H. Ishibuchi, K. Hoshino, and Y. Nojima "Strategy evolution in a spacial IPD game where each agent is not allowed to play against itself," Proc. of 2012 IEEE Congress on Evolutionary Computation, pp. 688-695, Brisbane, Austraria, June, 10-15, 2012.

Strategy Evolution in a Spatial IPD Game where Each Agent is not Allowed to Play against Itself

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*—Evolution of cooperative behavior has been examined in many studies on the IPD (Iterated Prisoner's Dilemma) game under various conditions. In some studies, each agent is allowed to play against itself. However, this setting is somewhat strange because we do not play any real-world games against ourselves. In this paper, we examine the effect of this somewhat strange setting on the evolution of cooperative behavior in a spatial IPD game. Two cases are compared with each other: Each agent is allowed to play against itself in one case and not allowed to do so in the other case. It is shown through computational experiments that similar results are obtained from these two cases when opponents of each agent are selected from a large number of its neighbors. However, the difference between the two cases is large when the number of neighbors is small. Actually the evolution of cooperative behavior is strongly facilitated by allowing each agent to play against itself when the number of neighbors is small. Our computational experiments are performed on a spatial IPD game with various specifications of the neighborhood size where binary and real number strings are used as game strategies.**

Keywords - Iterated prisoner's dilemma, spatial IPD game, evolutionary games, evolution of cooperative behavior, strategy representation, neighborhood size

I. INTRODUCTION

Prisoner's dilemma is a well-known two-player zero-sum game. Its iterated version, which has been referred to as the IPD (iterated prisoner's dilemma) game, has been used to examine the evolution of cooperative behavior [1]-[3]. For example, Ashlock et al. [4]-[6] demonstrated that the choice of a representation scheme for encoding game strategies has large effects on the evolution of cooperative behavior. That is, totally different results were obtained from different representation schemes in their studies. They examined various representation schemes for encoding game strategies such as a finite-state machine, a neural network and a look-up table. In our former studies [7]-[9], it was shown that the evolution of cooperative behavior in a spatial version of the IPD game depends on the size of neighborhood structures. We used two neighborhood structures: One is for local opponent selection in the IPD game and the other is for local parent selection in strategy evolution.

In the spatial IPD game of our former studies [7]-[9], each agent was allowed to play the IPD game against itself. That is, each agent was included in its own neighborhood for local opponent selection. Whereas this setting of opponent selection has been used in some other studies on the IPD game, it is somewhat strange for each agent to play the IPD game against itself. This is because we do not play any real-world games against ourselves. In this paper, we examine the effect of this somewhat strange setting of opponent selection on the evolution of cooperative behavior in the spatial IPD game.

In this paper, we use two neighborhood structures for local opponent selection and local parent selection as in our former studies [7]-[9]. Two parent strategies, which are recombined to generate a new strategy, are selected for each agent from its local parent selection neighborhood. Since it is not strange for an agent to generate a new strategy from its current strategy, each agent is included in its parent selection neighborhood. However, it is somewhat strange for an agent to play the IPD game against itself. So we examine two cases with respect to the specification of local opponent selection neighborhood. In one case, each agent is included in its local opponent selection neighborhood. This means that each agent is allowed to play against itself as in our former studies [7]-[9]. In the other case, each agent is not included in its local opponent selection neighborhood. That is, each agent is not allowed to play against itself. We compare these two cases with each other through computational experiments on our spatial IPD game with the two neighborhood structures. Experimental results demonstrate the relation between the neighborhood size for local opponent selection and the effect of allowing each agent to play against itself. We also show some experimental results for a mixture of heterogeneous agents with different representation schemes. We use two representation schemes: binary strings and real number strings. One of these two types of strings is assigned to each agent in a cell of a grid-world. In our spatial IPD game, it is possible that some agents have no neighbors with the same representation schemes. We discuss the handling of such a situation for local opponent selection and local parent selection.

This paper is organized as follows. We first explain our spatial IPD game with two neighborhood structures in Section II. Next we perform simple mathematical analysis of strategy evolution using expected payoff values to compare the two cases where agents are allowed or not allowed to play against themselves in Section III. These two cases are compared with each other through computational experiments in Section IV. Finally we conclude this paper in Section V.

II. OUR SPATIAL IPD GAME

A. IPD Game

We use a well-known standard payoff matrix in Table I where two players (an agent and its opponent) are supposed to simultaneously choose one of the two actions: "C: Cooperate" or "D: Defect". When both players cooperate, each receives three points as "Agent: 3" and "Opponent: 3" in Table I. When both players defect, each receives one point as "Agent: 1" and "Opponent: 1". The highest payoff of five points is obtained by defecting when the opponent cooperates. In this case, the opponent receives the lowest payoff of zero point (i.e., "Agent: 5" and "Opponent: 0"). The agent receives the lowest payoff of zero point by cooperating when the opponent defects (i.e., "Agent: 0" and "Opponent: 5").

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

Such a simultaneous choice of an action by each player is a single round of the prisoner's dilemma game. In its iterated version, the game is played between the same two players for a pre-specified number of rounds. Such an iterated version has been often referred to as the IPD game. In our computational experiments in this paper, the prisoner's dilemma game is iterated for 100 rounds. Our computational experiments, however, are performed under the following assumption as in many other studies on the IPD game: No player knows in advance how many rounds of the prisoner's dilemma game will be iterated. That is, no player knows when the iteration of the iterated prisoner's game will be terminated.

B. IPD Game Strategies

An agent's strategy determines its next action based on a finite memory of previous plays of the game. In this paper, we use only the previous action of the opponent to determine the agent's action. Binary strings have often been used to represent strategies where "1" and "0" usually mean "cooperate" and "defect", respectively. In Table II, we show a three-bit binary string strategy "101" called TFT (tit-for-tat). This strategy cooperates at the first round and then cooperates at each round only when the opponent cooperated in the previous round.

TABLE II. THREE-BIT BINARY STRATEGY "101" CALLED TFT.

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 the TFT strategy "101" can be interpreted as the probability of "C: Cooperate". For example, the first bit value "1" of "101" means that the agent cooperates in the first round with the probability 1. Using this interpretation, we can easily generalize binary strings to real number strings. For example, a real number string "0.9 0.1 0.7" can be viewed as a stochastic strategy that cooperates with the probability 0.9 in the first round, cooperates with the probability 0.1 in response to the opponent's defection, and cooperate with the probability 0.7 in response to the opponent's cooperation. Of course, all elements of real number strings should be in the closed interval [0, 1]. In this paper, we use binary strings and real number strings of length three. Real number strings are stochastic strategies while binary strings are deterministic strategies.

C. Spatial IPD Game with Two Neighborhood Structures

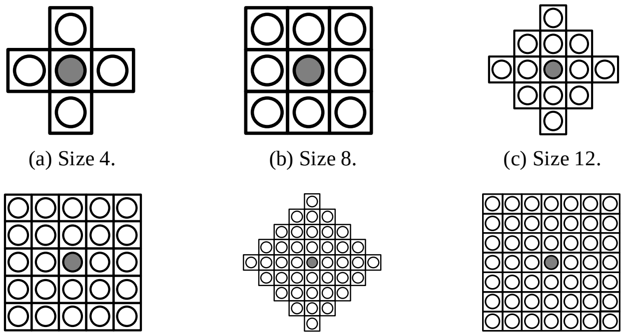
The IPD game has been extended in various directions in the literature. One direction is to introduce a spatial structure of agents to the IPD game [10]-[12]. Each agent is usually fixed spatially in a cell of a two-dimensional grid-world where the IPD game is performed between neighboring players. As in our former studies [7]-[9], we use an 11×11 grid-world with the torus structure. Since each cell has a single agent, the total number of agents is 121. Each agent has its own IPD game strategy. Strategies of the 121 agents are evolved by selection, crossover and mutation as we will explain in Section III.

In spatial versions of the IPD game, each agent plays the IPD game against its neighbors. We denote the set of neighbors of Agent i by $N_{\mathrm{IPD}}(i)$. Since opponents of Agent i are selected from $N_{\mathrm{IPD}}(i)$, $N_{\mathrm{IPD}}(i)$ can be viewed as a neighborhood structure for local opponent selection. As we have already explained, the following two cases are examined in this paper:

Case 1: $N_{\mathrm{IPD}}(i)$ includes Agent i itself. In this case, each agent is allowed to play the IPD game against itself.

Case 2: $N_{\mathrm{IPD}}(i)$ does not include Agent i itself. In this case, each agent is not allowed to play the IPD game against itself.

Examples of $N_{\mathrm{IPD}}(i)$ in Case 2 are shown in Fig. 1 where $N_{\mathrm{IPD}}(i)$ has 4, 8, 12, 24, 40 and 48 neighbors excluding Agent i itself. In this paper, we examine all of these six specifications of the neighborhood structure $N_{\mathrm{IPD}}(i)$. For comparison, we also examine an extreme specification of $N_{\mathrm{IPD}}(i)$ 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 $N_{\mathrm{IPD}}(i)$. Of course, we also examine the corresponding seven specifications of $N_{\mathrm{IPD}}(i)$ in Case 1 where Agent i is included in $N_{\mathrm{IPD}}(i)$.



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

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

Figure 1. Example of $N_{\mathrm{IPD}}(i)$ excluding Agent i itself (i.e., Case 2).

When $N_{\mathrm{IPD}}(i)$ includes five or less neighbors as in Fig. 1 (a), Agent i plays the IPD game against all of its neighbors in $N_{\mathrm{IPD}}(i)$. Average payoff of Agent i per round is calculated over those executions of the IPD game. The calculated average payoff is used as the fitness of the strategy of Agent i. When $N_{\mathrm{IPD}}(i)$ includes more than five neighbors as in Fig. 1 (b)-(f), five opponents are randomly selected from $N_{\mathrm{IPD}}(i)$ for Agent i (i.e., random sampling of five opponents from $N_{\mathrm{IPD}}(i)$ without replacement). Average payoff is calculated over the executions of the IPD game against the selected five opponents.

After the fitness value of the strategy of each agent is calculated by the IPD game, a new strategy for each agent is generated from its neighbors through selection, crossover and mutation. Parents of a new strategy of Agent i are selected from its neighbors. Let us denote a set of neighbors of Agent i for local parent selection by $N_{\mathrm{GA}}(i)$. Parents of a new strategy of Agent i are selected from $N_{\mathrm{GA}}(i)$. Thus $N_{\mathrm{GA}}(i)$ can be viewed as a neighborhood structure for local parent selection. It should be noted that the two neighborhood structures (i.e., $N_{\mathrm{IPD}}(i)$ for local opponent selection and $N_{\mathrm{GA}}(i)$ for local parent selection) are not necessarily the same. It should be also noted that $N_{\mathrm{GA}}(i)$ includes Agent i itself. This means that a new strategy of Agent i can be generated from its current strategy. In this paper, we examine six specifications of $N_{\mathrm{GA}}(i)$, which correspond to the six neighborhood structures of $N_{\mathrm{IPD}}(i)$ in Fig. 1 (including Agent i). We also examine an additional extreme specification of $N_{\mathrm{GA}}(i)$ with all the 121 agents in the 11×11 grid-world.

The main characteristic feature of our spatial IPD game in this paper and our former studies [7]-[9] is the use of the two neighborhood structures: One is for local opponent selection, and the other is for local parent selection. This feature has been motivated by the idea of structured demes [13]-[16] with two neighborhood structures: One is for the modeling of daily interaction with others, and the other is for the modeling of mating. There exist a number of real-world situations in nature with those two neighborhood structures such as territorial animals and plants. We use the two neighborhood structures to examine the effects of local opponent selection and local parent selection on the evolution of cooperative behavior separately from each other. Similar ideas of two neighborhood structures were used in a different spatial IPD game [17] and a cellular evolutionary algorithm for function optimization [18].

III. Mathematical Analysis of Strategy Evolution

A. Evolution of IPD Game Strategies

An initial population is randomly generated. When three-bit binary strings are used, each bit value is randomly specified as 0 or 1 with the same probability. When real number strings of length three are used, real numbers are randomly specified using the uniform probability distribution over the unit interval [0, 1]. In this manner, an initial population with 121 strategies is generated. Each strategy is assigned to a different cell (i.e., to a different agent) in the 11×11 grid-world. The fitness of each strategy is calculated as its average payoff in the IPD game.

Two parents are selected for Agent i from its local parent selection neighborhood $N_{\mathrm{GA}}(i)$ by binary tournament selection with replacement. A new strategy is generated by applying crossover and mutation to 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 [19]) 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 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 strategy for that agent. That is, the current population of 121 strategies is entirely replaced with the newly generated 121 strategies. The fitness evaluation through the IPD game and the generation update by genetic operations are iterated for 1000 generations in our computational experiments.

B. Simple Mathematical Analysis

When we use three-bit binary strings, the total number of strategies is $2^3 = 8$. Let us denote these eight strategies by S_i ($i = 0, 1, ..., 7$) as shown in the first column of Table III. In this case, we can calculate the average payoff of an agent against an opponent for each of the 8×8 combinations of their strategy assignment. Each entry of Table III shows the calculated average payoff of the corresponding agent's strategy against the corresponding opponent's strategy. For example, the average payoff of the agent with "always defect strategy 000" is 5.00 against the opponent with "always cooperate strategy 111", which is shown in the top-right field corresponding to the agent's strategy "000" and the opponent's strategy "111" among the 8×8 combinations in Table III. Let $P(S_j, S_k)$ be the average payoff of the agent's strategy S_j against the opponent's strategy S_k. For example, $P(S_3, S_0) = 0.01$ and $P(S_3, S_3) = 2.98$ in Table III where S_0 = "000" and S_3 = "011". The average payoff of each strategy against itself is underlined and highlighted by boldface in Table III.

TABLE III. Average Payoff from the IPD Game with 100 Rounds between Binary String Strategies.

Strategy of Agent	Strategy of Opponent								P_{Ave}
	S_0: 000	S_1: 001	S_2: 010	S_3: 011	S_4: 100	S_5: 101	S_6: 110	S_7: 111	
S_0: 000	**1.00**	1.00	4.96	4.96	1.04	1.04	5.00	5.00	**3.00**
S_1: 001	1.00	**1.00**	2.25	3.00	1.03	2.50	2.25	3.02	**2.01**
S_2: 010	0.01	2.25	**2.00**	4.94	0.06	2.25	5.00	5.00	**2.69**
S_3: 011	0.01	2.95	0.04	**2.98**	0.05	2.99	0.08	3.02	**1.52**
S_4: 100	0.99	1.03	4.91	4.95	**1.02**	1.06	4.94	4.98	**2.99**

S_5: 101	0.99	2.50	2.25	2.99	1.01	**3.00**	2.25	3.00	**2.25**
S_6: 110	0.00	2.25	0.00	4.93	0.04	2.25	**2.00**	4.98	**2.06**
S_7: 111	0.00	2.97	0.00	2.97	0.03	3.00	0.03	**3.00**	**1.50**

The average payoff $P_{\text{Ave}}(S_j)$ of each strategy S_j against all the eight strategies is calculated as

$$P_{\text{Ave}}(S_j) = \frac{1}{8}\sum_{k=0}^{7} P(S_j, S_k)\,,\ j = 0, 1, \ldots, 7, \qquad (1)$$

which is shown in the last column of Table III using bold font. For example, $P_{\text{Ave}}(S_0) = 3.00$ and $P_{\text{Ave}}(S_1) = 2.01$.

In an initial population, strategies are randomly generated. This means that each agent has one of the eight strategies with the same probability. When $N_{\text{IPD}}(i)$ does not include Agent i itself (i.e., when each agent is not allowed to play against itself), each opponent of Agent i can be viewed as having one of the eight strategies with the same probability. Thus the expected average payoff of Agent i with the strategy S_j against its opponents is the same as its average payoff $P_{\text{Ave}}(S_j)$ shown in the last column of Table III. Let us denote the expected average payoff of an agent with the strategy S_j as $E_{\text{C2}}(S_j)$, which is the same as $P_{\text{Ave}}(S_j)$. The subscript "C2" of $E_{\text{C2}}(S_j)$ denotes Case 2 where each agent is not allowed to play against itself. Note that the expected average payoff $E_{\text{C2}}(S_j)$ is independent of the size of the neighborhood structure $N_{\text{IPD}}(i)$ since any opponent has a randomly specified strategy in the initial population.

Let us discuss the expected average payoff $E_{\text{C1}}(S_j)$ of an agent with the strategy S_j in Case 1 where each agent is allowed to play against itself (i.e., Agent i is in $N_{\text{IPD}}(i)$). When we use $N_{\text{IPD}}(i)$ with four neighbors in Fig. 1 (a), Agent i plays the IPD game against all the four neighbors and Agent i itself in Case 1. Since each of the four neighbors has one of the eight strategies with the same probability, $E_{\text{C1}}(S_j)$ can be calculated as

$$E_{\text{C1}}(S_j) = \frac{1}{5}\{P(S_j, S_j) + 4 \times P_{\text{Ave}}(S_j)\}\,,\ j = 0, 1, ..., 7. \ (2)$$

Now we can compare $E_{\text{C1}}(S_j)$ with $E_{\text{C2}}(S_j)$ for each strategy when we use the neighborhood structure $N_{\text{IPD}}(i)$ with four neighbors in Fig. 1 (a) as follows:

$E_{\text{C1}}(S_0) = 2.60 < 3.00 = E_{\text{C2}}(S_0)$ for S_0 = "000"
$E_{\text{C1}}(S_1) = 1.81 < 2.01 = E_{\text{C2}}(S_1)$ for S_1 = "001"
$E_{\text{C1}}(S_2) = 2.55 < 2.69 = E_{\text{C2}}(S_2)$ for S_2 = "010"
$E_{\text{C1}}(S_3) = 1.81 > 1.52 = E_{\text{C2}}(S_3)$ for S_3 = "011"
$E_{\text{C1}}(S_4) = 2.60 < 2.99 = E_{\text{C2}}(S_4)$ for S_4 = "100"
$E_{\text{C1}}(S_5) = 2.40 > 2.25 = E_{\text{C2}}(S_5)$ for S_5 = "101"
$E_{\text{C1}}(S_6) = 2.05 < 2.06 = E_{\text{C2}}(S_6)$ for S_6 = "110"
$E_{\text{C1}}(S_7) = 1.80 > 1.50 = E_{\text{C2}}(S_7)$ for S_7 = "111"

These calculations show that the average expected payoff of uncooperative strategies "000" and "100" is most heavily decreased by allowing each agent to play against itself (i.e., by changing the setting from Case 2 to Case 1) while that of cooperative strategies "111" and "011" is most heavily increased. It should be noted that "100" always defects except for the first round while "011" always cooperates except for the first round. These discussions suggest that the evolution of cooperative behavior is facilitated by the IPD game of each agent against itself when the size of the neighborhood structure $N_{\text{IPD}}(i)$ is very small. This effect will be examined through computational experiments later in this paper.

When $N_{\text{IPD}}(i)$ is large, the effect of the IPD game of each agent against itself is small. For example, let us assume that $N_{\text{IPD}}(i)$ with 48 neighbors in Fig. 1 (f) is used. In Case 1, Agent i is also in $N_{\text{IPD}}(i)$. In this case, $E_{\text{C1}}(S_j)$ can be formulated as

$$E_{\text{C1}}(S_j) = \frac{1}{49}\{P(S_j, S_j) + 48 \times P_{\text{Ave}}(S_j)\}\,,\ j = 0, 1, ..., 7. \ (3)$$

As we have already explained, $E_{\text{C2}}(S_j)$ is independent of the neighborhood size. Thus $E_{\text{C1}}(S_j)$ and $E_{\text{C2}}(S_j)$ are compared as follows when we use $N_{\text{IPD}}(i)$ with 48 neighbors in Fig. 1 (f):

$E_{\text{C1}}(S_0) = 2.96 < 3.00 = E_{\text{C2}}(S_0)$ for S_0 = "000"
$E_{\text{C1}}(S_1) = 19.9 < 2.01 = E_{\text{C2}}(S_1)$ for S_1 = "001"
$E_{\text{C1}}(S_2) = 2.68 < 2.69 = E_{\text{C2}}(S_2)$ for S_2 = "010"
$E_{\text{C1}}(S_3) = 1.55 > 1.52 = E_{\text{C2}}(S_3)$ for S_3 = "011"
$E_{\text{C1}}(S_4) = 2.95 < 2.99 = E_{\text{C2}}(S_4)$ for S_4 = "100"
$E_{\text{C1}}(S_5) = 2.27 > 2.25 = E_{\text{C2}}(S_5)$ for S_5 = "101"
$E_{\text{C1}}(S_6) = 2.06 = 2.06 = E_{\text{C2}}(S_6)$ for S_6 = "110"
$E_{\text{C1}}(S_7) = 1.53 > 1.50 = E_{\text{C2}}(S_7)$ for S_7 = "111"

The maximum difference between $E_{\text{C1}}(S_j)$ and $E_{\text{C2}}(S_j)$ is only 0.04, which is much smaller than 0.4 in the previous discussions for the neighborhood structure with four neighbors in Fig. 1 (a). As shown from these calculations, the increase in the size of $N_{\text{IPD}}(i)$ deceases the difference between the two cases: Case 1 and Case 2. This will be also examined through computational experiments later in this paper.

Let us also calculate the average payoff for five real number strings in Table IV: "0.1 0.2 0.1", "0.9 0.1 0.2", "0.4 0.6 0.5", "0.9 0.2 0.8" and "0.9 0.8 0.9". As we have already explained, these real number strings are stochastic strategies for the IPD game. We assume that the IPD game is performed for 100 rounds for each of the 5×5 combinations of the five strategies in Table IV. Since our strategies are stochastic, different results are obtained from each execution of the IPD game even when we use the same pair of strategies. So we calculated the average results over 10,000 executions of the IPD game with 100 rounds for each of the 5×5 combinations of the five strategies. Experimental results are shown in Table IV in the same manner as in Table III with the eight binary strings of length three.

TABLE IV. AVERAGE PAYOFF FROM THE IPD GAME WITH 100 ROUNDS BETWEEN REAL NUMBER STRING STRATEGIES

Strategy of Agent	Strategy of Opponent					P_{Ave}
	0.1 0.2 0.1	0.9 0.1 0.2	0.4 0.6 0.5	0.9 0.2 0.8	0.9 0.8 0.9	
0.1 0.2 0.1	**1.51**	1.30	3.11	2.01	4.04	**2.39**
0.9 0.1 0.2	1.60	**1.34**	3.06	1.97	3.93	**2.38**
0.4 0.6 0.5	0.90	0.99	**2.33**	2.29	3.46	**1.99**
0.9 0.2 0.8	1.31	1.21	2.36	**2.27**	3.13	**2.06**
0.9 0.8 0.9	0.56	0.78	1.76	2.40	**2.88**	**1.68**

Let us assume that one of these five real number strategies is randomly assigned to each agent with the same probability.

Under this assumption, the expected average payoff $E_{C2}(S_j)$ of each strategy S_j in Case 2 is the same as the average payoff $P_{Ave}(S_j)$ in the last column of Table IV. Let us further assume that $N_{IPD}(i)$ with four neighbors in Fig. 1 (a) is used. Under these settings, the expected average payoff $E_{C1}(S_j)$ of each strategy S_j in Case 1 is calculated from Eq.(2) as follows:

$$E_{C1}(S_1) = 2.21 < 2.39 = E_{C2}(S_1) \text{ for } S_1 = \text{“0.1 0.2 0.1”}$$
$$E_{C1}(S_2) = 2.17 < 2.38 = E_{C2}(S_2) \text{ for } S_2 = \text{“0.9 0.1 0.2”}$$
$$E_{C1}(S_3) = 2.06 > 1.99 = E_{C2}(S_3) \text{ for } S_3 = \text{“0.4 0.6 0.5”}$$
$$E_{C1}(S_4) = 2.10 > 2.06 = E_{C2}(S_4) \text{ for } S_4 = \text{“0.9 0.2 0.8”}$$
$$E_{C1}(S_5) = 1.92 > 1.68 = E_{C2}(S_5) \text{ for } S_5 = \text{“0.9 0.8 0.9”}$$

Among these five strategies, the first two frequently defect (since all the three real numbers of these strategies are close to 0). The IPD game of each agent against itself decreases the expected average payoff of these two uncooperative strategies by about 0.2. On the other hand, the last strategy frequently cooperates since the three real numbers of this strategy are close to 1. The IPD game of each agent against itself (i.e., the change from Case 2 to Case 1) most heavily increases the expected average payoff of this cooperative strategy. These discussions suggest that the evolution of cooperative behavior is easier in Case 1 than Case 2. The two cases will be compared through computational experiments later in this paper for real number string strategies as well as binary string strategies.

IV. Computational Experiments

A. Settings of Computational Experiments

In this section, we compare the following two cases with each other through computational experiments:

Case 1: $N_{IPD}(i)$ includes Agent *i* itself. In this case, each agent is allowed to play the IPD game against itself.

Case 2: $N_{IPD}(i)$ does not include Agent *i* itself. In this case, each agent is not allowed to play the IPD game against itself.

In Case 1, our computational experiments are performed in exactly the same manner as in our previous study [9]. Those results in Case 1 are compared with experimental results in Case 2. Parameter specifications in this paper and our previous study [9] are summarized as follows:

Parameters in 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: 100 rounds.

Parameters in the Evolutionary Algorithm:

Population size: 121 in the 11×11 grid-world,
Representation schemes: Three-bit binary strings, and
real number strings of length 3,
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.

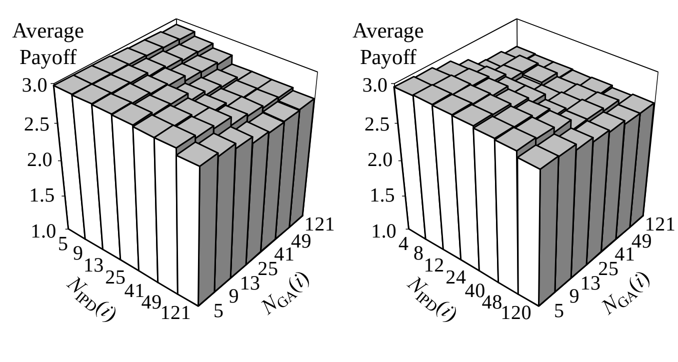
The six neighborhood structures in Fig. 1 were depicted as $N_{IPD}(i)$ for Case 2. Thus, in Case 1, Agent *i* is added to each neighborhood structure $N_{IPD}(i)$. Agent *i* is also included in each neighborhood structure $N_{GA}(i)$ for local parent selection.

All of our experimental results in this section are average results over 500 runs for each setting.

B. Results with Homogeneous Population

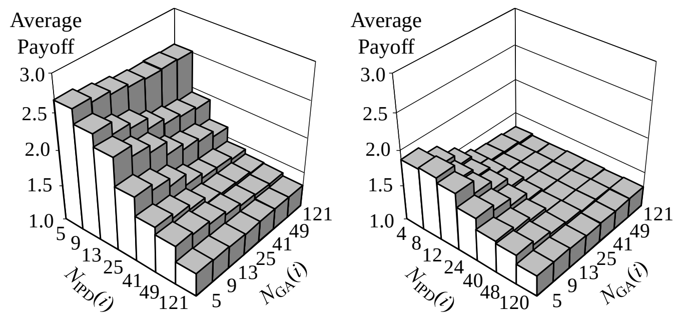
First we show experimental results with a homogeneous population of three-bit binary strings. That is, all of the 121 agents in the 11×11 grid-world use three-bit binary strings. Under this setting, we calculate the average payoff of all agents over 500 runs through 1000 generations for each of the $7 \times 7 = 49$ combinations of the two neighborhood structures: $N_{IPD}(i)$ for local opponent selection in the IPD game and $N_{GA}(i)$ for local parent selection in the evolution algorithm for IPD game strategies. These computational experiments are performed for the two cases of $N_{IPD}(i)$: Case 1 and Case 2. We show experimental results in Fig. 2. In Fig. 2, the average payoff is slightly decreased by changing from Case 1 to Case 2. In other words, the average payoff is slightly decreased by forbidding each agent to play against itself. Even in Case 2 (i.e., in Fig. 2 (b)), the average payoff is always higher than 2.5 independent of the specifications of the two neighborhood structures.

Next we show experimental results with a homogeneous population of real number strings. That is, all of the 121 agents in the 11×11 grid-world use real number strings of length 3. Experimental results are shown in Fig. 3 in the same manner as in Fig. 2. The average payoff is much more severely decreased by changing from Case 1 to Case 2 in Fig. 3 with real number strings than Fig. 2 with binary strings. In Case 2 (i.e., in Fig. 3 (b)), the average payoff is always lower than 2.0 independent of the specifications of the two neighborhood structures. That is, the evolution of cooperative behavior among agents with real number strings of length 3 is very difficult when each agent is not allowed to play against itself (i.e., in Fig. 3 (b) in Case 2).



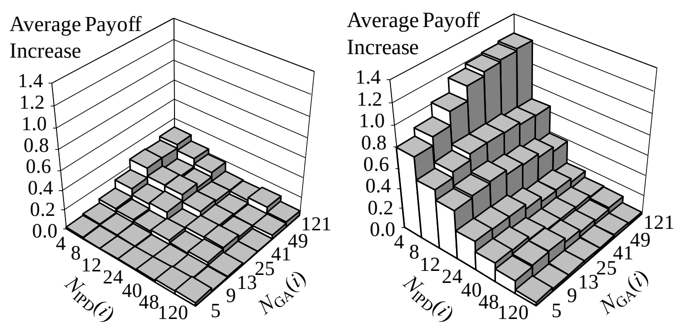
(a) Experimental results in Case 1. (b) Experimental results in Case 2.

Figure 2. Average payoff obtained from a homogeneous population of 121 three-bit binary strings.



(a) Experimental results in Case 1. (b) Experimental results in Case 2.

Figure 3. Average payoff obtained from a homogeneous population of 121 rearl number strings of length 3.



(a) Three-bit binary strings. (b) Real number strings of length 3.

Figure 4. Increase in the average payoff from Case 2 to Case 1. Each agent is allowed to play against itself in Case 1 while it is not allowed in Case 2.

In order to clearly show the effect of allowing each agent to play against itself, we calculate the increase in the average payoff from Case 2 to Case 1 (i.e., from Fig. 2 (b) to Fig. 2 (a), and from Fig. 3 (b) to Fig. 3 (a)). The calculated increase in the average payoff is shown in Fig. 4.

Fig. 4 can be viewed as showing the positive effect of the IPD game of each agent against itself on the evolution of cooperative behavior in the spatial IPD game. From Fig. 4, we can obtain the following observations:

(i) The effect of allowing each agent to play against itself on the evolution of cooperative behavior is large when the smallest local opponent selection neighborhood $N_{IPD}(i)$ is used in a homogeneous population of real number strings of length 3 in Fig. 4 (b).

(ii) This effect is decreased in Fig. 4 (b) by increasing the size of the local opponent selection neighborhood $N_{IPD}(i)$. This is because the effect of the IPD game of each agent against itself on its expected average payoff is decreased by increasing the size of $N_{IPD}(i)$ as explained in Section III.

(iii) This effect is much smaller in Fig. 4 (a) with three-bit binary strings than Fig. 4 (b) with real number strings of length 3. This is because the evolution of cooperative behavior among three-bit binary strings is not difficult even when each agent is not allowed to play against itself in Fig. 2 (b). However, the evolution of cooperative behavior among real number strings of length 3 is very difficult when agent is not allowed to play against itself in Fig. 3 (b).

C. Results with Heterogeneous population

As in our former studies [8], [9], we randomly choose 50% of agents and assign them a three-bit binary string as the representation scheme of their IPD strategies. A real number string of length 3 is assigned to the remaining agents. An example of such a random assignment of the two representation schemes is shown in Fig. 5. Each agent continues to use the assigned representation scheme during the evolution of game strategies with 1000 generations. The assignment of 50% binary strings and 50% real number strings to the 121 agents is randomly initialized at each of 500 runs for each setting. As in the previous subsection, we examine the two cases (i.e., Case 1 and Case 2) with respect to the local opponent selection neighborhood $N_{IPD}(i)$.

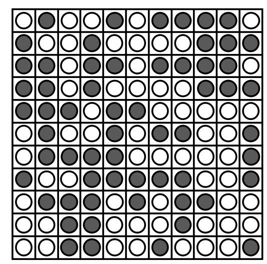


Figure 5. Heterogeneous population with 50% three-bit binary strings (closed circles) and 50% real-number strings of length 3 (open circles). In this figure, 60 agents have binary strings while 61 agents have real number strings.

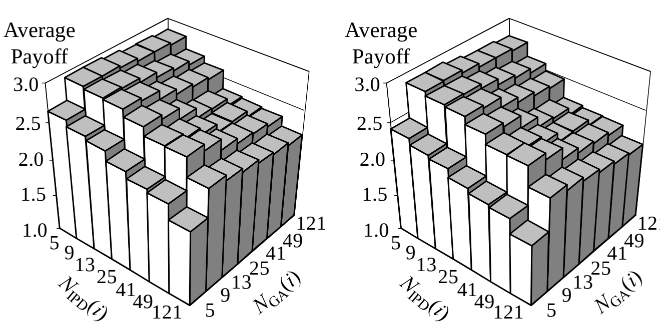
When an offspring strategy is generated for Agent *i*, a pair of parent strategies with the same representation scheme as Agent *i* are selected from its local selection neighborhood $N_{GA}(i)$. We never apply any crossover operation to a pair of parent strategies with different representation schemes. Since Agent *i* itself is included in $N_{GA}(i)$, $N_{GA}(i)$ always has at least one agent with the same representation scheme as Agent *i*. Thus an offspring strategy can be generated for every agent. In an extreme situation where all neighbors in $N_{GA}(i)$ have a different representation scheme from Agent *i*, Agent *i* is always selected as a parent to generate its offspring strategy. As a result, crossover becomes meaningless. Thus an offspring strategy is always generated for Agent *i* from its current strategy by mutation when no other neighbors in $N_{GA}(i)$ have the same representation scheme as Agent *i*.

With respect to the IPD game between agents with different representation schemes, we examine two settings. In one setting, the IPD game is performed between two agents regardless of their representation schemes. However, in the other setting, the IPD game is performed between two agents only when they have the same representation scheme. When each agent is allowed to play against itself, the local opponent selection neighborhood $N_{IPD}(i)$ always includes at least one agent (i.e., Agent *i* itself) with the same representation scheme as Agent *i*. Thus the fitness of Agent *i* can be always evaluated through the execution of the IPD game between Agent *i* and its opponents in $N_{\mathrm{IPD}}(i)$. However, when each agent is not allowed to play against itself, it is possible that $N_{\mathrm{IPD}}(i)$ includes no agents with the same representation scheme as Agent *i*. In this situation, the fitness of Agent *i* cannot be evaluated when the IPD game is not played between two agents with different representation schemes. Thus an additional troubleshooting procedure is needed in such a special situation where the fitness of Agent *i* cannot be evaluated through the IPD game. In our computational experiments, we handle such a special situation by using a larger neighborhood structure as $N_{\mathrm{IPD}}(i)$ only for Agent *i*. For example, the fitness of Agent *i* cannot be evaluated under $N_{\mathrm{IPD}}(i)$ of size 4 in Fig. 1 (a), that of size 8 in Fig. 1 (b) is used as $N_{\mathrm{IPD}}(i)$ only for Agent *i*. When the fitness of Agent *i* cannot be evaluated under $N_{\mathrm{IPD}}(i)$ of size 8 again, that of size 12 in Fig. 1 (c) is used as $N_{\mathrm{IPD}}(i)$ only for Agent *i*. In this manner, the fitness of each agent is evaluated through the execution of the IPD game. It should be noted that this trouble shooting procedure is needed only for the following setting: Each agent can play the IPD game only against its neighbors with the same representation scheme (i.e., each agent is not allowed to play the IPD game against itself, and the IPD game is not played between agents with different representation schemes).

Experimental results are summarized in Figs. 6-9 for the following four settings:

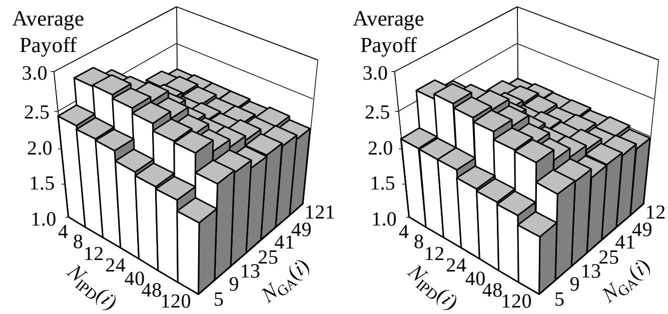
1. The IPD game is played between two agents with different representation schemes. Each agent is allowed to play against itself (Case 1). Results are shown in Fig. 6.
2. The IPD game is played between two agents with different representation schemes. Each agent is not allowed to play against itself (Case 2). Results are shown in Fig. 7.
3. The IPD game is not played between two agents with different representation schemes. Each agent is allowed to play against itself (Case 1). Results are shown in Fig. 8.
4. The IPD game is not played between two agents with different representation schemes. Each agent is not allowed to play against itself (Case 2). Results are shown in Fig. 9.

First, let us compare between Fig. 6 (a) and Fig. 6 (b). When a homogeneous population was used, totally different results were obtained from three-bit binary strings in Fig. 2 and real number strings of length 3 in Fig. 3. However, similar results are obtained from these two representation schemes in Fig. 6 when a heterogeneous population is used as in Fig. 5 with 50% binary strings and 50% real number strings. The same observation is obtained from Fig. 7.



(a) Three-bit binary strings. (b) Real number strings of length 3.

Figure 6. Average payoff from heterogeneous populations. The IPD game is played between two agents with different representation schemes. Each agent is allowed to play against itself (i.e., Case 1).

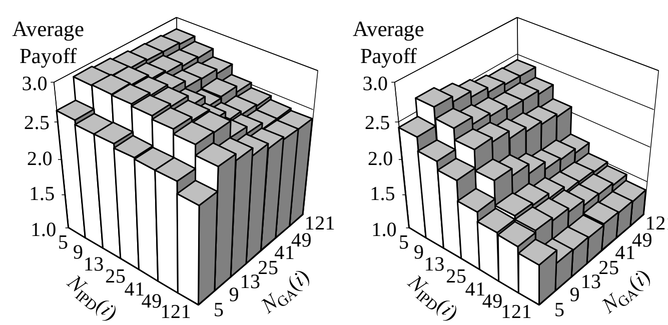


(a) Three-bit binary strings. (b) Real number strings of length 3.

Figure 7. Average payoff from heterogeneous populations. The IPD game is played between two agents with different representation schemes. Each agent is not allowed to play against itself (i.e., Case 2).

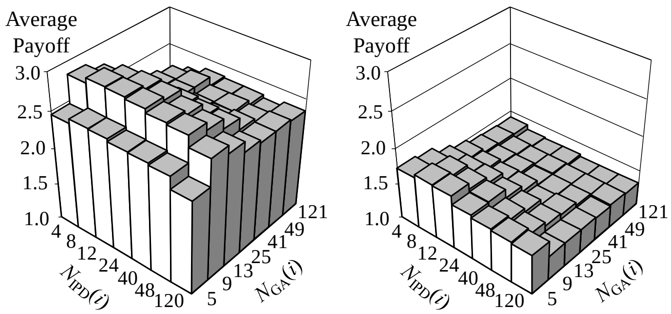
Now, let us compare between Fig. 6 and Fig. 7. The average payoff is increased from Fig. 7 to Fig. 6 by allowing each agent to play against itself. We can see that the IPD game of each agent against itself has a large positive effect on the evolution of cooperative behavior when a small neighborhood structure is used as the local opponent selection neighborhood $N_{\mathrm{IPD}}(i)$ in Fig. 6. This positive effect is similar between the two representation schemes (i.e., the increase in the average payoff of binary string agents from Fig. 7 (a) to Fig. 6 (a) is similar to that of real number string agents from Fig. 7 (b) to Fig. 6 (b)).

While the IPD game is played between two agents with different representation schemes in Fig. 6 and Fig. 7, it is not played between two agents with different representation schemes in Fig. 8 and Fig. 9. That is, strategies with one representation scheme are evolved in Fig. 8 and Fig. 9 independently from strategies with the other representation scheme. Since there exists no interaction between agents with different representation schemes, experimental results in Fig. 8 and Fig. 9 are more similar to Fig. 2 and Fig. 3 with homogeneous populations than Fig. 6 and Fig. 7 with the interaction between the two representation schemes through the IPD game (e.g., Fig. 9 (b) is more similar to Fig. 3 (b) than Fig. 7 (b)). The effect of allowing each agent to play against itself in Fig. 8 and Fig. 9 is also similar to that in Fig. 2 and Fig. 3.



(a) Three-bit binary strings. (b) Real number strings of length 3.

Figure 8. Average payoff from heterogeneous populations. The IPD game is not played between two agents with different representation schemes. Each agent is allowed to play against itself (i.e., Case 1).



(a) Three-bit binary strings. (b) Real number strings of length 3.

Figure 9. Average payoff from heterogeneous populations. The IPD game is not played between two agents with different representation schemes. Each agent is not allowed to play against itself (i.e., Case 2).

In our former studies [8], [9], similar results were obtained from different representation schemes when the IPD game was played between them in Case 1 (i.e., when each agent is allowed to play against itself) as shown in Fig. 6. We can see from Fig. 7 that the same observation is obtained in Case 2 (i.e., when each agent is not allowed to play against itself). It was also observed in [8], [9] that totally different results were obtained from different representation schemes when the IPD game was not played between different representation schemes as shown in Fig. 8. Differences between the two representation schemes are more clearly shown in Fig. 9 in Case 2 than Fig. 8 in Case 1. This is because the positive effect of the IPD game of each agent against itself made the experimental results from the two representation schemes similar to each other when $N_{\mathrm{IPD}}(i)$ is very small (e.g., compare the experimental results between Fig. 8 (a) and Fig. 8 (b) for $N_{\mathrm{IPD}}(i)$ of size 5).

V. CONCLUSIONS

In this paper, we examined the effect of allowing each agent to play against itself in the spatial IPD game. It was clearly shown through computational experiments that the IPD game of each agent against itself has a large positive effect on the evolution of cooperative behavior when the local opponent selection neighborhood is small. This is because (i) each agent is more likely to play against itself under a smaller local opponent selection neighborhood and (ii) cooperative strategies can get a high average payoff from the IPD game against themselves. When we removed this positive effect (i.e., when we did not allow each agent to play against itself), the evolution of cooperative behavior became very difficult among agents with real number strings of length 3. However, a high average payoff was still obtained from real number agents when they were played against binary string agents in a heterogeneous population of 50% binary string agents and 50% real number string agents. Moreover, similar experimental results were obtained from those two representation schemes in such a heterogeneous population when the IPD game was played between them independent of the setting about the IPD game of each agent against itself.

We also discussed the handling of a special situation where an agent has no qualified opponent in its local opponent selection neighborhood. In our computational experiments, we handled such a situation by using a lager neighborhood structure. This idea can be also used for local parent selection when an agent has no qualified parents in its local parent selection neighborhood except for the agent itself. This is left as a future research topic.

When each agent is allowed to play against itself, the size of the local opponent selection neighborhood has a dominant effect on the evolution of cooperative behavior. That is, a high average payoff is almost always obtained when we use a very small neighborhood for local opponent selection. However, when each agent is not allowed to play against itself, the size of the local opponent selection neighborhood does not have such a dominant effect. As a result, we will be able to examine more clearly the effect of other settings such as the size of the local parent selection neighborhood and the percentage of agents with each representation scheme while forbidding each agent to play the IPD game against itself. Examination of the effect of those settings on the evolution of cooperative behavior is also left for future study. Of course, the use of heterogeneous populations with more than two representation schemes is also an interesting future research issue.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. A. Nowak, R. M. May, and K. Sigmund, "The arithmetics of mutual help," *Scientific American*, vol. 272, no. 6, pp. 50-53, June 1995.

[11] A. L. Lloyd, "Computing bouts of the prisoner's dilemma," *Scientific American*, vol. 272, no. 6, pp. 80-83, June 1995.

[12] K. Brauchli, T. Killingback, and M. Doebeli, "Evolution of cooperation in spatially structured populations," *Journal of Theoretical Biology*, vol.

200, no. 4, pp. 405-417, October 1999.

[13] D. S. Wilson, “Structured demes and the evolution of group-advantageous traits,” *The American Naturalist*, vol. 111, no. 977, pp. 157-185, January-February 1977.

[14] D. S. Wilson, “Structured demes and trait-group variation,” *The American Naturalist*, vol. 113, no. 4, pp. 606-610, April 1979.

[15] M. Slatkin and D. S. Wilson, “Coevolution in structured demes,” *Proc. of the National Academy of Sciences*, vol. 76, no. 4, pp. 2084-2087, April 1979.

[16] B. Charlesworth, “A note on the evolution of altruism in structured demes,” *The American Naturalist*, vol. 113, no. 4, pp. 601-605, April 1979.

[17] M. Ifti, T. Killingback, and M. Doebelic, “Effects of neighbourhood size and connectivity on the spatial continuous prisoner’s dilemma,” *Journal of Theoretical Biology*, vol. 231, no. 1, pp. 97-106, November 2004.

[18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, “Implementation of cellular genetic algorithms with two neighborhood structures for single-objective and multi-objective optimization,” *Soft Computing*, vol. 15, no. 9, pp. 1749-1767, September 2011.

[19] 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.