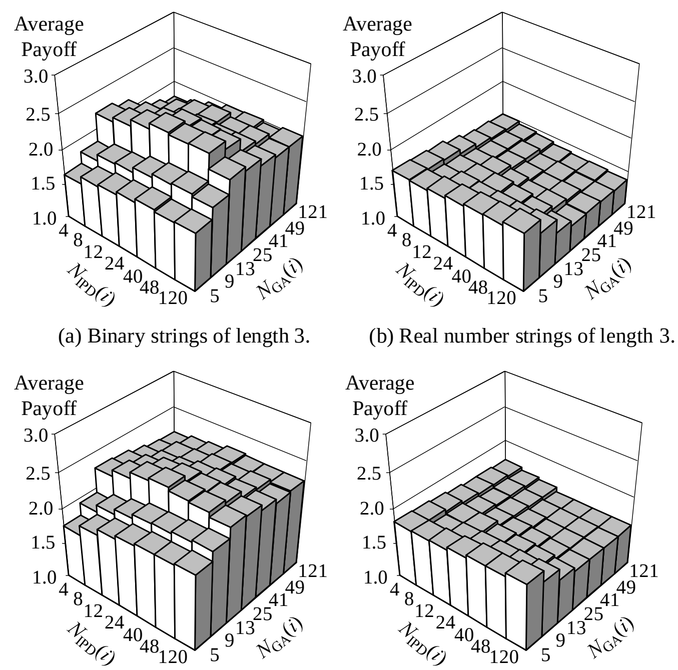
(a) Binary strings of length 7. (b) Real number strings of length 7.

Figure 7. Results from a mixture of binary and real number strings of length 7 (with the IPD game between binary and real number strings of length 7).

C. Results using Four Representation Schemes

Finally we report experimental results by a mixture of the four types of strings. In Fig. 8, we show experimental results with no interaction among the four types of strings. Strategies with each type of strings are evolved independently from those with the other three types. In Fig. 8, a high average payoff close to 3.0 is not obtained. This may be because the number of agents with each string type is small (i.e., 30 or 31). When binary strings of length 3 were assigned to all of the 121 agents, we almost always obtained a high average payoff close to 3.0 in Fig. 2 (a) in Subsection IV.A. Even when this type of strings was randomly assigned to a half of the 121 agents, a high average payoff close to 3.0 was obtained in Fig. 4 (a) in Subsection IV.B. However, the average payoff in Fig. 8 (a) by binary strings of length 3 is not so high (i.e., it is lower than Fig. 1 (a) and Fig. 4 (a)). Moreover, negative effects of using small parent selection neighborhood $N_{GA}(i)$ are prominent when $N_{GA}(i)$ includes 5 or 9 neighbors in Fig. 8 (a) and Fig. 8 (c). This is because only a quarter of neighbors in $N_{GA}(i)$ are qualified mates of Agent i in recombination.

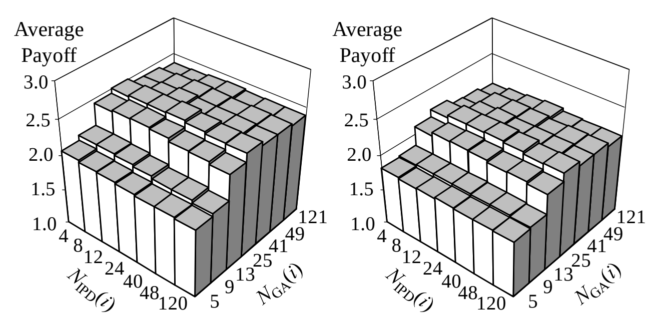
Fig. 9 shows the average payoff of agents with each type of strings when the IPD game is performed between neighbors regardless of their string types. It is interesting to observe that similar results are obtained from the four types of strings in Fig. 9 where agents with different types of strings are interacted with each other through the IPD game.



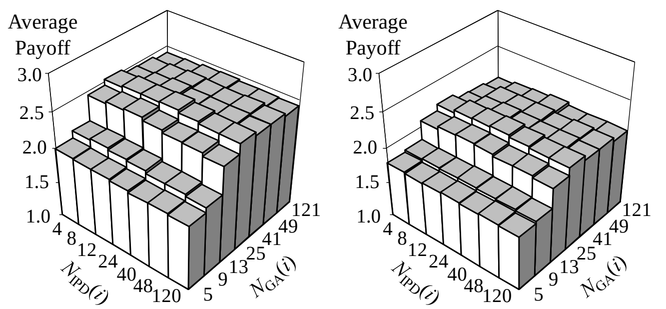
(a) Binary strings of length 3. (b) Real number strings of length 3.

(c) Binary strings of length 7. (d) Real number strings of length 7.

Figure 8. Average payoff from a mixture of the four types of strings (without the IPD game between agents with different representation schemes).



(a) Binary strings of length 3. (b) Real number strings of length 3.



(c) Binary strings of length 7. (d) Real number strings of length 7.

Figure 9. Average payoff from a mixture of the four types of strings (with the IPD game between agents with different representation schemes).

In Fig. 9, it is also interesting to observe that the size of opponent selection neighborhood $N_{IPD}(i)$ has almost no effect on the average payoff in Fig. 9. From the comparison between Fig. 8 and Fig. 9, we can see that a higher average payoff is obtained for almost all combinations of $N_{IPD}(i)$ and $N_{GA}(i)$ from Fig. 9 with the interaction between different types of strings through the IPD game than Fig. 8 with no interaction. That is, the interaction among the four types of strings through the IPD game increases the average payoff of agents from Fig. 8 to Fig. 9.

D. *Discussions on Experimental Results*

In order to compare all experimental results reported in this section, we calculate the overall average payoff over all the 121 agents regardless of their types of strings in all IPD games for all combinations of $N_{IPD}(i)$ and $N_{GA}(i)$. For example, the overall average payoff is calculated over all experimental results in Fig. 9 (i.e., over all experimental results related to all bars in the four plots in Fig. 9 (a)-(d)). The calculated overall average payoff is summarized in Table V. The third column shows the number of agents with each type of strings in a population. For example, "121(B3)" means a population of 121 agents with binary strings of length 3 while "60(B3), 61(R3)" means a mixture of 60 agents with binary strings of length 3 and 61 agents with real number strings of length 3. The last column of Table V shows the interaction between different species (i.e., between agents with different string types).

TABLE V. OVERALL AVERAGE PAYOFF OVER ALL AGENTS.

Average Payoff	Corresponding Figure	Number of Agents with Each Type of Strings	Interaction between Species
2.76	Fig. 2 (a)	121(B3)	-
1.40	Fig. 2 (b)	121(R3)	-
2.64	Fig. 2 (c)	121(B7)	-
1.59	Fig. 2 (d)	121(R7)	-
1.98	Fig. 4 (a)-(b)	60(B3), 61(R3)	No Interaction
2.28	Fig. 5 (a)-(b)	60(B3), 61(R3)	IPD Game
2.06	Fig. 6 (a)-(b)	60(B7), 61(R7)	No Interaction
2.01	Fig. 7 (a)-(b)	60(B7), 61(R7)	IPD Game
1.87	Fig. 8 (a)-(d)	30(B3), 30(R3), 30(B7), 31(R7)	No Interaction
2.18	Fig. 9 (a)-(d)	30(B3), 30(R3), 30(B7), 31(R7)	IPD Game

The first four values of the average payoff in Table V show that the evolution of IPD game strategies strongly depends on the choice of a representation scheme (when the selected representation scheme is used by all agents). When we use a mixture of two string types of length 3, the interaction between different representation schemes through the IPD game increases the overall average payoff by 0.30 from 1.98 to 2.28 (see the next two values of the average payoff corresponding to Fig. 4 (a)-(b) and Fig. 5 (a)-(b)). When we use the four representation schemes (i.e., in the last two rows of Table V), the interaction through the IPD game increases the overall average payoff by 0.31 from 1.87 to 2.18.

In order to further examine the evolution of IPD game strategies, we monitor the percentage of each strategy at each generation for binary strings of length 3. It should be noted that this representation scheme can represent only eight strategies: 000 (ALLD: always defect), 001, 010, 011, 100, 101 (TFT), 110, 111 (ALLC: always cooperate). We calculate the average percentage of each strategy at each generation in the five settings with this representation scheme in Table V (i.e., the five settings with "B3" in Table V). We use experimental results from the combination of $|N_{IPD}(i)| = 12$ and $|N_{GA}(i)| = 13$ because this seems to be one of the best combinations with a high average payoff in all the five settings (i.e., in Fig. 2 (a), Fig. 4 (a), Fig. 5 (a), Fig. 8 (a) and Fig. 9 (a)).

Experimental results are summarized in Figs. 10-12. In Fig. 10, all agents have binary strings of length 3. We can see from Fig. 10 that their strategies are evolved to "101" (TFT) and "111" (ALLC). It should be noted that exactly the same payoff is obtained from "111" and "101" when there exist no other strategies in the population. Thus these two strategies have the same fitness value in their population. As a result, both strategies can exist in the same population (if no other strategies are generated by mutation).

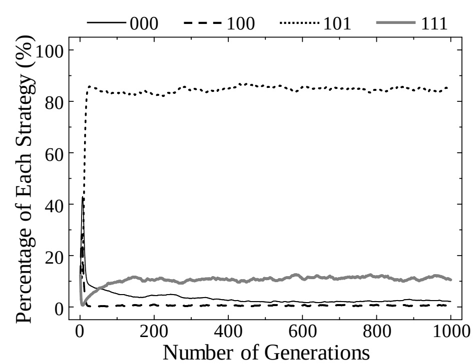
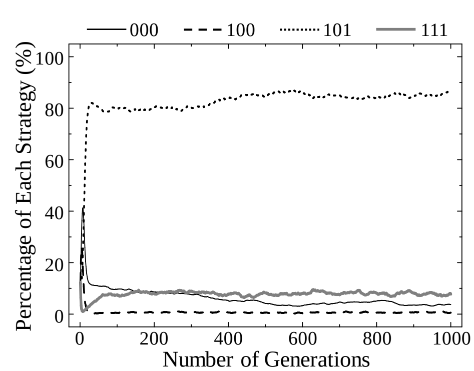
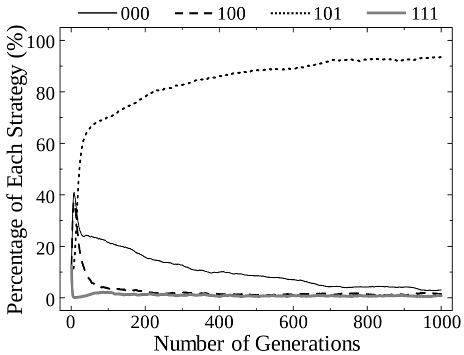


Figure 10. Percentage of each strategy among 121 binary strings of length 3.

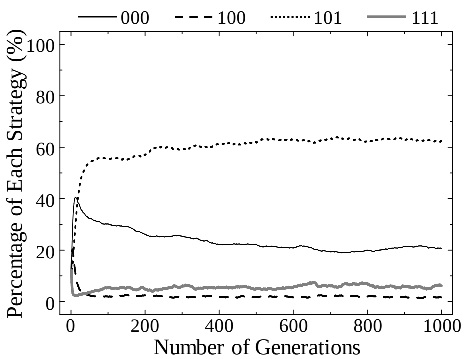


(a) With no IPD game against 61 real number strings of length 3.

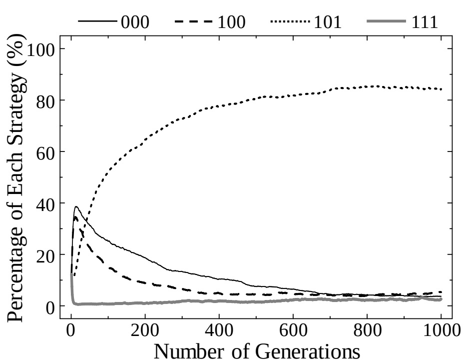


(b) With the IPD game against 61 real number strings of length 3.

Figure 11. Percentage of each strategy among 60 binary strings of length 3.



(a) With no IPD game against 91 agents with the other representation schemes.



(b) With the IPD game against 91 agents with the other representation schemes.

Figure 12. Percentage of each strategy among 30 binary strings of length 3.

In Fig. 11 (a) and Fig. 12 (a), agents with binary strings of length 3 evolve their strategies independently from other agents with different representation schemes. Thus agents with this representation scheme in Fig. 10, Fig. 11 (a) and Fig. 12 (a) can be viewed as having similar environments. As a result, their strategies are evolved towards "111" and "101" in a somewhat similar manner (the evolution in Fig. 12 (a) is not easy).

In Fig. 11 (b) and Fig. 12 (b), agents with binary strings of length 3 play the IPD game with stochastic strategies (i.e., real number strings). Thus agents in Fig. 11 (b) and Fig. 12 (b) can be viewed as having similar environments, which are different from Fig. 11 (a) and Fig. 12 (a). As a result, Fig. 11 (b) and Fig. 12 (b) are somewhat similar to each other and different from Fig. 11 (a) and Fig. 12 (a). For example, we can see that the average percentage of "111" is much smaller in Fig. 11 (b) and Fig. 12 (b) than Fig. 11 (a) and Fig. 12 (a). This may be because "111" cannot appropriately handle probabilistic strategies represented by real number strings.

V. CONCLUSIONS

We discussed the evolution of strategies for a spatial IPD game where each agent has a different representation scheme. A population of agents in our IPD game can be viewed as a mixture of different species (i.e., ecology since strings of different representation schemes are not recombined). The main difficulty in computational experiments using a mixture of different species is the decrease in the number of qualified neighbors as mates for recombination. That is, some agents may have no neighbors with the same representation scheme. In this case, those agents cannot generate their new strategies by recombination. If agents are not allowed to play the IPD game against their neighbors with different representation schemes, some agents may have no opponents to play the IPD game. In this paper, we proposed an idea of using a larger neighborhood structure only for those agents who have no neighbors with the same representation scheme.

Using this idea, we examined the evolution of IPD game strategies through computational experiments where the three settings were compared: a homogeneous population of a single type of strings, a heterogeneous population of two types of strings, and a heterogeneous population of four types of strings. It was demonstrated that similar results were obtained from different types of strings when they were used in a heterogeneous population with the IPD game between different types of strings. It was also demonstrated that the IPD game between different types of strings increased the average payoff in a heterogeneous population of four types of strings.

Whereas we reported some interesting results, we could not clearly explain why those results were obtained (e.g., why similar results were obtained from the four representation schemes in Fig. 9). In our computational experiments, we always used the 11×11 grid-world. A larger grid-world may be needed when we examine five or more representation schemes. In this paper, we only used string-based representation schemes. It would be interesting if we could examine a variety of representation schemes (e.g., decision trees and neural networks). All of these studies are left as future research topics.

REFERENCES

[1] D. Ashlock, E. Y. Kim, and N. Leahy, "Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints," *IEEE Trans. on Systems, Man, and Cybernetics: Part C*, vol. 36, no. 4, pp. 464-475, July 2006.

[2] D. Ashlock and E. Y. Kim, "Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies," *IEEE Trans. on Evolutionary Computation*, vol. 12, no. 5, pp. 647-659, October 2008.

[3] D. Ashlock, E. Y. Kim, and W. Ashlock, "Fingerprint analysis of the noisy prisoner's dilemma using a finite-state representation," *IEEE Trans. on Computational Intelligence and AI in Games*, vol. 1, no. 2, pp. 154-167, June 2009.

[4] H. Ishibuchi and N. Namikawa, "Evolution of iterated prisoner's dilemma game strategies in structured demes under random pairing in

game playing," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 6, pp. 552-561, December 2005.

[5] 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.

[6] 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, Korea, August 31 - September 3, 2011.

[7] H. Ishibuchi, K. Hoshino, and Y. Nojima, "Strategy Evolution in a Spatial IPD Game where Each Agent is not Allowed to Play against Itself," *Proc. of 2012 IEEE Congress on Evolutionary Computation*, Brisbane, Australia, June 10-15, 2012 (in press).

[8] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms 2*. San Mateo, CA: Morgan Kaufman, pp. 187-202, 1993.