On Evolving Fixed Pattern Strategies for Iterated Prisoner's Dilemma

Daniel Jang, Peter A. Whigham, Grant Dick

Department of Information Science
University of Otago
Dunedin, New Zealand

daniel.jang@paradise.net.nz, {pwhigham,gdick}@infoscience.otago.ac.nz

Abstract

This paper describes the social evolution of an environment where all individuals are repeating patterns of behaviour. The paper follows Axelrod's work [1] of computer simulations of Iterated Prisoner's Dilemma (IPD), which is widely regarded as a standard model for the evolution of cooperation. Previous studies by Axelrod [2], Hirshleifer and Coll [3], Lindgren [4], Fogel [5], Darwen and Yao [6] focused on strategies that are history dependent. In other words, these strategies use the outcome of the opponent's past game history in making a decision on a given move. This includes the most well-known strategy, tit-for-tat.

The way strategies are encoded in the computer program reflects the model's assumption concerning individual decision-making. In this paper, we study environments where all players are simply repeating patterns of behaviour without using past game history. In doing so, a genetic algorithm is used to evolve such strategies in a co-evolution environment. Simulations indicate that such an environment is harmful to the evolution of cooperation.

1 Introduction

Iterated Prisoner's Dilemma (IPD) provides a formal representation of a ubiquitous type of collective action problem that arises when individual interests undermine the collective welfare of the group [6]. The typical prisoner's dilemma involves two players each having two alternative choices: cooperate or defect. Cooperation increases the total gain of both players, whereas defection increases one's own reward at the expense of the other player. However, if both players defect, the total gain of each player is much smaller than that of mutual cooperation. A typical payoff matrix of the game is shown in ***Table 1***.

Table 1 shows the payoff to player one. The same matrix also holds for player two. Player one can gain the maximum 5 points (T = 5) by defection if player two cooperates.

However, if both players defect in the light of maximum profit, both players can only gain 1 point (P =1), which is less than 3 points (R =3) if both cooperated.

		Player 2	
		Cooperate	**Defect**
Player 1	**Cooperate**	R =3	S =0
	Defect	T =5	P =1

Table. 1 - The payoff matrix of the prisoner's dilemma as used in Axelrod [1]

This is an interesting aspect of the game. Individual rationality leads to a worse outcome than is possible, and thus leads to a dilemma. Even though two people make logical choices, their attempt to improve their prospects makes everyone worse off.

If the game is played only once, the most compelling choice is defection. However, if the game is iterated over many trials, the most logical behaviour becomes unclear. Because of such nature, the IPD is used as a model for studying our social world, where ongoing interactions with others take place.

2 Previous Studies

In Axelrod's original work [1], two computer tournaments based on IPD were undertaken. In this process, Axelrod invited professional game theorists to submit programs for playing a computer-based Iterated Prisoner's Dilemma.

According to the *Social Science Citation Index*, Axelrod's work had been quoted more than one thousand times by 1992 and more than 2500 times by March 2000 [10]. This suggests that his approach has been extremely influential in other studies.

In a later article, Axelrod [2] used a Genetic Algorithm (GA), an artificial intelligence technique inspired by biological evolution, to simulate agent learning. The GA uses operators based on evolution such as mutation and crossover. In investigating evolutionary future round tournaments, Axelrod considered a set of strategies that is deterministic and uses outcomes of the past three moves to determine a current move.

Further works [3][4][5][6][7] adapt the same approach. This means that all strategies encoded in these environments are history dependent. In other word, such environment consists only of memory-based strategies.

However, alternative representation of strategies could exist. Kraines [8][9] showed that 'Pavlovian' strategies are able to support the evolution of cooperation in tournaments where players are making errors.

Two well-known strategies, All C (all-time-cooperator) and All D (all-time-defector), simply exhibit a fixed pattern behaviour regardless of the past game history. These two pattern strategies have been frequently included in many simulation environments, along with other memory-based strategies such as tit-for-tat

3 Context of Study

So the question is: what happens in an environment where all players are simply repeating different patterns? There appears no literature that addresses this question.

These players could be organised to have a specific bit length, and to repeat different patterns according to the bit length. For instance, 5 bit patterns include CCCCC (All C), CCCCD, CCCDC … DCDCD … DDDDD (All D) and so on. Players have two choices in each move, either cooperate (C) or defect (D), therefore for a 5 bit pattern length, there exists $2^5 = 32$ different patterns that could be encoded as fixed pattern strategies.

Such an environment is considered to exhibit more controlled and regulated characteristics since player representation and the number of strategies in a given environment can be controlled by variable bit lengths. In addition, each player's move in every game is fixed by the pattern behaviour.

However, one may argue that this kind of environment essentially violates the nature of the IPD as the strategies do not make use of the past game history that is readily available to them. However, an environment where individuals simply repeat patterns of behaviour could also exist in our social interaction with others. A weak analogy could be religious and military groups. Therefore, this work hopes to represent another or alternative dimension of our social world.

To make the investigation more realistic, various degrees of fixed pattern strategies could be mixed with memory-based strategies and the characteristics of such environment can be studied. It is contended that such environment better represents the real world, where different individuals have different behaviours of varying complexity, ranging from a simple pattern to conditional rules based on past experience with the other player. This is clearly an area for further research.

4 The Fixed Pattern Computer Tournament

4.1 N-bit pattern only tournament

Fixed pattern players have a specific bit length, which represents the length of each pattern.

First, we examine simple 2-bit pattern strategies. As players have two choices in each move (either cooperate or defect), there exist $2^2 = 4$ different patterns for 2-bit strategies. These 2-bit patterns play the IPD against each other (including itself) which is iterated 100 times (hence total iteration 2 x 100 = 200) with payoff parameters of R =3, T=5, S=0 and P=1 as defined in ***Table 2***. Pattern details and their total average score are summarised in ***Table 2***:

Pattern Number	Pattern Details	Average Score
0	DD	600
1	DC	450
2	CD	450
3	CC	300

Table 2 - 2-bit pattern strategies IPD tournament summary

Table 2 shows that the mere defector, All D (DD) is the winner of this environment. This is because it can take advantage of cooperative moves of all the other players who do not change their behaviour during the game.

In other words, cooperators suffer in this kind of environment. All C (CC) scored the lowest average of 300.

A more complex environment that consists of 5-bit pattern strategies is the subject of the next investigation. In this case, there are 32 different patterns that make up the IPD tournament environment. The game condition and the parameters remain the same as for 2-bit environment. ***Figure 1*** illustrates the tournament results.

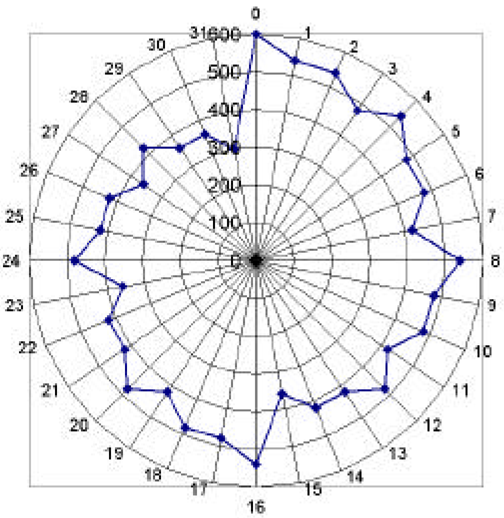

Fig. 1 - 5-bit pattern strategies IPD average score

In ***Figure 1***, numbers inside the circle are the average score, with an increment of 100 (maximum 600 – the highest score achieved). Numbers outside the circle are arbitrary pattern numbers, from No. 0 (DDDDD - All D) to No. 31 (CCCCC All C). Full details of all 32 patterns can be obtained from the authors.

Again, All D is the winner in this environment, with the tournament score of 600. Likewise, All C is again the poorest victim with the lowest score. A surprising fact is that 5-bit All D's score is the same as that of 2-bit All D, despite the fact that more cooperative victims exist in the 5-bit environment.

This occurs because there are also an increased number of defective patterns in this environment, which makes All D perform poorly against them. This effectively cancels out any high scores obtained by exploiting naïve cooperative pattern strategies.

Similarly, 10-bit pattern environment which consists of 1024 different patterns could be considered. This is a truly complex environment where 1024 individuals interact with each other 200 times. This means that each individual plays the IPD 204800 (1024 x 200) times. Simulation confirmed the same results as per 2-bit and 5-bit length environments.

Therefore we can conclude that this particular fixed pattern arrangement is harmful to the evolution of cooperation.

4.2 Mixed bit length environment

In the previous section, we have seen *n*-bit only environments. So a question arises: what happens when different bit lengths are put together in one environment? Do more diverse patterns (longer bit length) perform better? To answer this question, we could put both 2-bit (4 strategies) and 5-bit (32 strategies) patterns into one tournament environment. This way we can see whether the strategies with longer pattern length indeed do better. ***Figure 2*** below illustrate the finding of this analysis:

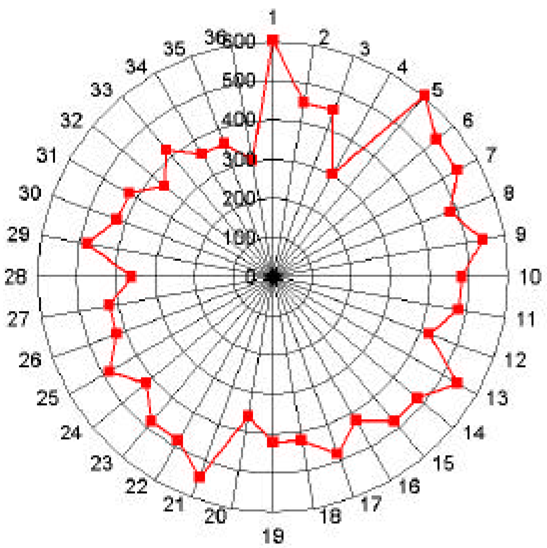

Fig. 2 – combined 2-bit and 5-bit IPD environment average score

In ***Figure 2***, No. 1 to 4 represents 2-bit strategies (DD, DC, CD, CC) and 5 to 36 represent 5-bit strategies (starting from DDDDD to CCCCC as before). Therefore there are a total of 36 patterns playing the IPD.

Surprisingly, all strategies preserved their original average score obtained in their environments. The average score of strategies No. 1 to 4 are identical to those in ***Table 2***.

Therefore, it can be concluded that the length of the patterns has no significant influence on the average performance in a mixed environment, if at all.

4.3 Fixed patterns vs. tit-for-tat

So far, we have examined environments that consist only of fixed pattern strategies. Results suggest that such an environment is harmful to the evolution of cooperation. So the question is: how will tit-for-tat, the most successful strategy in Axelrod's original tournament [1], do in this sort of IPD environment? To answer this question, a new computer simulation is undertaken. In this simulation, tit-for-tat is included in the 5-bit pattern environment. **Figure 3** is the graphical output of the simulation result.

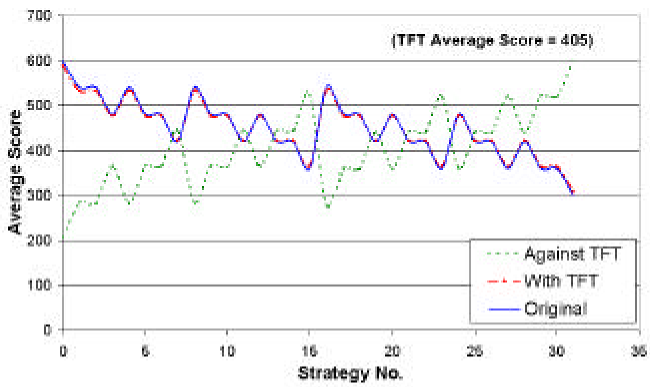

Fig. 3 – 5-bit IPD environment with tit-for-tat average score

In **Figure 3**, the dotted line (Against TFT) represents how well each pattern strategy does against tit-for-tat. As expected, All D (Strategy No. 0) scored the lowest, as repetitive mutual defection is quickly established between All D and tit-for-tat. In contrast, All C (Strategy No. 31) scored the highest. This outcome is essentially 'upside-down' form of the original pattern-only environment. The semi-dotted line (With TFT) shows the average score of each strategy when tit-for-tat is included in the environment. Notice that there is a little difference in the average score between the original and tit-for-tat included environments. But we can see that in the environment which includes tit-for-tat, defective strategies do slightly worse and cooperative strategies do slightly better than the original counterparts.

Tit-for-tat itself scored 405 in the environment. This is lower than the grand average of 5-bit IPD environment, which is 450. This leaves tit-for-tat one of the poorer performers in the environment. This is due to following two reasons:

- Retaliation to defective strategies – although tit-for-tat can do well against cooperative strategies such as All C, half of the strategies in the environment are of defective nature (i.e. more Ds than Cs in their bit string), and tit-for-tat retaliates by throwing defective moves. This results in poor score against them hence cancelling out any gain from mutual cooperation with other co-operators.

- Defectors score more under IPD payoff matrix – like tit-for-tat, defective strategies such as All D do poorly against other defective strategies. However, when defectors meet cooperators, they are bound to gain more than tit-for-tat due to the way IPD payoff matrix is formulated: the payoff of defection (T=5) is greater than mutual cooperation (R=3). Hence all the defectors can outperform tit-for-tat.

Therefore, we can conclude that tit-for-tat cannot do well in a fixed pattern environment due to its unique 'defector wins' nature. This means that tit-for-tat's success is highly dependent on the way other strategies in the environment are encoded and formulated.

4.4 More cooperative environment

In environments where all individuals are repeating a pattern of behaviour, cooperative strategies (i.e. more Cs than Ds in the pattern) were unsuccessful compared to defective strategies (i.e. more Ds than Cs in the pattern). Therefore, it would be interesting to see what would happen in this environment if some of the strategies are randomly replaced by the cooperator, All C. This process essentially 'seed' more cooperators in the environment.

Can defectors do even better due to increased number of 'prey' or can cooperators improve their performance due to less defectors exploiting them? To answer this question, a computer simulation is carried out under the following condition, using 5-bit strategies:

- Randomly replace 25, 50, 75 per cent of all strategies in the environment with All C.

- For each percentage variation, individuals play the usual IPD games and the average score is recorded. Assign a game number counter to each individual which is incremented by 1 each time the game is played.

- Repeat above process 1000 times to ensure validity of simulation. Randomly replaced strategies are not incremented, as this would give the 'incorrect' average score of All C as opposed to that of the original strategy.

- After 1000 iterations, calculate total average score by adding all average scores of each round divided by the game number counter.

Table 3 below illustrates a typical output of this algorithm in a table format. Note that this is a subset of the actual simulation result.

Pattern No.	Average Score	No. of Games
0	688	888
1	623	789
2	623	762
3	559	734
…	…	…

Table 3 – A sample simulation result of the 5-bit environment where 25 per cent of strategies randomly replaced by All C.

A complete simulation result is summarised in ***Figure 4***. In ***Figure 4***, the dotted line represents the original 5-bit environment, which is basically the same pattern as shown in ***Figure 1***.

The simulation result indicates that such variation is beneficial for both cooperators and defectors. All D scored higher with increased percentage of replacement, as expected. In addition, All C improved its result due to increased chance of mutual cooperation with other individuals. Therefore, existence of more cooperators makes everyone better off.

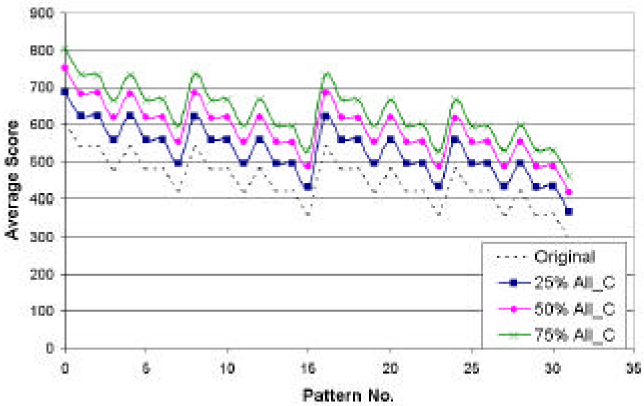

Fig. 4 – Analysis of more cooperative environments (random replacement by All C strategy)

However, note that the improvement of All D is greater than that of All C due to the nature of the IPD payoff matrix (i.e. defection gain more than cooperation). If we compare the original 5-bit environment with the 75 per cent All C replacement in ***Figure 4***, it can be seen that All D improved its average score by over 200. This compares with All C's improvement, 160. Hence it can be concluded that although existence of more cooperators improve everyone's profit, All D continue to dominate in this kind of pattern environments.

5 Evolution of Fixed Pattern Strategies

In complex environments, it is generally difficult for individuals to analyse the situation and invent the optimal strategy to use. Instead, most people are adopting existing strategies that are proven to be effective under certain environments. These successful strategies are found over time, usually after many trials and applications.

This is essentially the same as adaptation processes found in biological evolution. In evolution, strategies that have been relatively effective in a population become more widespread, and strategies that have been less effective become less common in the population.

In a dynamic environment, it is somewhat unrealistic to assume that all strategies are fixed over time. Rather, new and modified strategies need to be constantly introduced at various point of time.

5.1 Overview of the Genetic Algorithm (GA)

The genetic algorithm is a model of machine learning which derives its behaviour from a metaphor of some of the mechanisms of biological evolution in nature. An artificial intelligence procedure developed by John Holland [11][12], it provides useful way of conducting simulation experiments of genetics and evolution.

A GA usually begins with a population of binary strings and each member of the population is evaluated according to the evaluation function called the *fitness*. After evaluation of the fitness of each string, a new population (the offspring) is formed from the old. In this context, the better performers (those who with higher fitness values) are selected for reproduction.

Since reproduction does not actually introduce new strategies to the search space, the last step in the production of new population involve the manipulation of these copied strings via genetic operators [13]. The two main genetic operators are *mutation* and *crossover*.

Mutation is about changing bit stings (from 0 to 1 or vice versa) in a reproduced individual. The mutation rate is usually kept very low [12].

Crossover is the most dominant mechanism of genetic rearrangement [14]. The operation of crossover starts with selection of two individuals. Then a random crossover point is chosen at which each individual in a particular pair is cut into two pieces. Each string then exchanges a section of itself with its partner. This operation is illustrated in ***Figure 5***.

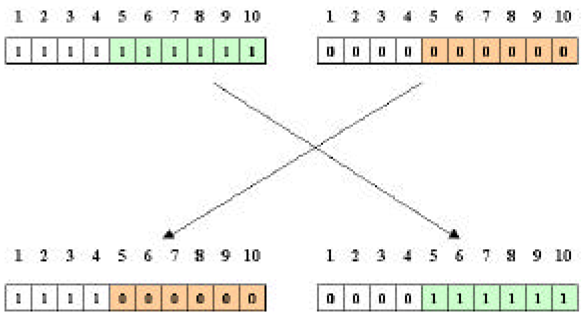

Fig. 5 – crossover between two genotypes, with a cutting point at position 5.

In ***Figure 5***, a random crossover point was chosen at position 5 in the bit string. Two individuals simply swap their segment after position 5 with each other.

5.2 GA used in the study

The outline of the simulation algorithm used in this study is given below. 10-bit strings were used in order to create a large search space.

1. Randomly select n = 200 samples out of 1024 possible strategies, with replacement. This creates an environment made up of randomly chosen fixed strategies which has a total population of 200. This is generation 0 (t =0).

while generation t not filled **do**

   a. Each strategy plays IPD against all others in the environment, as described in section 4.
   b. Fitness function is calculated for each strategy, which is the individual's average IPD tournament score over all competitors.
   c. Parents P_1 and P_2 are selected from the current generation using 3-Tournament selection (Bäck, Hammel and Schwefel, 1997).
   d. Crossover is applied with a probability of 60% to produce new individuals $P_1^{\prime}$ and $P_2^{\prime}$.
   e. Mutation is applied to each bit of each parents (by swapping C to D and vice versa) with a probability of 1%, to produce $P_1^{\prime\prime}$ and $P_2^{\prime\prime}$.
   f. Insert $P_1^{\prime\prime}$ and $P_2^{\prime\prime}$ into generation t.

2. Once new population set of n = 200 is filled, calculate the grand average score of all individuals in the set (generation t). Set $t := t+1$. Repeat the above **while** loop for $t+1$.
3. Repeat step 2 until (t < maxGeneration).

5.3 Results and Analysis

Using the algorithm outlined above, 10-bit pattern strategies were evolved over many generations. All previous simulations suggested that All D is the best strategy to adopt in this kind of environment. So will All D evolve in a long run? What effect will it have on each individual's performance? The GA simulation answers these questions. The simulation result is illustrated in ***Figure 6***.

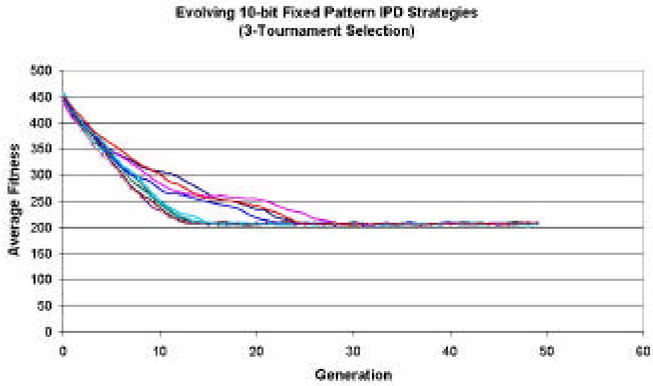

Fig. 6 – Evolving 10-bit fixed pattern strategies using the GA.

These results suggest that the population stabilised at about the 25th generation. From then on, most of the strategies that evolved in the simulation actually resemble All D as expected. Most individuals quickly adapt All D like behaviour and as a result, their preys quickly vanished.

This is an interesting aspect of evolving such an environment. After about the 25th generation, mutual defection is established where everybody essentially threw defective moves and received a payoff of 1 (p=1, ***Table 1***) against other individuals. Since each game consists of 200 iterations (i =200), the total average fitness can be calculated by the following formula:

$$Average = \left(\sum_{j=1}^{n} (\sum_{i=1}^{i} p) \right) \div n$$

where p is payoff, i is the number of iteration in each game and n is the number of selected strategies in each generation (n = 200 in our study)

We may recall from ***Table 1*** that the payoff of mutual cooperation (R=3) is greater than that of mutual defection (P=1). Given 200 iterations in each game, this means a total payoff of 600 (3 x 200) for mutual cooperators and 200 (1 x 200) for mutual defectors.

Therefore, in a fixed pattern environment where everyone adopts the best known strategy (All D), individuals actually receive considerably lower payoffs than they would do if mutual cooperation takes place. But at the same time, it pays not to cooperate as such moves would be exploited by other individuals and the chance of survival is almost zero in this kind of environment.

This leads to a paradox – everyone can improve the payoff gain by 3 times if mutual cooperation is established, but nobody will be prepared to change his or her behaviour due to a desire for survival.

It seems that only way to facilitate mutual cooperation is to 'convince' the population of the cooperative benefits and hence allow them to change their behaviour altogether. This is because no individual in the population will change their behaviour due to the risk of a low payoff.

6 Discussion

So far, we have focused solely on the characteristics of fixed pattern environments. Mixing various degrees of fixed pattern strategies with memory-based strategies can extend this work further.

A number of authors [16][17][18][19] introduced spatial dimensions to the GA matching process in studying memory-based strategies. Likewise, fixed pattern strategies can be arranged to interact with nearest neighbours in a grid and the characteristics of such an environment can be compared with the non-spatial counterpart [19].

7 Conclusion

This paper examined the characteristics and the evolution of various IPD environments where all individuals are repeating patterns of behaviour. The purpose of our study is as follows [20][21]:

- To extract meaningful measures from the seemingly random behaviour.
- To verify that the model has some similarity to the real world
- To use the model to understand how the real world will behave.

In n-bit pattern only environments, All D emerged as the best performer. Tit-for-Tat performed below average in these environments due to their unique 'defector wins' nature. This means that tit-for-tat's success is dependent on the way other strategies in the environment are encoded and formulated.

In the light of promoting cooperation, a new environment set is created with All C randomly replaces some of the existing pattern strategies. Even in this context, All D continued to dominate and improved its profit even further.

Evolution of the fixed pattern environment was carried out using the GA. Simulation results led to an interesting paradox. Individuals perform poorly due to established 'mutually-defective' environment. However, no individuals can possibly change their behaviour due to the hostile defector environment.

Therefore, we can conclude that a fixed pattern only environment is not only harmful to the evolution of cooperation, but also makes it difficult for individuals to become co-operators. Interestingly, the only way to improve everyone is to let them change their behaviour in a synchronised fashion.

8 Acknowledgement

We would like to thank Professor Norman Foo (UNSW, Australia) for help and guidance in developing this project during the initial stages. We also would like to thank Dr. Rex Kwok (UNSW, Australia) and Dr. Colin Aldridge (University of Otago, New Zealand) for feedback, support and ongoing interest in this work.

9 References

[1] Axelrod, R. (1984). *The Evolution of Cooperation*, Penguin Books, England.

[2] Axelrod, R. (1987). *Evolving New Strategies*, Genetic Algorithm and Simulated Annealing, London, pp 32-41.

[3] Hirshleifer, J and Coll, JM (1988). *What strategies can support the evolutionary emergence of cooperation?* Journal of Conflict Resolution 32(2). pp 367-398.

[4] Lindgren, K (1992). *Evolutionary Phenomena in Simple Dynamics*, In Langton, CG(ed.) Artificial Life II. Addison-Wesley.

[5] Fogel, D. (1993). *Evolving Behaviours in the Iterated Prisoner's Dilemma*, Evolutionary Computation 1(1):77-97.

[6] Darwen, P. and X, Yao (1994). *On Evolving Robust Strategies for Iterated Prisoner's Dilemma*, Lecture Notes in Artificial Intelligence, AI '93 and AI'94 Workshops on Evolutionary Computation, Melbourne, Australia.

[7] Darwen, P. and X. Yao. (2001). *Why more choices cause less cooperation in Iterated Prisoner's Dilemma*, Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, South Korea, May 27-30, 2001.

[8] Kraines, D and Kraines, V (1993). *Learning to Cooperate with Pavlov – an adaptive strategy for the Iterated Prisoner's Dilemma with Noise*, Theory and Decision 35 pp.107-150.

[9] Kraines, D and Kraines, V (1995). *Evolution of Learning among Pavlov Strategies in a Competitive Environment with Noise*, Journal of Conflict Resolution v39 i3 pp.439-366.

[10] Hoffmann, R. (2000). *Twenty Years on: The Evolution of Cooperation Revisited*, Journal of Artificial Societies and Social Simulation vol. 3, no. 2.

[11] Holland, J. (1975). *Adaptation in Natual and Artificial Systems*, Ann Arbor: University of Michigan Press.

[12] Holland, J. (1980). *Adaptive Algorithms for Discovering and Using General Patterns in Growing Knowledge Bases*, International Journal of Policy Analysis and Information Systems 4:245-268.

[13] Sutton, P. and Boyden, S. (1994). *Genetic algorithm: A general search procedure*, American Journal of Physics, vol 62, no. 6.

[14] Holland, J. (1992). *Genetic algorithm.* Scientific American, 267(4):44-50, July 1992.

[15] Back, T., Hammel, U. and Schwefel H. (1997). Evolutionary Computation: *Comments on the History and Current State*, IEEE Transactions on Evolutionary Computation Vo l1:1, pp3-17, April 1997.

[16] Nowak, M. A. and May, R. M. (1992). *Evolutionary Games and Spatial Chaos*. Nature 359. pp 826-829.

[17] Lindgren, K and Nordahl MG (1994). *Evolutionary dynamics of spatial games*, Physica D v75 i1-3 p.292-309.

[18] Hoffmann, R. (1999). *The independent localisations of interaction and learning in the Repeated Prisoner's Dilemma*. Theory and Decision 47. pp.57-72.

[19] Gamble, T and X Li (2002). *Emergence of Cooperation in the IPD Game using Spatial Interactions*, AI 2002 Workshop proceedings, The 6th Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems, Canberra, Australia.

[20] Brembs, B. (1996). *Chaos, cheating and cooperation: potential solutions to the Prisoner's Dilemma*, Minireviews, OIKOS 76: 14-24, Copenhagen 1996.

[21] Whigham, P. and Fogel, G. (2002). *Ecological Applications of Evolutionary Computation*, Ecological Informatrics, F Recknagel (Ed). 2002 Springer Verlag. 49-71.