Ecology of Evolutionary Game Strategies

Takashi Ikegami *

The Graduate School of Science and Technology

Kobe University, Rokkodai, Nada-ku, Kobe 657, JAPAN

Abstract

A model of the noisy Iterated Prisoner's Dilemma game with evolving strategies is presented. It is called a noisy game as a strategy sometimes makes an error by noise. A strategy to be discussed is a memory strategy, using the previous moves to make a choice in the next move. Each strategy is represented by a binary tree and a new strategy is synthesized through the *Genetic Fusion* [Ikegami and Kaneko, 1990]. A pair of strategies generates a new one by linking a root of one tree to a leaf of the other.

In each generation, strategies are paired with each other in a round robin tournament. Each strategy can increase its population size proportional to its score. After several generations, a population will consist of relatively successful strategies. In this paper, when and how long memory strategies can become dominant in a population will be reported.

In a low noise regime, long memory strategies can emerge through the duplicated fusions of contradictious strategies. In the population of kin strategies, their scores bifurcate into high and low values. This bifurcation makes the long memory sustainable in a population. In a high noise regime, the effectiveness of the score bifurcation is lost and long memory merely emerges. One of the best strategy but of a short memory, Tit for Tat, may only be advantageous in the noiseless or very high noise regime.

1 Introduction

A game theory can analayze strategic interactions often seen in economic and biological systems. There is no energy function to minimize in general. There exists no absolute fitness landscape to optimize. Reward to an individual strongly depends on the other strategies in the current population.

If every possible strategy is known in advance, a game theory can suggest which strategy is rational to play. Many solution concepts have been proposed to classify equilibrium states of games with rational players with a finite number of strategies.

In a real situation, however, each player often invents new strategies rather than selecting from a given set of strategies. The states of games are far from equilibrium, in general.

*E-mail address : ikegami@gradis.scitec.kobe-u.ac.jp

Solution concepts in a game theory have little to do with this non-equilibrium situation. For example, an evolutionary stable strategy(ESS) is believed to build up a stable equilibrium state. It is shown, however, that ESS may be dynamically inaccessible [Nowak 1990].

Here we present an evolutionary dynamics that creates new strategies brought from our Genetic fusion algorithm [Ikegami and Kaneko, 1990]. Two genes represented by bit strings are combined to generate a longer gene by a fusion process. It is a powerful evolutionary source of genetic systems.

In the present paper, we apply the genetic fusion method to the iterated prisoner's dilemma (IPD) game: This game has been substantially studied as the simplest strategic game. In the IPD game, individuals play the game in pairs in the given population. During an each match, a player chooses the move 'Cooperate'(C) or 'Defect'(D) without knowing the opponent's next move.

		Player 2	
		C	D
Player 1	C	(R,R)	(S,T)
	D	(T,S)	(P,P)

Table I. The payoff matrix for the prisoner's Dilemma game. In each element, (S_1, S_2) corresponds to the score of player 1 and 2, respectively. Following Axelrod's tournament, we use R=3, S=0, T=5 and P=1.

The payoffs to players with respect to their moves are depicted in Table I. Those payoff values satisfy the relations $T > R > P > S$ and $R > (T+S)/2$, so that playing D dominates playing C if the game is played only once. For the long iteration of the game, always playing D is no more good strategy as well as always playing C. A player should develop sophisticated strategy to get a high score. Reciprocity is believed as a necessary ingredient in a strategy to get high scores. One such strategy with a minimum reciprocity is named "Tit for Tat(TfT)", which starts with cooperation and immediately defects against defection. TfT also immediately cooperates against cooperation. Although simple it may look, TfT restrains the damage to minimal. It develops a mutual cooperative behavior, getting high scores. Particularly since the Axelrod's computer tournament, various strategies have been proposed as the best strategies [Axelrod 1984, Donniger 1986]. Most of them are memory strategies which determine their next move from previous moves.

In addition to the original Axelrod's rule, a probability (ϵ) to make an error in chosing the next move is concerned here. A player who wants to play C according to the strategy sometimes plays D with the probability ϵ. It has been shown that in such a noisy game, TfT is replaced by more generous strategies against defection [Molander 1985, Bendor 1987]. When it is difficult to distinguish between intended and unintended defection, an unconditional generosity results in a high score in the low error regime. In the high error regime, however, there is a trade-off between generosity and exploitation. Reciprocal strategies with a longer memory will always outperform TfT but may be eliminated by sophisticated defectors [Bendor 1987].

Within an evolutionary dynamics, long memories will be gradually acquired from initial short memories. Hence the meaning of long memory in strategies can be explained from the evolutionary point of view [Lindgren 1991].

2 Model

2.1 Time evolution

Time evolution of strategies is described by a set of dynamical equations[Hofbauer and Sigmun Each game consists of 100 moves per each time step:

$$y_i' - y_i = \alpha y_i (\sum_{j=1}^{n(t)} g_{i,j} y_j - \sum_{i=1}^{n(t)} \sum_{j=1}^{n(t)} g_{i,j} y_i y_j), \tag{1}$$

where y_i and y_i' represent the frequency of the strategy i in the population of time step t and t+1, respectively. The total sum of whole frequency is normalized to unity($\sum_{i=1}^{n} y_i = 1$). A reward to the strategy i against the strategy j is denoted by $g_{i,j}$, which is divided by the game length 100. A frequency of strategy increases in proportion to the difference of the acquired score($\sum_{j=1}^{n(t)} g_{i,j} y_j$) and the average score in the population($\sum_{i=1}^{n(t)} \sum_{j=1}^{n(t)} g_{i,j} y_i y_j$).

It should be noted that the number of degrees of freedom(e.g. the number of strategies) $n(t)$ also evolves with time. The strategy whose frequency becomes lower than the given threshold N^{-1}(fixed to 1 percent)will be removed from the population. On the other hand, a banch of new strategies is brought into the population by a genetic fusion.

2.2 Tree coding

Each strategy determines the next move by comparing the previous moves with its *memory list*. The memory list holds a set of patterns to cooperate with, which is encoded on a binary tree(Fig.1). A node is called C(D) node if it is a upper(lower) branching node with respect to its parent's node.

A root of a tree corresponds to the current state. The depth of (2M-1)th node corresponds to the opponent's move and that of (2M)th node to its own move at the M steps before. An initial move of the strategy is set besides the tree. This tree structure and initial move characterize behaviors of the strategy. The next choice is made by the following procedures:

(i) The sequence of previous moves is coded on a string $P_M = (\sigma_1, \sigma_2, \cdots, \sigma_{2M-1}, \sigma_{2M})$, where $\sigma_i = C$ or D. As well as the tree structure, the opponent's move and own move at the k steps before are expressed by σ_{2k-1} and σ_{2k} of the string, respectively.

The length of the string P_M is increased by two bits per each iteration of the game(e.g. $P_2 \rightarrow P_4 \rightarrow P_6 \rightarrow \cdot\cdot$).

(ii)Every possible paths from a root to leaves of a tree is compared with the string P_M. The depth of k-th node is refered to the h-th site of the string P_M. The shorter length of the two has an advantage.

(iii) It will chose C if there occurs a full coincidence, otherwise play D.

The above procedures are repeated over each move of the game.

2.3 Fusion and Mutation

We assume a hypothetical population size to manage an evolution algorithm. The population size of each strategy is obtained by multiplying each frequency y_i by the size N,

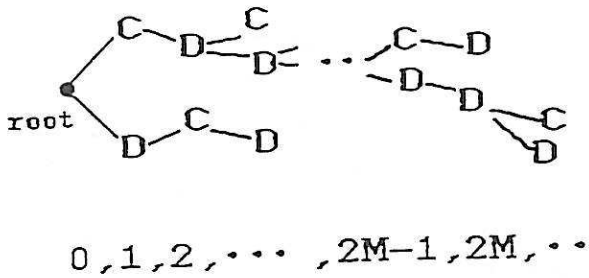

Figure 1: A binary tree structure which encodes a strategy. The tree has C or D as its nodes. Counting from a root of the tree, the depth of (2M-1)th node corresponds to the opponent's move and that of (2M)th node to own move at the M steps before. If a past move pattern is found in the tree, a strategy plays C and otherwise plays D.

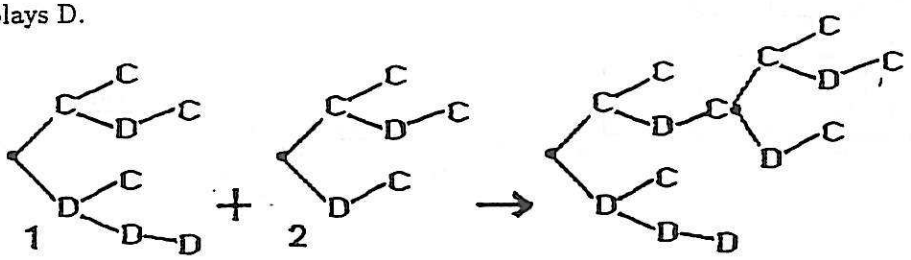

Figure 2: A strategy 1 and 2 synthesize a new tree by linking a leaf of a strategy 1 with a root of a strategy 2.

which is fixed at 100 through our simulation. There will be at most N mutation events and N/2 fusion events per each time step.

i) Mutation: For each individual strategy, a state of a randomly selected node of each path from a root to a leaf is reversed (C→D: D→C) with a probability μ.

ii)Fusion: A pair of trees generates a new tree by linking a root of one tree to a leaf of the other with a fixed probability φ. The generated tree always gets longer than the parent strategies(Fig.2).

Since the population size is decreased by the fusion events, the frequency of strategies should be renormalized per each time step.

There exist two advantages of this coding and the fusion algorithm. One is that a tree only encodes the important part of the memory, saving much memory space. Also a "don't care" symbol, which effectively functions in a classifier system, can easily be implemented in a tree structure. The other one is that a strategy can mix up different memory patterns by fusion. A fusion process is like a virus infection, bringing along a useful or useless memory to the infected host.

3 Simulations

Irrespective of the initial conditions, a population will soon be dominated by one or two major strategies. The state with one dominant strategy may appear transient, and will be taken over by newly evolved strategies. Newly evolved strategies are mutated from the latest dominant strategy or synthesized through a fusion process. After several punctuation of quasi-equilibrium states, it arrives at a stable equilibrium state where no new strategy can merely emerge. This game possess many such stable states with one dominant strategy.

Memory length of strategies is often getting long, however, cooperative behavior does not always follow. The emergence of cooperative behavior depends on the amount of error and

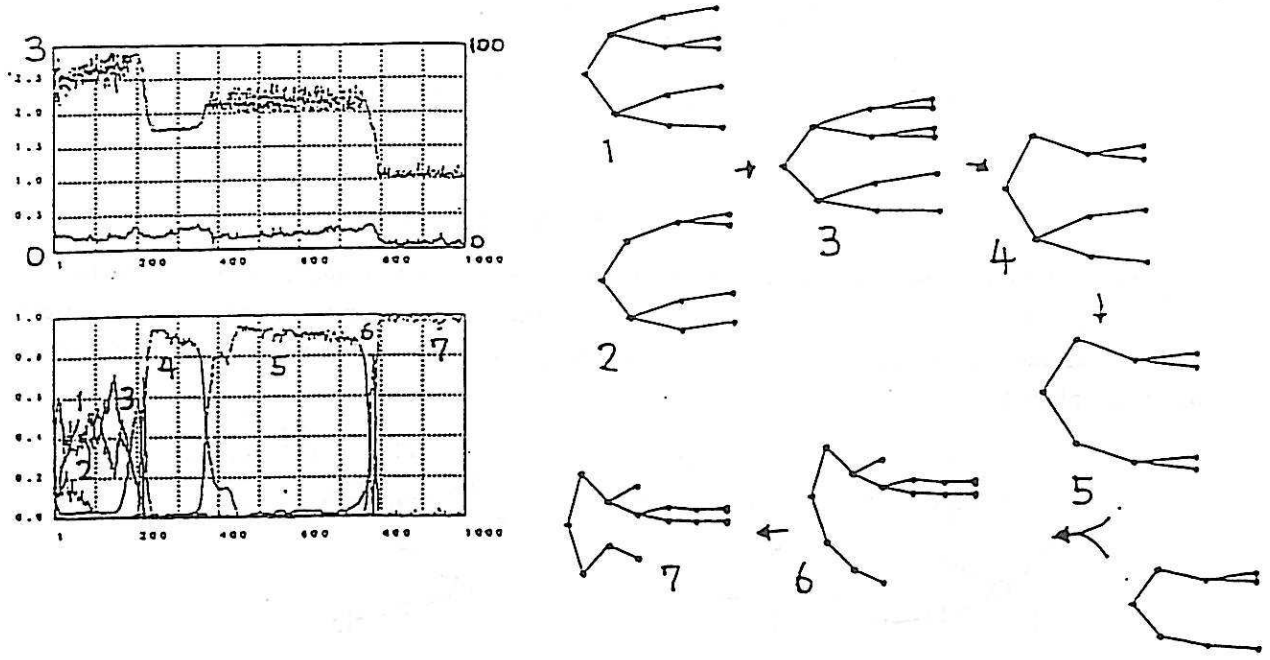

Figure 3: Example of evolution from initial strategies with a memory length 3. Parameters are set at, $\alpha = 0.5, \epsilon = 0.01, \mu = 0.0005$ and $\varphi = 0.003$. The upper left figure shows a time evolution of averaged score and that of the number of different strategies(a lower line) in a population. The lower left figure shows a time evolution of frequency of each strategy. The right figure depicts the tree representation of strategies. At each node, a lower branch corresponds to D and an upper to C.

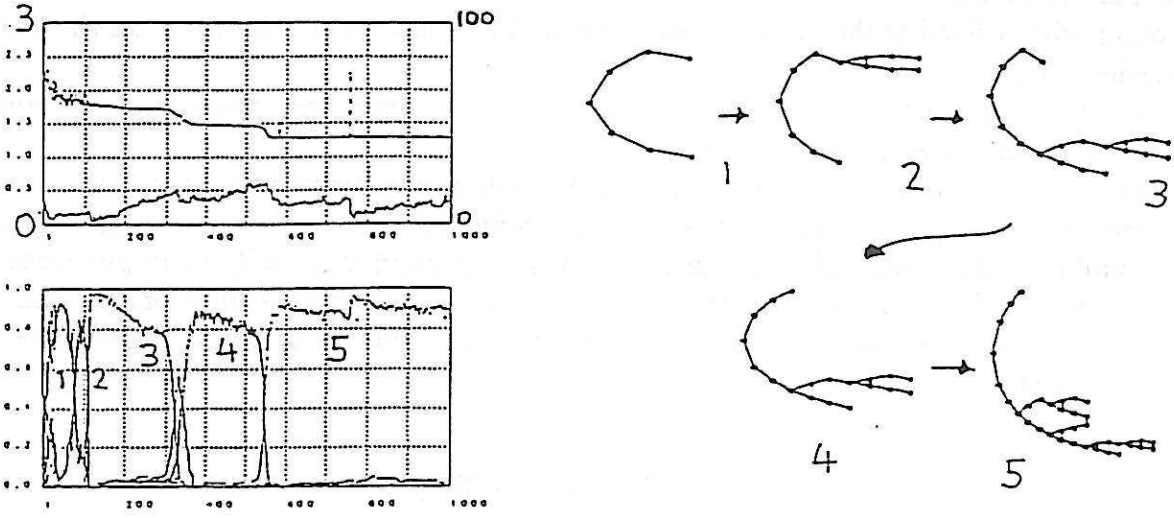

Figure 4: Parameters are set at, $\alpha = 0.2, \epsilon = 0.01, \mu = 0.0005$ and $\varphi = 0.003$. Initial strategy 1 duplicates three times to have a strategy 3. The strategy 3 mutates once and duplicates to have the strategy 5 with a memory length 12.

the structure of initial population. It is often observed that generosity(playing C against opponent's playing D) is gradually decreased through the evolution. But sometimes a higher degree of generosity is sustained in a stable equilibrium state.

In figures 3 and 4, examples of evolutions of strategies are depicted.

In figure 3, the initial population is dominated by the coexistence of two distinct strategies with memory length 3. Each strategy fails to behave cooperatively against itself. Instead, they behave cooperatively against each other. If one strategy misplaces D in place of C, the other strategy manages to recover mutual cooperation. Due to this mutually repairing mechanism, both strategies get high scores. In this sense, this pair strategy is called a symbiotic pair. This kind of symbiotic pair is also reported by Lindgren [Lindgren 1991] in the same context.

The strategy 5 fuses with its rare mutant strategy 5', generating the strategy 6 with increasing a memory length to 6. The strategy 6 is immediately exploited by its mutant strategy 7. The strategy 7 has a memory length 6 but it mostly behaves as All-D. No further strategy evolves from the strategy 7.

In figure 4, the initial dominant strategy evolves through self-duplication. The degree of generosity gradually decrease with respect to its memory length.

Those two examples end up with less cooperative strategies. But this is not always the case. The cooperativeness of strategies depends on the error rate. We next study the evolution from a particular initial set of strategies of memory length 2 with TfT.

When there exists no noise, only TfT and its mutant strategy dominate the population. Those mutant strategies often possess long memories , however, they effectively play TfT. The long memory seems redundant in the noiseless game. For any positive ϵ, however, various strategies besides TfT will evolve. For the relatively low ϵ regime, roughly three patterns emerge as a final strategy(Fig.5). Each type is attained though a bifurcation of a common strategy F which plays D for the past moves of $[D, C, \cdots]$ and plays C for otherwise. The $\hat{A}$ strategy appears as the duplicated fusion of the mutant strategy, which plays C only for the past moves of $[C, C, \cdots]$ and of $[D, D, \cdots]$. The $\hat{A}$ strategy well sustains mutual cooperation. When played against itself, it will defect twice and return to the mutual cooperation for the erroneous defection.

On the other hand, the $\hat{B}$ and $\hat{C}$ strategy are not mutually cooperative. A $\hat{B}$ strategy can have several memory lengthes such as 2,4,8,16,·· depends on the error amount. The $\hat{B}$ strategy with the memory length 2 is known as a contradictious strategy , which plays opposite to the opponent's last move. The $\hat{B}$ strategies with longer memory are obtained from the duplicated fusions of that contradictious strategy. When played against itself, the

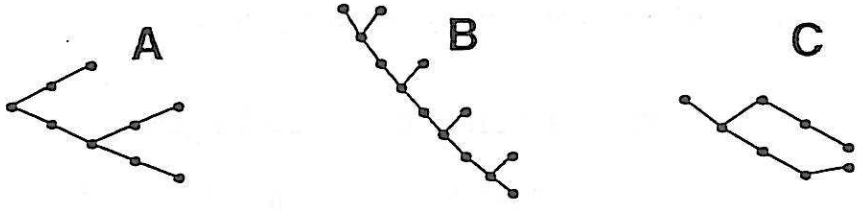

Figure 5: A particular initial state with memory length 2 leads to roughly three types of strategies for the low error regime. Each strategy has D as an initial move.

$\hat{B}$ strategy continues to defect until each of the other cooperates by mistake. Those who failed to play D has to play C and the other to play D all the way. As the result, one playing D gets the highest score 5 and the other gets the lowest 0. In the other words, the players adopting the same strategy become rich or poor by mistake. Once this unequal relationship is established, it is sustained unless the rich player fails to defect against the poor player. The relationship cannot be resolved from the poor player side. Within this unequal relationship, the $\hat{B}$ strategy gets scores of 2.5 as an average.

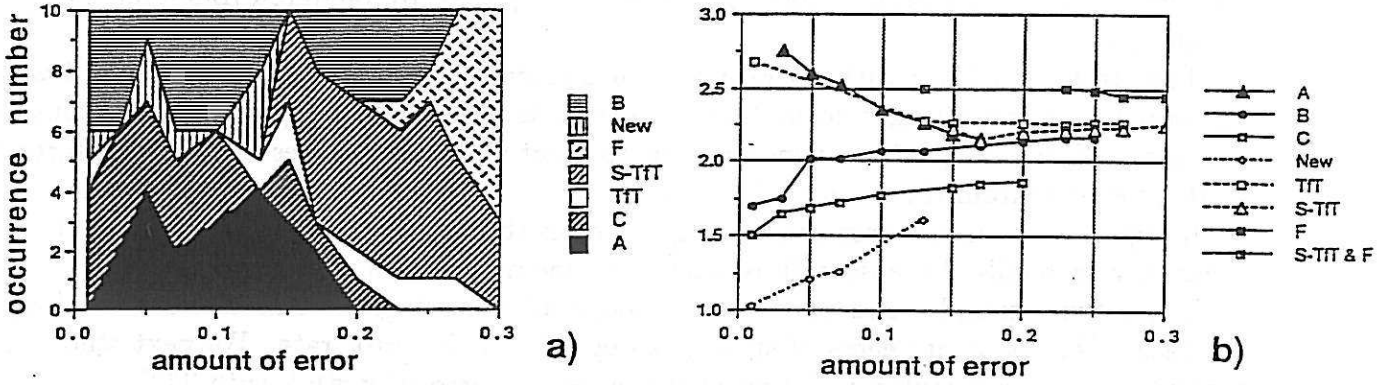

Figure 6: The occurrence of finally dominated strategies after 2000 time steps from the initial set of memory length 2. It is computed over 10 times for each value of ϵ with $\alpha = 0.2, \mu = 0.0003$ and $\varphi = 0.003$. Tit for Tat only dominates the population for the very low or very high error regime. (b) The averaged score is depicted with respect to error rate. An asterisk is attained by the coexistence of two strategies, the strategy F and S-TfT.

The strategy $\hat{B}^k$ (a memory length of slK) must ccoperate after K times mutual defection. Hence it plays C against defection of the kin $\hat{B}$ strategies with longer memory($> k$).

For the relevant fusion rates, the $\hat{B}$ strategies continuously dominate the population by enlarging the memory length.

The $\hat{C}$ strategy is a less exploitable strategy than $\hat{B}$. It also brings about a score (rich/poor) bifurcation, however, cannot enlarge its memory length beyond 8. Since the duplicated strategies decrease their generosity, leading to lower scores.

For the high amount of ϵ, the effect of score (rich/poor) bifurcation is getting lost because of the high rate of mistake. Evolutions cannot go beyond the simple contradictious strategy or the strategy of type F. TfT(and S-TfT) will again become one of the final strategies. The advantage of long memory will be wiped out by increasing the ϵ.

4 Complexity of Strategies

In the previous section, we have seen that a memory length of strategy often evolves in time. The evolution of memory is not a simple one-dimensional growth but a two-dimensional tree-like growth.

By taking an algorithmic point of view, we can characterize memory content of strategies apart from its memory length. If behavior of a strategy is simulated with a shorter program,

the strategy is said to have a lower algorithmic complexity. On the other hand, the strategy with a high algorithmic complexity needs a longer program to run. In our model, a strategy with nodes which have isomorphic subtrees should be less complex than those who have only nonisomorphic subtrees. Since a tree with more nonisomorphic subtrees has to look up further previous moves. We define such strategy as an algorithmically more complex strategy.

This algorithmic complexity [Huberman and Hogg, 1986, Bachas and Wolff, 1987] can be calculated by integrating the number of nonisomorphic subtrees in each node of strategy. Two trees are defined as nonisomorphic trees when they have at least one different branching.

From a root to leaf of a tree, we recursively compute the number of nonisomorphic branching at each node. Consider a subtree (T_n) which begins at a given node. This subtree consists of at most two subtrees T^1_{n-1} and T^2_{n-1} which start at the node of the next level. The measure for the diversity of a tree ($\tilde{D}(T)$) is computed as;

$$\tilde{D}(T_n) = B\sum_{i=1}^{b}\tilde{D}(T^i_{n-1}) + \delta_{b,1}, \tag{2}$$

The parameter b denotes the number of branching at the node. If the node is a leaf, b takes 0 (e.g. $\tilde{D}(T) = 0$). If a node has one descendant, b takes 1. The maximum value of b is 2. If a node has two subtree and the two subtrees(T^1_{n-1} and T^2_{n-1}) have an isomorphic structure, the coeficient B takes 1/2. Othewise it takes 2. If a node has one subtree(b=1), the coeficient B takes 1. For the tree where every node has two nonisomorphic subtrees, the measure $\tilde{D}(T)$ is given by 2^N, where N is a total number of nodes.

By taking a logarithm of $\tilde{D}(T)$ of a base 2, we define a complexity of tree:

$$\tilde{C}(T_n) = \log_2(\tilde{D}(T_n) + 1). \tag{3}$$

By this definition, ALL-C strategies have zero complexity as well as ALL-D strategy. On the other hand, the complexity of $\tilde{B}$ strategy with memory length k is computed as $k/2$.

In general, $\tilde{C}(T)$ roughly gives the number of nonisomorphic branching in a tree T. A complete isomorphic tree (ALL-C) or a simple root (ALL-D) give a minimum complexity zero. The highest complexity appears in a tree with many nonisomorphic subtrees.

By using the above definitions, we compute the average value of $\tilde{C}(T)$ in the population. Two examples are depicted in figure 7.

Figure 7-a) corresponds to the evolution in figure 3. The value of $\tilde{C}(T)$suddenly drops at time=400 and again rise to $\tilde{C}(T)$=3.4 at time=800. During the interval ($400 < time < 800$), a strategy whose root has two isomorphic subtrees dominates the population(see Fig.3). Hence its complexity decreases to around 1.2. At time=800, the complexity of 3.5 is acquired by the fusion event.

On the othe hand, the complexity increases in a stepwise way in figure 7-b). The evolution starts with the F strategy (which only play D for $[D, C, \cdots]$), which is soon overtaken by the B^2 strategy (e.g. a contradictious strategy of a memory length 2). The B^2 strategy evolves by a duplicated fusion into B^4. This self fusion process generates kin strategiesa(B^8, B^{16}, B^{32}....). As is stated, their complexity values $\tilde{C}(T)$ are given by 4,8 and 16, respectively.

It seems that the average score decreases (increases) when $\tilde{C}(T)$ increases (decreases) in figure 7-a). In general, strategies with higher complexity gain lower scores against themselves.

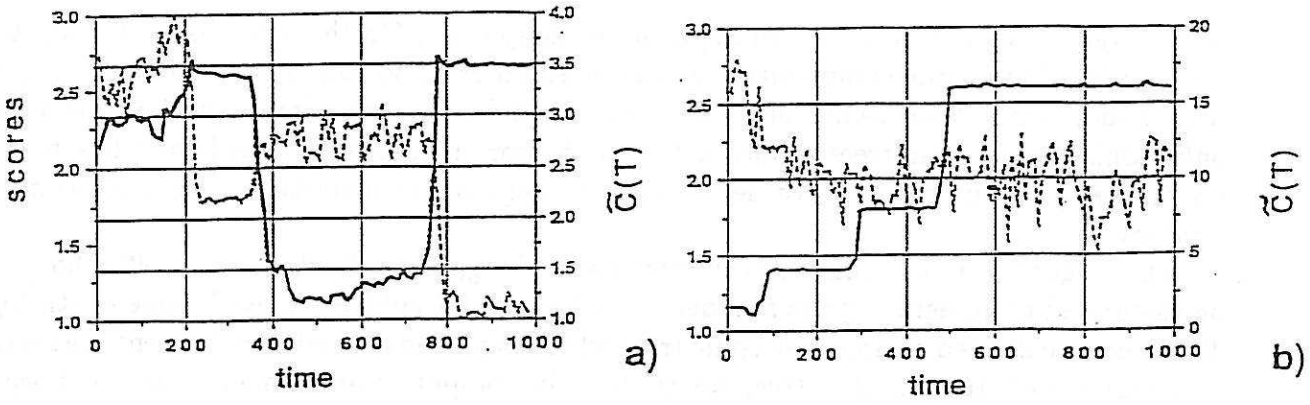

Figure 7: Time evolution of averaged $\bar{C}(T)$ (a solid line) and scores (a dashed line) in the population.
a)The same parameters and the initial condition are set as same as in figure 3.
b)Parameters are set at, $\alpha = 0.5, \epsilon = 0.07, \mu = 0.0001$ and $\varphi = 0.005$. Initial strategy is a F strategy.

One of the reasons for this is that strategies with higher complexity $\bar{C}(T)$ often show less generous behaviors. Namely, they play less cooperatively. Hence they easily fail to cooperate. Our preliminary results of noisy IPD game between randomly generated strategies shows that the strategy with higher $\bar{C}(T)$ have twice as many chances as to overperform the lower $\bar{C}(T)$ strategies.

5 Discussions

In the noiseless IPD game, TfT does not lose seriously against almost any other strategies. Such robustness of a strategy is sometimes more important than ESS concept for an evolutionary dynamics. Since a strategy does not meet with every possible strategy in its evolutionary way.

The $\hat{A}$ strategy found in our simulation serves as a robust strategy in the noisy game. This strategy has a rather short memory length 4. A strategy to be robust in the noiseless game should be nice, provocative and forgiving strategy [Axelrod 1984]. In the noisy game, fault-tolerancy may be the forth property to be a robust strategy. A fault-tolerancy is a repairing mechanism which recovers cooperative behavior from erroneous defection. The strategy 1 and 2 with memory length 3 in figure 3 also acquire this mechanism. The $\hat{A}$ strategy has obtained the fault-tolerancy by the memory length 4.

On the other hand, a longer memory seems to be a counter property to robustness. What then makes a memory length grow? From the present result, a long memory is only advantageous at the specificity of the population. A long memory is used to generate a worker-parasite relationship. One continues to cooperate (worker) and the other to defect (parasite) all the way. If the population is constantly supplied with randomly generated strategies, no long memory would be developed. The $\hat{A}$ strategy is a non specific robust strategy. On the other hand, the strategies $\hat{B}$ and $\hat{C}$ can only get high scores in the population of its related strategies. A long memory is only established within the latter type of strategies.

It reminds us of a runaway evolution in a biological arm race. Hosts and parasites coevolve in opposition to each other's new strategy. As the result, both species become too specific, inviting extinction. A strategy with a specified long memory also have this fear of extinction.

In our simulation, synthesized strategy with a long memory often appear as a homologous polymer of the same strategy (e.g. duplication process). Only a few of them appear as a heterogeneous polymer. This is because a population is mostly dominated by one type of strategy, so that a homologous rather than heterogeneous fusion easily takes place. By considering a spatially extended system [Matsuo and Adachi, 1987] or a weakly chaotic system, diversity of strategies may be sustained [Kaneko and Ikegami 1992] in a population. For such systems with full of diversity, a heterogeneous fusion will be essential for the innovation of strategies.

Acknowledgments

The present research is partially supported by Grandt-in-Aids (No. 04854025) for Scientific Research from the Ministry of Education, Science and Culture of Japan.

References

[1] T.Ikegami and K.Kaneko, Phys. Rev. Lett. 65 (1990) 3352.

[2] M.Nowak, J.theor. Biol. 142 (1990) 237.

[3] See, e.g., R.Axelrod, *The Evolution of Cooperation* (Basic Books, NewYork 1984).

[4] C.Donniger, in *Paradoxical Effects of Social Behavior*, (eds. A.Dickmann and P.Mitter, Physica-Veralg, Heidelberg, 1986)123.

[5] P.Molander, J.Conflict Resolut. 29 (1985)611.

[6] J.Bendor, Am.J.Polit.Sci. 31(1987)531.

[7] See, e.g. K.Lindgren, "Evolution in a Population of Mutating Strategies" in *Artificial Life II* pp295. (eds. C.Langton et al., Addison Wesley, 1991).

[8] See e.g, J.Hofbauer and K.Sigmund,*The Theory of Evolution and Dynamical Systems* (Cambridge University Press, 1988).

[9] B.A.Huberman and T.Hogg, Physica 22D(1986) 376.

[10] C.P.Bachas and W.F.Wolff, J.Phys.A. 20(1987)L39.

[11] K.Matsuo and N.Adachi, "Ecological Dynamics of Strategic Species in Game World" , IIAS-SIS Research Report No.74, Numazu Japan, 1987.

[12] K.Kaneko and T.Ikegami, Physica 56D(1992)406.