Controllability of Population Characteristics of IPD Game Players based on the Spatial Structure

Masahiro Ono and Mitsuru Ishizuka
University of Tokyo,
Bunkyo-ku, Tokyo 113–8656, Japan
{mono,ishizuka}@mi.ci.i.u-tokyo.ac.jp

Abstract

In this paper, we deal with the influence of the spatial structure, which limits the communication of agents, on the characteristics of individual agents. We investigate the population characteristics and the behavior of the agents playing iterated prisoner's dilemma (IPD) games on the spatial structure. We first propose a new agent model that plays the IPD game, which contains the gene of the coded parameters of reinforcement learning. The agents evolve and learn while playing the games. Second, we report an empirical study. In our simulation, we observe that the spatial structure affects learning and evolution. Learning is generally not conducive to the mutual cooperation between agents, except in some special conditions. Then, we try to control the population characteristics. We find that they are controllable to some extent when we fix the strategies of several agents.

Introduction

The question "How do the environment and the society influence the individual agent and vice versa?" is a very interesting one, and many researchers have studied it in the past. Sociological studies explain the interaction between the environment and society in qualitative terms. Recently, we have been able to quantitatively investigate this interaction by using computer simulations.

Researchers in other fields have dealt with learning and evolution, and the interactions between the two. One of the famous earlier works is the Baldwin effect(Baldwin 1896)-the hypothesis that the characteristics of the way in which the individual learns affect the evolution of a species. Despite years of research, this problem remains unclear.

In this paper, the spatial structure among agents is treated as an environment. We investigate the influence of the spatial structure on the population characteristics and the behavior of the agents, namely, the way in which they evolve and learn.

Generally, the following aspects affect the actions of the autonomous agents: the spatial structure, the payoff structure, and the decision-making of the agent model.

The payoff structure includes two categories of games in game theory, the cooperative game and the noncooperative game. In the cooperative game, each agent cannot perform a different action after he or she has agreed to cooperate with other agents. The purpose of this game is to determine which agent should an agent cooperate with. The noncooperative game, on the other hand, is the game that inherently involves the absence of explicit rules. We adopt the iterated prisoner's dilemma (IPD), a noncooperative game, since it is suitable for primitive functions such as learning and evolution.

Thus far, researchers have studied the prisoner's dilemma (PD) because the game itself arouses our curiosity and is suited for the study of learning and evolution. Evolutionary game theory is a branch of game theory that is exclusively devoted to analyzing evolution. In general, there are studies on the equilibrium point of the game in the field of game theory. The evolutionary game theory led to the development of an alternative equilibrium point known as the evolutionary stable strategy (ESS), for which there exists both the numerical and the analytical approaches. Lindgren advanced one such numerical approach (Lindgren 1992). His model expresses a meta strategy that decides the next move according to the game history. The development of reinforcement learning has led to the study of games on learning. For example, studies show that the reinforcement learning agents in a multi-agent system create instability due to mutual learning (Sandholm & Crites 1996). There exist few studies that deal with both learning and evolution. Studies that are classified into this area combine learning and evolution and adapt the two functions to the stochastic game (Hingston & Kendall 2004), simple learning case in the investigation of the Baldwin effect (Suzuki & Arita 2004).

With respect to the original concern of IPD, the basic issue pertains to a contradiction between the mathematical solution, where the noncooperative strategy is stable, and the phenomenon in the real world, i.e., when people often cooperate each other. Researchers examined this issue for a long time and ultimately paved the way for the introduction of the spatial structure in the evolutionary game theory. Players are located on the spatial structure, for example, a two-dimensional regular lattice (Nowak & May 1992)(Lindgren & Nordahl 1994) or a small-world network (SWN) model (Watts 1999)(Masuda & Aihara 2003)(Ono & Ishizuka 2005). These players evolve in every generation after they play games across neighborhoods. In this case, we observe the emergence of a cooperative strategy after a certain period during which the noncooperative strategy is dominant. Thus far, there are no adequate investigations in studies on learning and evolution on the spatial structure.

In this paper, we investigate the influence of the spatial structure, which limits the communication of the agents, on the emergence of cooperation. In particular, we focus on the population characteristics and the behavior of the agents who evolve and learn. Next, we study the controllability of the population characteristics. We propose a simple method to fix the strategies of several agents and examine the controllability. using the simulation. In the remainder of this paper, we first introduce the prisoner's dilemma. We then describe a model that we designed for combining the functions of learning and evolution. We then discuss the experimental results. Finally, we present our conclusions and the directions for future study.

Prisoner's Dilemma

Since it is an elegant model for expressing various social phenomena, PD is the most popular game in game theory. Albert Tucker coined the name and developed the typical payoff matrix of this game in the 1950s.

Table 1 expresses the payoff matrix of PD in a symmetric two-player game, where R, T, S, and P represent reward, temptation, sucker, and punishment, respectively. Payoff relations ($T > R > P > S, 2R > T + S$) exist among the players, which leads to the dilemma.

When we assume each player to be rational, both players in the game would select the defect strategy. Player 1 considers that he should defect and earn a higher payoff irrespective of whether player 2 cooperates or defects. Player 2 would also defect after the same consideration. Ultimately, both the players defect, and (D,D) is the only Nash equilibrium in this game. However, this state is Pareto inferior in that it is not optimal for both the players. This is the reason why this game involves a dilemma.

Table 1: The payoff matrix of PD

		Player2	
		C (cooperate)	D (defect)
Player 1	C	R R	T S
	D	S T	P P

($T > R > P > S, 2R > T + S$)

IPD is a type of super-game-a game comprising several subgames. Two players play subgames repeatedly in a super-game. In this paper, we assumed that the number of subgames is fixed and the players are not aware of this number. Therefore, they cannot use backward induction.

Learning Agent Model

We propose a new agent model that learns and evolves for the IPD game. The agent is a player who plays the games. The agent selects the moves in the game and learns by the reinforcement learning mechanism. In addition, he or she has a gene of the parameters of reinforcement learning and evolves over generations.

Reinforcement Learning

Reinforcement learning belongs to a class of machine learning. Assuming that an agent takes an action and receives a reward from the environment in return, the reinforcement learning algorithm attempts to determine a policy for maximizing the agent's cumulative reward.

Basically, an agent has inner states. The learning process is as follows. The agent selects an action in a state according to each value function, evaluates the action in the state, and updates the value function. The agent carries out a state transition and selects the next action in the state. This process is repeated, and the agent continues to refine the action-value function. Although there are several algorithms for reinforcement learning, we take up the SARSA algorithm (Sutton & Barto 1998) in this paper.

Selection of the move

The process by which the agent selects his or her next move is as follows.

We assume that the agent recalls the moves of the previous game in an iterated game, and he or she selects the next move based on the first-order meta strategy. The agent has four inner states-CC, CD, DC, and DD-derived from the possible combinations of the moves in the previous game (his or her own previous move and the opponent's previous move). In other words, the agent selects the next move in a particular state based on his or her moves in the previous game.

Basically, when the agent selects a move in the game, he compares the action-values Q of each action in the states and chooses the bigger one, shown as Eqn. (1).

$$\text{next move} = \begin{cases} C, & Q(s,C) \geq Q(s,D) \\ D, & Q(s,C) < Q(s,D) \end{cases} \tag{1}$$

where $Q(s,a)$ denotes evaluation of action a in state s, and s is a possible states. For example, in state CC, the next move is C in the condition $Q(CC,C) \geq Q(CC,D)$. Since the agent has information about all the possible combinations of actions and states, we can express all the possible first-order meta strategies. Table 2 shows examples of this.

Table 2: Examples of the first-order meta strategy

PM	OPM	strategy examples			
		All C	All D	TFT	Pavlov
C	C	C	D	C	C
C	D	C	D	D	D
D	C	C	D	C	D
D	D	C	D	D	C

PM: previous move, OPM: opponent's previous move

The agent selects a move regardless of the previous moves in the case of all C's and D's. Tit-for-tat (TFT), a famous strategy created by Anatol Rapoport, involves repeating the opponent's previous move. This strategy is known as a winner of the famous tournament (Axelrod 1985). The Pavlov strategy (Nowak & Sigmund 1993)-also known as the "win-stay, lose-shift" strategy-involves selecting the move opposite to the previous move when the agent is unable to earn a high reward.

In return for his or her own action and the opponent's action, the agent receives a reward (the payoff of the game) and updates the action-values. The rule for updating the action-values is expressed in Eqn.(2). $Q(s_t, a_t)$ denotes evaluation of action a_t in the state s_t, and the next action-value is $Q(s_{t+1}, a_{t+1})$.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \tag{2}$$

where α is the learning rate, r_{t+1} is the reward in return for the action a_t, and γ is the discount rate. Here, a_{t+1} is decided by an ϵ-greedy method, in which the next move basically depends on the action-values, but is randomly selected according to a possibility ϵ. The agent adheres to the action-value completely, that is, he or she selects a move in a deterministic manner in the case of $\epsilon = 0$. Otherwise, the agent selects randomly and the selection is based on ϵ to some extent.

This updating process is repeated during each selection. This implies that the agent learns through the iterated games. Since the strategy depends on the action-values as explaines before, updating of the action-values implies a change in the strategy, e.g., from TFT to Pavlov. In other words, the agent has learned the game.

This model also takes into account the eligibility trace parameter λ.

The characteristics of the learning process depend on the reinforcement learning parameters. α expresses the speed at which the value function is changed. γ expresses the evaluation of the gain that the agent expects to received in the future. ϵ is a type of curiosity. $lambda$ expresses the extent to which the agent considers the future while updating the action-value.

Combination of learning and evolution

The agent has a gene that codes for two components. One is the part that expresses the initial action-values for each of the possible combinations of the actions and states. The other is the part expressing the four parameters of reinforcement learning, α, γ, λ, and ϵ.

Population characteristics

First, we investigate the population characteristics that are dependent on the spatial structure, particularly the learning tendency. The purpose of the experiment is to investigate the influence of the spatial structure on learning and evolution. We perform our simulation as follows:

1. Generate the population (population size) that comprises agents using the method defined by the spatial structure.
2. Execute the process given below g times:
   - i Play the super-games, which consist of (number of game repetitions) subgames, of all the agent pairs that are determined by the spatial structure.
   - ii Randomly kill (generation gap)% players in the population according to the gain earned by the agent who belongs to the generation after all the super-games are played.
   - iii Fill the vacancies in the population using the method defined by the spatial structure.

We assume that the random noise in the agent's selection of the move reverses according to the probability of the noise.

We carry out the experiment in two different spatial structure cases, i.e., a pool case and a network case. The processes 1, $2-$i, and $2-$iii are dependent on the spatial structure.

The pool case

We assume a pool of agents without a spatial structure. In this case, the agents are not linked with each other and they have an opportunity to meet any agent. First, we need for generate (population size) agents. Thus, we present the original process of each generation below.

2. i Let each agent play games (number of link) times with randomly selected agents. Because the spatial property lacks a structure, there is a possibility of them playing a game with any one of the agents. However, the number of super-games is limited in the process $2-$i in order to provide an opportunity for learning with an equivalent frequency for the network case.
   - iii Fill the vacancies in the population by the tournament selection (tournament size: 2), two-point crossover, and mutation

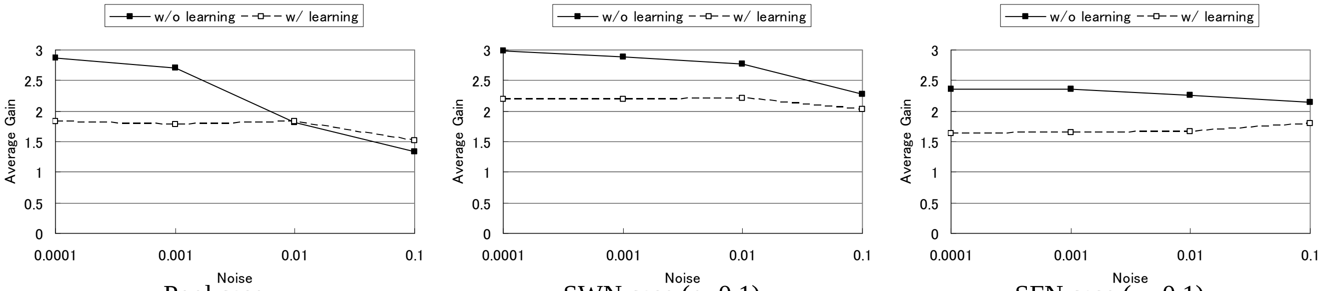
The network case

This is the case with a spatial structure that limits the communication among agents to some extent. We adopt two characteristic networks, namely, the small-world network (SWN) and scale-free network (SFN).

The network comprises (population size) agents, who have (number of links) links.

2. i Let each agent play games with other agents who have a direct connection on the network. The order in which the agents play games is randomly decided because a fixed order in each generation could an influence the learning.
   - iii Fill the vacancies in the population. Let new agents replace the dead agents; a new agent is generated as a copy of the agent with the highest earning in the first neighborhood of the dead agent. The mutation of the gene is generated with a probability of (mutation rate)% in copying.

The network structure is below 2.

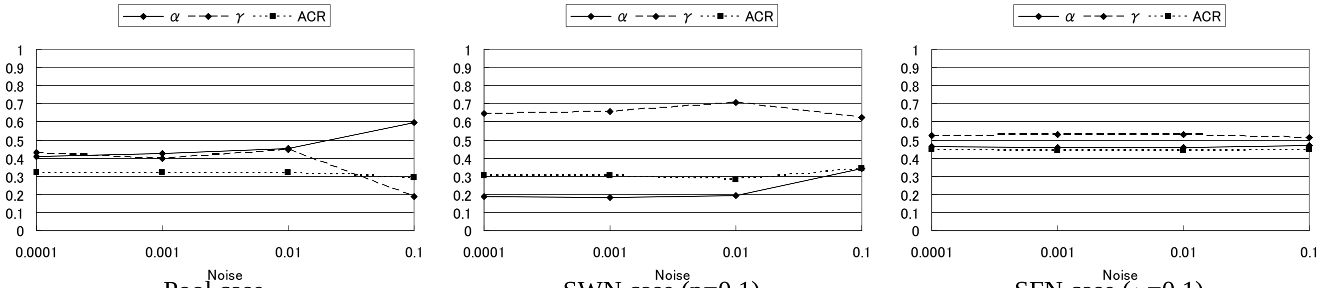


Pool case

SWN case (p=0.1)

SFN case (μ=0.1)

Figure 1: Average Gain



Pool case

SWN case (p=0.1)

SFN case (μ=0.1)

Figure 2: Reinforcement Learning Parameter

Small-world Network The SWN is defined using two characteristic parameters, namely, *characteristic path length* and *clustering coefficient* (Watts & Strogatz 1998). The *characteristic path length* L is the average of the length of the shortest path between any two vertices on the network. C indicates the extent to which the vertices adjacent to any vertex are adjacent to each other on an average.

When we assume the numbers of vertices and edges on a graph are fixed, the structure changes with variation in network's randomness parameter p. Every vertex is mutually connected to its neighborhoods in the case of $p = 0$. The edges changes stochastically as p increases. In the case of $p = 0$, it is a regular network, where both L and C are large. On the other hand, a random network appears at $p = 1$, where L and C are small. Midway between these two extremes, the network bears the property of a SWN, where L is small and C is large.

Scale-free Network The first SFN model was the BA model(Barabasi & Albert 1999) The SFN belongs to a class of graphs with a power law degree distribution.

In this paper, we adopt a model with the characteristics of both the scale-free distribution of degree and the small-world effect(Klemm & Eguíluz 2002a) in order to investigate the effect of the degree distribution of SFN in comparison with that of SWN. This model is a combination of the KE model (highly clustered scale-free networks)(Klemm & Eguíluz 2002b) and BA models (random scale-free networks)(Barabasi & Albert 1999) in the network growth process. The crossover parameter μ, which is the possibility of a crossover between the KE model and BA models, determines the C of the networks. Since μ is very similar to the randomness parameter p of the small-world network, these two parameters are similarly treated.

The randomness parameter p of SWN or the crossover parameter μ of SFN is given as an initial value and is fixed during the simulation.

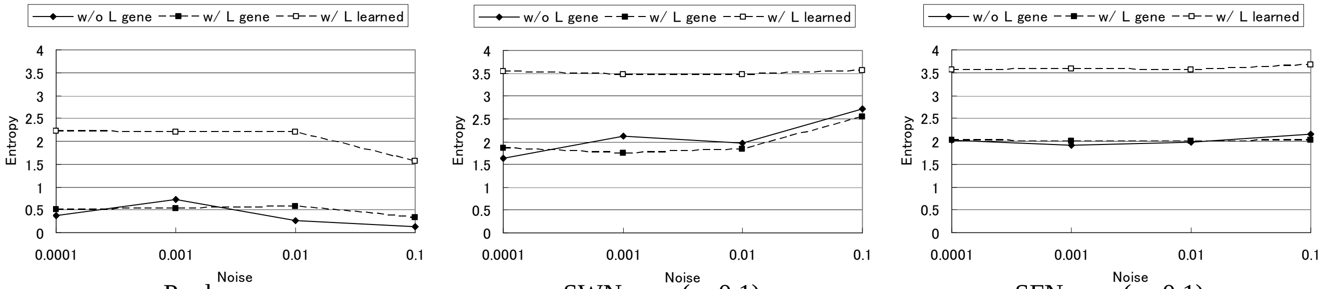
Table 3 presents the simulation parameters. As the first step, we fix two reinforcement learning parameters, ϵ and λ. The other parameters α and γ evolve freely.

Table 3: Simulation parameters

Parameters	Value
population	400
number of links	3
number of game repetition	100
generation gap	20%
mutation rate	0.02%
PD payoff parameters (T,R,P,S)	(5,3,1,0)
reinforcement learning parameters (ϵ,λ)	(0.1, 0)

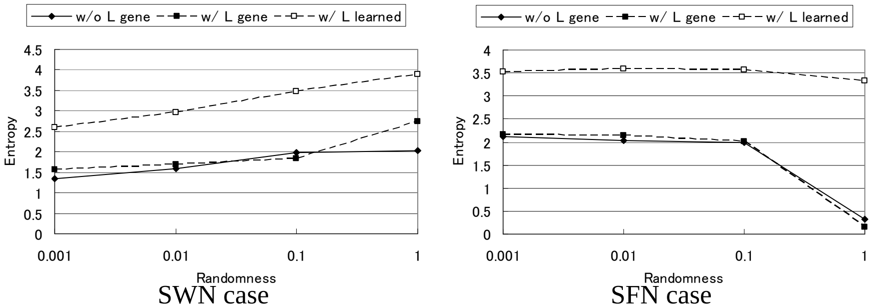
We conducted simulations by varying the parameters of the noise probability and the spatial structure. Further, the network parameter p or μ is variable in the network case.

Results First, we focus on the average gain, which is the index of how many cooperators exist in the population.



Pool case SWN case (p=0.1) SFN case (μ=0.1)

Figure 3: Entropy



SWN case SFN case

Figure 4: Entropy (noise=0.01)

When this index is near the value 3, it implies that there are many cooperators.

Figure 1 presents the average gain. The vertical axis is the average gain and the horizontal axis is the noise.

In the pool case, the difference between the learning and no-learning cases is clear in this figure. In the case with learning, the value remains almost constant irrespective of the noise probability. On the other hand, the value of the point of the case without learning is approximately 3 in the absence of the noise. This implies that almost all the agents cooperate with each other. The line decreases monotonically from the value 3 as the noise increases. We can observe that the lines cross approximately at the point 1.0×10^{-2}. This implies that the case in which agents cooperate well depends on the conditions, in this case, the noise probability.

With regard to the SWN and the SFN cases, it is obvious that all the points in the no-learning case are higher than those in the learning case. Both lines decrease monotonically as the noise increases. However, this tendency is higher in the no-learning case than in the learning case. In the figure, the line of the learning case is almost flat.

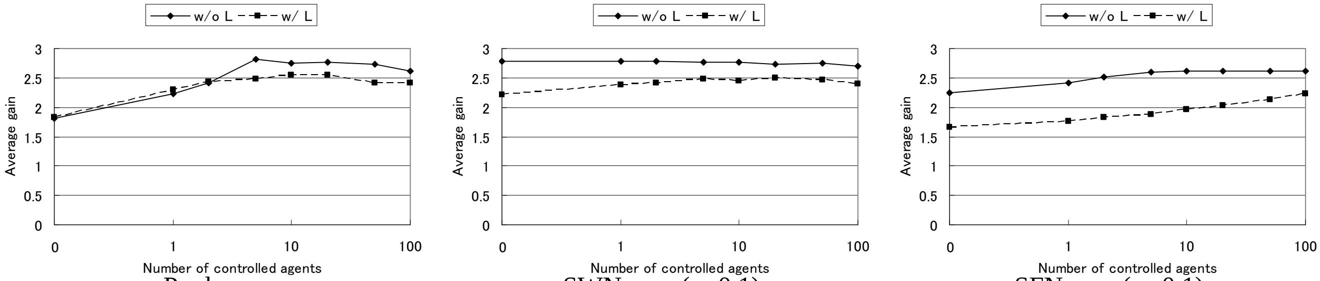
The fact that the no-learning case is better than the learning case except under a few special conditions implies that learning does not always satisfy the agents. In other words, the agents tend to become too greedy and select the Nash equilibrium in a traditional PD game.

We define an index called the action change rate (ACR) to investigate the effect of learning. Figure 2 depicts ACR and the reinforcement parameters, α and, γ. ACR is the rate of the action difference in each possible state between the strategy changed by learning and the initial strategy decided by the gene. α expresses the speed at which the value function is changed. γ expresses the evaluation of the gain that the agent expects receive in the future.

In all the cases, the lines depicting ACR remain almost flat at a certain value depending on the spatial structure. This implies that there is no inherent stable strategy. Learning always plays a role in adapting to the environment.

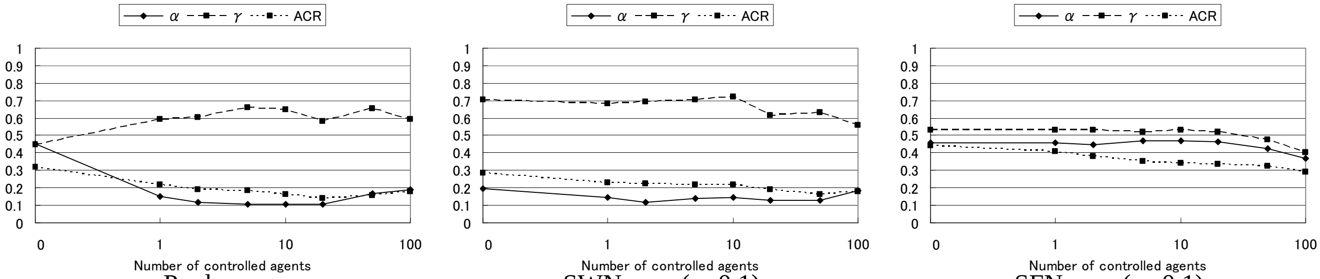
With regard to the variable reinforcement parameters of the agents, both α and γ are in the range of 0.2 and 0.5 in the pool case. On the other hand, in the SWN case, γ is considerably higher than α. In the SFN case, α and γ maintain almost the same value irrespective of the value of the noise.

These data imply that in the SWN case, the agents evaluate the gain that they expect to receive in the future as important and they will therefore not change their strategy immediately; they tend to maintain their relations with the other agents, and they can expect that there is high possibility that the opponent uses the same strategy. On the other hand, in the pool case, the agents evaluate the gain received at the moment as important, and they change their strategy immediately because they face unknown opponents in almost all the games. The SFN case is midway between the pool and the SWN cases. Although the network topology is fixed in both the SFN and the SWN cases, the agents would tend to be confused because they have to play games with agents who use different strategies, for example, a cooperative agent first and a noncooperative agent next.



Pool case SWN case (p=0.1) SFN case (μ=0.1)

Figure 5: Average Gain (noise=0.01)



Pool case SWN case (p=0.1) SFN case (μ=0.1)

Figure 6: Reinforcement Learning Parameter (noise=0.01)

At the point where noise=0.1, α increases and γ decreases in the pool case and the SWN cases. These reinforcement parameters depend on the spatial structure and the noise intensity. Generally, α is high and γ is low when it is difficult to forecast the future reward. An interesting point is that the learning process is differs considerably depending on the spatial structure and noise. However, the line in the SFN case is independent of noise. We suspect that the noise tolerance is high in the scale-free structure. Alternatively, encounters with different type of agents are dominant as compared to the noise.

Figure 3 shows the entropy of the population. In our simulation, there are 32 strategies, and therefore, the maximum value of the entropy is 5. In the figure, gene entropy refers to the inherent entropy of the strategies. Learned entropy refers to the entropy of the strategies that the agents learned after playing the games. ”L” represents learning. In all the cases, the learned entropy is considerably higher than the gene entropy. As the noise increases, the entropy increases to a certain extent in the pool case, decreases in the SWN case, and remains almost constant in the SFN case.

Figures 4 and 3 depict the entropy of the population. However, the horizontal axis represents the randomness of SWN or the crossover rate of SFN.

In the SWN case, all the entropies increase as the randomness of the network increases. On the other hand, all the entropies decrease in the SFN case; the gene entropy is particularly low at the point where noise = 0.1.

Controllability

In this section, we consider the controllability of the population.

One approach to do this is to change the rule of the game. For instance, we can change the design of the mechanism by monitoring and imposing sanctions. In this case, we assume a centralized organization that monitors games stochastically. If the organization recognizes that there is a certain probability that an agent defects in a game, it changes the payoff for the defector so as to punish his defection.

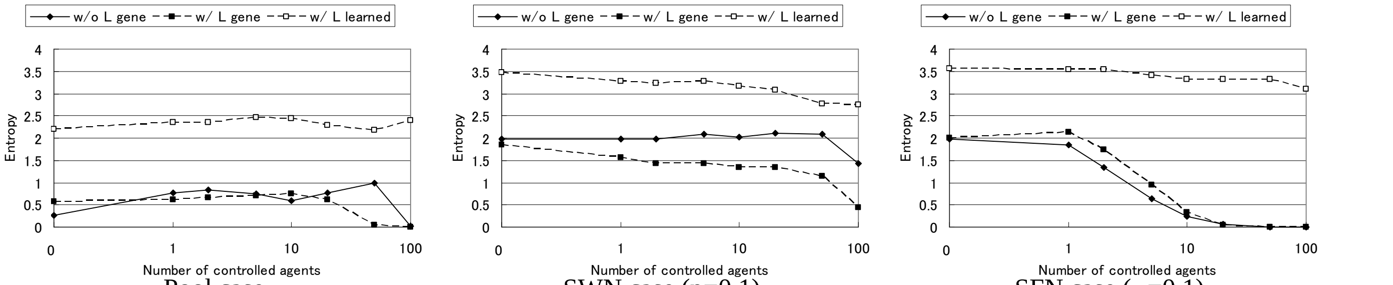
Another approach is the method that makes use of a positional bias in the network. We control several agents in a specific location in order to make the population characteristics desirable. The simplest way to select agents is to use the number of links an agent has. We select the top agents in the order of the number of links an agent has as controlled agents. We investigate this method as follows. TFT is fixed as the strategies that the the controlled agents use. These agents do not evolve and learn.

We perform the simulation described earlier, in the same manner except with respect to the controlled node. The number of the controlled node is the simulation parameter.

Result

Figure 5 depicts the average gain. The horizontal axis represents the number of controlled agents.

The pool case is the most efficient case for this method. In this case, the two lines rise rapidly as the number of controlled agents increase. Thereafter, the lines maintain certain



Pool case

SWN case (p=0.1)

SFN case (μ=0.1)

Figure 7: Entropy (noise=0.01)

values, and the line of the no-learning case is higher than that of the learning case.

In the SWN case, we observe a slight improvement. Irrespective of the controlled agent, the average gain is almost constant.

With regard to the SFN case, the tendency of the lines in the learning case is gentler than that of the lines in the no-learning case. However, the line of the no-learning case increases linearly in log scale even though the other line saturates at the point of less-controlled agents.

Figure 6 presents the method of learning of the agents for the pool case. Only one controlled agent dramatically changes the characteristics of the other members in the population. The fact that the probability of access to the other members is not zero achieves a wide propagation in the population. The controlled agent uses the TFT strategy; therefore, a noncooperative agent would tend to receive punishment, and he or she would have a high probability of dying. Basically, cooperative agents expect rewards in the distant future and hence select cooperative moves. This figure shows the process described above.

Figure 7 depicts the entropy of the strategies. It is clear that the gene entropy decreases dramatically in the SFN structure. The noise is the answer to the obvious question of why the average gain is low despite the fact that TFT is the major strategy. It is observed that the average gain improves to nearly 3 in the case of noise = 0.

Discussion

In this section, we consider administrating the population characteristics.

If the method by which agents make decisions is unknown, the SWN network is a desirable spatial structure because it has a higher average gain in the two learning cases. However, typically, many networks have a scale-free tendency (Barabasi & Albert 1999).

When we assume the network to have a scale-free structure, how can we improve population characteristics? The answer to this question depends on the extent of the manipulation, that is, how freely we are able to control the population. If the network is completely controllable, we can achieve a desirable network. In the case that permits limited manipulation, at least three methods exist.

One method is to reduce the noise. However, it is not very effective in SFN since the noise is not critical in this network structure.

The second method involves making an announcement to the agents, e.g., a recommendation to access the agents directly connected to the first-order neighborhoods. This method induces spontaneous network dynamics by each agent.

The last method is to locate a controlled agent in the biased position. This method involves some difficulties; however, negotiation with the agent who already exists in the biased position is much easier than controlling all the agents. Furthermore, the administrative agents who exist in the initial network are effective in the growing network model.

We should consider the strategy that the controlled agents use. In the SFN case, the reason why the average gain is low even though the entropy is also low is because TFT does not perform well when noise = 0.01. The Pavlov strategy might be better than TFT in this case.

In the SFN case, the dependence on the crossover rate influences the entropy of strategies. The decrease in diversity in response to the increase in randomness could be another problem apart from the average gain problem; this creates a need for us to discuss this in concrete cases.

Conclusion

We studied PD game on the spatial structure played by agents who learn and evolve.

We proposed the agent model with learning and the evolution of strategy. The results of our experiments show that the spatial structure of the agents influences the cooperation between agents and the learning and evolution. As compared to the learning case, agents in the no-learning case perform well except in some special conditions that contradict our intuitions. There are no inherent stable strategies, and learning always plays a role in adapting to the environment. Furthermore, we proposed a simple method to control the population characteristics. The simulation results explain the controllability.

Further study should consider the dynamic network case. In addition, further research should attempt to determine the most efficient strategy that the controlled agents use.

References

Axelrod, R. 1985. *The Evolution of Cooperation*. Basic Books.

Baldwin, J. M. 1896. A new factor in evolution. *American Naturalist* 30:441–451.

Barabasi, A., and Albert, R. 1999. Emergence of scaling in random networks. *Science* 286(5439):509–512.

Hingston, P., and Kendall, G. 2004. Learning versus evolution in iterated prisoner's dilemma. In *Proceedings of the Congress on Evolutionary Computation, CEC2004*, volume 1, 364–372.

Klemm, K., and Eguíluz, V. M. 2002a. Growing scale-free networks with small-world behavior. *Physical Review E* 65(5):057102–1–057102–4.

Klemm, K., and Eguíluz, V. M. 2002b. Highly clustered scale-free networks. *Physical Review E* 65(3):036123–1–036123–5.

Lindgren, K., and Nordahl, M. G. 1994. Evolutionary dynamics of spatial games. *Physica D* 75:292–309.

Lindgren, K. 1992. Evolutionary phenomena in simple dynamics. *Artificial Life II: Proceedings of the Workshop on Artificial Life held February 1990 in Santa Fe, New Mexico* 295–312.

Masuda, N., and Aihara, K. 2003. Spatial prisoner's dilemma optimally played in small-world networks. *Physics Letters A* 313:55–61.

Nowak, M. A., and May, R. M. 1992. Evolutionary games and spatial chaos. *Nature* 359:826–829.

Nowak, M. A., and Sigmund, K. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature* 364:56–58.

Ono, M., and Ishizuka, M. 2005. Prisoner's dilemma game on network. In *Proceedings of the Eighth Pacific-Rim International Workshop on Multi-Agents, PRIMA2005*, 9–22.

Sandholm, T. W., and Crites, R. H. 1996. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* 37(147–166).

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning*. MIT Press.

Suzuki, R., and Arita, T. 2004. Interactions between learning and evolution: Outstanding strategy generated by the baldwin effect. *Biosystems* 77(1-3):57–71.

Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of 'small-world' networks. *Nature* 393:440–442.

Watts, D. J. 1999. *Small Worlds*. Princeton University Press.