Chapter 3

Learning IPD Strategies Through Co-evolution

Siang Yew Chong[1], Jan Humble[2], Graham Kendall[2], Jiawei Li[2,3], Xin Yao[1]

University of Birmingham[1], University of Nottingham[2], Harbin Institute of Technology[3]

3.1. Introduction

Complex behavioral interactions can be abstracted and modelled using a game. One particular aspect in modelling interactions that is of great interest is in understanding the specific conditions that lead to cooperation between selfish individuals. The iterated prisoner's dilemma (IPD) game is one famous example. In its classical form, two players engaged in repeated interactions, are given two choices: cooperate and defect [Axelrod (1984)]. The dilemma of the game is captured by having both players who are better off mutually cooperating than mutually defecting being vulnerable to exploitation by one of the party who defects. Although the IPD game has become a popular model to study conditions for cooperation to occur among selfish individuals, which was due in large part to a series of tournaments reported in [Axelrod (1980a,b)], it has also received much attention in many other areas of study, and used to model social, economic, and biological interactions [Axelrod (1984)].

The classical IPD can be easily defined as a nonzero-sum, noncooperative, two-player game [Chellapilla and Fogel (1999)]. It is nonzero-sum because the benefits that a player obtains do not necessarily lead to similar penalties given to the other player. It is noncooperative because it assumes no preplay communication between the two players.

The IPD game can be formulated by considering a predefined payoff matrix that specifies the payoff that a player receives for the choice it makes for a particular move given the choice that the opponent makes. Referring to the payoff matrix given by figure 3.1, both players receive R (*reward*) units of payoff if both cooperates. They both receive P (*punishment*) units of payoff if they both defect. However, when one player cooperates while the other defects, the cooperator will receive S (*sucker*) units of payoff while the defector receives T (*temptation*) units of payoff.

With the IPD game, the values R, S, T, and P must satisfy the constraints; $T > R > P > S$ and $R > (S + T)/2$. Axelrod in [Axelrod (1980a,b)] used the following set of values: $R = 3$, $S = 0$, $T = 5$, and $P = 1$. However, any set of values can be used as long as they satisfy the IPD constraints. The game is played when both players choose between the two alternative choices over a series of moves (i.e., repeated interactions). Note that the game is fully symmetric, i.e., the same payoff matrix is applied to both players.

	Cooperate	Defect
Cooperate	R (upper right), R (lower left)	T (upper right), S (lower left)
Defect	S (upper right), T (lower left)	P (upper right), P (lower left)

Fig. 3.1. The payoff matrix framework of a two-player, two-choice game. The payoff given in the lower left-hand corner is assigned to the player (row) choosing the move, while that of the upper right-hand corner is assigned to the opponent (column).

For the simple case of the one-shot prisoner's dilemma (both players only get to make one move), the rational play will be to defect [Chellapilla and Fogel (1999)]. This can be viewed by considering the obtained payoff for a choice made by a player in light of the opponent's. For example, a cooperating player will receive either R (opponent cooperates) or S (opponent defects). A defecting player will receive either T (opponent cooperates) or P (opponent defects). As such, from the player's point of view (i.e., self-interested), the rational play will be to defect because regardless of the opponent's play, a higher payoff is obtained ($T > R$ and $P > S$).

However, when the game is iterated over many rounds of moves and that players can adopt game strategies where a response is based on what happened in the previous moves, defection is not necessarily the best choice of play. Instead, many studies have shown cooperative play to be a viable strategy, starting with the tournaments organized by Axelrod (reported in [Axelrod (1980a,b)]). More importantly, later studies (of which Axelrod himself is one of the early pioneers) showed that cooperative strategies can be learned from an initial, random population using evolutionary algorithms [Axelrod (1987); Fogel (1991, 1993); Darwen and Yao (1995)].

In particular, studies made in [Axelrod (1987); Fogel (1991, 1993); Darwen and Yao (1995)] (and many others) used a co-evolutionary learning approach. The motivation for the co-evolutionary learning approach is the learning of strategy behaviors through an adaptation process on strategy representations based solely on interactions (i.e., game-play). This approach is different compared to the classical evolutionary game approach (and also the ecological game approach used in [Axelrod (1980b); Axelrod and Hamilton (1981)]) that is mainly concerned with frequency dependent reproductions of fixed and predetermined strategies. As such, the use of co-evolutionary learning approach allows for one to construct a game (i.e., specifying the possible interactions between players, the rules that govern the interactions, and the payoffs) and then to search for effective game strategies without the need of human intervention (e.g., specify viable strategies) [Chellapilla and Fogel (1999)].

Within the framework of the co-evolutionary learning of game strategies, it is natural to explore more complex interactions that is closer to real-world interactions compared to highly abstracted models like the classical IPD. This review aims to provide a survey of studies using the co-evolutionary learning approach of more complex IPD games since the tournaments organized by Axelrod that were held almost 20 years ago. In particular, focus is placed on the motivations of certain extensions to the classical IPD and the general observations made when co-evolutionary learning systems are used.

The following section describes the framework of co-evolutionary learning and the general issues of co-evolving IPD strategies. Section 3.2 surveys studies that extend the classical IPD with more choices, noise, N-players, and others. The review concludes with some remarks on the future directions for research in co-evolutionary learning of IPD strategies. It is emphasized again that this review focusses on the co-evolutionary learning approach to IPD games, rather than all possible work related to IPD games.

3.2. Co-evolving Strategies for the IPD Game

3.2.1. *Co-evolutionary Learning Framework*

Co-evolutionary learning refers to a broad class of population-based, stochastic search algorithms that involves the simultaneous evolution of competing solutions (to a problem) with coupled fitness [Yao (1994)]. A co-evolutionary learning system can be implemented using evolutionary algorithms (EAs) [Fogel (1994a); Bäck *et al.* (1997)]. That is, a co-evolutionary learning system iteratively apply the process of variation (e.g., mutation, crossovers, and others) and selection (e.g., choosing solutions to procreate in the next iterative step) on the competing solutions in the population. With this view, the framework of co-evolutionary learning (and also that of EAs) can be illustrated using figure 3.2.

---

(1) Initialize the population, X(t=0)
(2) Evaluate the fitness of each individual through a comparison process with other individuals in X(t)
(3) Select parents from X(t) based on their evaluated fitness
(4) Generate offsprings from parents to produce X(t+1)
(5) Repeat steps (2-4) until some termination criteria are reached

---

Fig. 3.2. The general framework of co-evolutionary learning.

Co-evolutionary learning is different from EAs in the sense of assigning fitness, i.e., the quality or worth of a solution (Step 2 in Fig. 3.2). EAs are often viewed and constructed in terms of an *optimization* context, whereby an absolute fitness function is required to assign fitnesses to contending solutions. With co-evolutionary learning, the fitness of a solution is obtained through its interactions with other contending solutions in the population. That is, a solution fitness in a co-evolutionary learning system is *relative* and *dynamic* because a solution's fitness not only depends on the population, but also changes as the composition of solutions in the population changes.

Although the difference between co-evolutionary learning systems and traditional EAs appear to be small at first, from the contexts of certain problems, it can lead to significantly different outcomes. For example, consider the problem of searching for *optimal* solutions. In many real-world problems, designing a suitable fitness function that can lead to the search of solutions can be very difficult, if possible [Yao (1994)]. However, with co-evolutionary learning, this need of having a fitness function is essentially removed. Instead, a co-evolutionary learning system only needs to be able to rank contending solutions based on how they compared to one another.

Here, games are well-suited, natural problem applications for co-evolutionary learning systems. In particular, although games can be approached from an optimization context, it may not be possible to construct a fitness function that fully represent the problem of the game and fully discriminate solutions found through optimization algorithms. With co-evolutionary learning, however, the search can be directed to find for *better* game strategies (e.g., defeat more strategies) as the evolutionary process continues [Chellapilla and Fogel (1999)].

In particular, for the IPD game, there have been many different approaches since Axelrod's early study in [Axelrod (1987)] that investigated a particular co-evolutionary learning system. Like the study of EAs (commonly known as Evolutionary Computation) [Yao (1994); Fogel (1994a); Bäck *et al.* (1997); Fogel (1995); Bäck (1996)], there are a wide variety of specific strategy representations, selection and variation operators in the co-evolutionary learning approach used for the IPD game. A complete survey is beyond the scope of this chapter. Instead, the more popular choices will be reviewed here. The important thing to note is that all the co-evolutionary learning systems used were based on the framework illustrated in figure 3.2, i.e., they involved an adaptation process on IPD strategies in some form of representations (involves variations and selection) based on interactions (game-play between strategies).

For strategy representations, particularly on deterministic and reactive IPD strategies that were mostly studied, Axelrod and Lindgren [Axelrod (1987); Lindgren (1991)] were among the first few who used binary strings of ones (cooperation) and zeroes (defection) encoding for a look-up table (essentially a binary decision tree) representation. The look-up table in particular determines the outcome for the strategy based on the pairs of previous moves made by the strategy and the opponent. Since the strategies require histories of previous moves in order to make a response, they are encoded with the necessary histories for previous moves. We [Chong and Yao (2005)] recently introduced a look-up table representation that directly represents IPD strategies based on responses to previous moves. For the case of looking back the previous pair of moves made by the strategy and the opponent, direct look-up table represents the strategy responses as a two-dimensional table. Each table element represents the response based on the pair of previous moves. Instead of some fictitious histories required to start the game, the direct look-up table specifies the first move directly.

Fogel among many others [Fogel (1991, 1993, 1996); Miller (1989); Stanley *et al.* (1995)] used finite state machines (FSMs) for their capability of representing complex behaviors of IPD strategies. With FSMs, behavioral responses of an IPD strategy based on previous moves depend on the states and the next-state transitions. The motivation for using FSM compared to look-up table is to have a behavioral representation of IPD strategies instead of the look-up table representation of responses based on histories of previous moves (see [Fogel (1993)] for the full discussion on the origin of using FSM and evolution to simulate intelligent behaviors).

In addition to the simple look-up table and FSM, neural network representations had also been experimented with and studied [Harrald and Fogel (1996); Darwen and Yao (2000); Chong and Yao (2005); Franken and Engelbrecht (2005)]. Although neural networks are primarily used for their ability of providing nonlinear input-output responses [Chellapilla and Fogel (1999)], the initial motivation to representing IPD strategies also include the capability of neural networks to process and represent a continuous range of behaviors [Harrald and Fogel (1996)].

After selecting a strategy representation, the next step is to consider the design of variation operators that are aimed at providing variations of IPD strategies in the population. In most cases, variation operators are dependent of the strategy representation considered. For example, look-up table encoded as binary strings can use crossovers and bit-flip mutation as in the case of standard genetic algorithms [Axelrod (1987)]. For the case of FSMs, variation operators may include altering a next-state transition, adding or removing states, and altering the output symbol (corresponding to making a choice). With neural networks, especially those that are real-valued representations, self-adapting mutations based on some probability distribution (i.e., Gaussian or Cauchy) can be used [Chong and Yao (2005)] (one of us has provided a comprehensive review on evolving neural networks in [Yao (1999)]).

As for designing the process of selecting IPD strategies for the next generation, many other selection operators can be used (those found in EAs [Fogel (1994a); Bäck *et al.* (1997)]) and not just limited to proportional selection used by Axelrod in the first study of co-evolving IPD strategies [Axelrod (1987)]. For the case of obtaining the fitness for a particular IPD strategy in the population, payoffs obtained from the IPD game are usually used. In particular, many studies considered calculating the expected IPD-payoff-based-fitness using a round robin tournament whereby all pairs of strategies compete, including the pair where a strategy plays itself.

3.2.2. *Shadow of the Future*

In the IPD game, the shadow of the future refers to the situation whereby the number of moves of a game is known in advance. In this situation, there is no incentive to cooperate in the last move because there is no risk of retaliation from the opponent. However, if every player defects on the last move, then there is no incentive to cooperate in the move prior to the last one. If every player defects in the last two moves, then there is no incentive to cooperate in the move before that, and so forth. As such, we would end up with mutual defection in all moves.

One popular way to address this issue and to allow for cooperation to emerge is to have a fixed probability in ending the game on every move, thereby keeping the game length uncertain. Most of the studies that used the co-evolutionary learning approach considered a fixed game length (number of moves) in all game plays. For example, Axelrod [Axelrod (1987)] and others such as [Fogel (1991, 1993); Chong and Yao (2005)] used 150 moves (move start from 0). Other game lengths can be used, although the choice depends on the motivation of the study, e.g., a sufficiently long game length to allow for strategies to reciprocate cooperation. In any case, the fixed game length is used because the strategy representation cannot count the number of moves that have been played and how many more remain.

3.2.3. *Issues for Co-evolutionary Learning of IPD Strategies*

For the IPD game, there are two main contexts in which co-evolutionary learning can be considered. First, co-evolutionary learning can be used to search for effective strategies, given the specific the rules of the game that govern the complexity of strategy interactions. Second, a co-evolutionary learning system can serve as a model for investigating how certain conditions (e.g., game rules, co-evolutionary learning system setup, or others) can lead to the evolution of certain behaviors.

For the context of using co-evolutionary learning to search for effective strategies, the main issue is to evolve IPD strategies that perform well (e.g., defeat) against a large number of opponents. Axelrod [Axelrod (1987)] used a co-evolutionary learning system and compared the evolved strategies with the representative strategies (e.g., tit for tat) obtained from his earlier tournaments that accounted for average performance of all strategies that participated the tournaments [Axelrod (1980a,b)]. He noted that some of the evolved strategies outperformed these representative strategies.

Although results obtained from evolving effective IPD strategies were promising, the study in [Axelrod (1987)] had the important implication on specifying a principled method to determine the effectiveness (or robustness [Axelrod and Hamilton (1981)]) of evolved IPD strategies by testing them against some representative strategies. One of us (Yao) first framed this particular study in the context of *generalization* [Darwen and Yao (1995); Yao *et al.* (1996)]. In particular, co-evolutionary learning is a machine learning system that can be analyzed for its generalization performance. Here, the generalization performance of a co-evolutionary learning system for the IPD game can be thought of as the performance of the best strategy in the population or the population itself (e.g., using a gating algorithm that effectively combines different IPD strategies of the population as a single strategy entity [Darwen (1996); Darwen and Yao (1997)]) against a large number of IPD strategies, especially those that the evolved strategies have yet to play with during evolution.

For the context of using co-evolutionary learning as a model to understand the conditions of how, why, and what IPD strategy behaviors are evolved, there are many issues that can be studied. First, one can consider the impact of specific IPD game specifications (e.g., payoff matrices [Fogel (1993)] and duration of interactions or game length [Fogel (1996)]) on evolved IPD strategy behaviors. Second, there are also studies that have focused on the impact of the interaction or game-play itself, which are not just limited to noisy interactions [Julstrom (1997)], continuous behavioral responses [Harrald and Fogel (1996)], and the possibility of refusal to interact [Stanley *et al.* (1995)]. Third, the specific the design of the co-evolutionary learning system itself can have an impact whereby certain IPD behaviors are favored and persist for a long period (e.g., investigating whether systems that provided genotypic diversity actually lead to a diverse population of IPD strategies with a variety of behaviors [Darwen and Yao (2000, 2001, 2002)]).

3.3. Extending the IPD Game

The primary motivation in most studies that extend the classical IPD game is to model more complex IPD interactions that are closer to real-world interactions. This section describes some of the extended IPD games that have been investigated using the co-evolutionary learning approach. Each subsection starts with the motivation for extending the IPD game in a specific manner, and the important issues of studying the more complex IPD games. Each subsection discusses and concludes general observations obtained from the co-evolutionary learning of the particular extended IPD game.

3.3.1. *Extending the IPD with More Choices*

Several studies have extended the classical IPD with more than two extreme choices that are available for play. That is, there are intermediate choices between full cooperation and full defection that strategies can response with. Fogel [Harrald and Fogel (1996)] investigated a continuous IPD game. We have investigated the IPD with multiple, discrete levels of cooperation [Darwen and Yao (2000, 2001, 2002); Chong and Yao (2005)], which could be use to approximate the continuous IPD game when the number of levels is sufficiently large.

The main motivation of extending the IPD with more choices is to allow for the modelling of subtle behavioral interactions that are not possible with only two extreme choices. With the classical IPD game, the possible behaviors that strategies can exhibit are severely limited. For example, a strategy for the classical IPD game cannot play intermediate choices that allow for some degree of exploitation of the opponent without risking retaliation from an otherwise cooperative opponent [Harrald and Fogel (1996)].

The co-evolutionary learning approach usually considers a neural network strategy representation because it can be used to process a continuous range of behaviors (i.e., real numbers for representing the degree of cooperation) easily. Furthermore, for the case of IPD games with multiple, discrete levels of cooperation, a neural network is scalable to the number of levels considered.

Fogel [Harrald and Fogel (1996)] showed that for the extended IPD a continuous range of choices, the evolution of cooperation is unstable, with fluctuations of average scores representing short periods of cooperation and defection. We have further shown that with increasingly higher number of choices to play in the IPD game with multiple, discrete levels of cooperation, evolution to cooperation are more difficult to achieve [Darwen and Yao (2000, 2001, 2002)].

From these studies, it appears that a co-evolving population of IPD strategies has a higher tendency of evolving to play full defection. However, this does not mean that evolution to cooperation is not possible, or that cooperative behaviors that persist cannot be evolved. For example, it has been shown that evolving cooperative behaviors depends on the complexity of strategy representation that is used. In the case of neural networks, the number of nodes in the hidden layer can affect the co-evolutionary learning system to produce IPD strategies with cooperative responses [Harrald and Fogel (1996)].

In addition to the complexity of strategy representation, another important factor for evolving cooperative strategies is that of behavioral diversity. Early studies [Darwen and Yao (2000, 2001)] have shown that genetic diversity (i.e., variations at the genotypic level of strategy representations) does not equate to behavioral diversity (i.e., variations of IPD strategy responses) in the population. Without sufficient behavioral diversity, the co-evolving population can overspecialize to a specific strategy behavior that is vulnerable to invasion (e.g., cycles between tit for tat, naive cooperators, and defectors). As such, increasing the level of genetic diversity in the co-evolutionary learning system does not necessarily lead to an increase in behavioral diversity that can help with the evolution of cooperative strategies.

We have recently further shown that strategy representation also plays an important factor in introducing behavioral diversity in the co-evolutionary learning system [Chong and Yao (2005)]. We considered the n-choice IPD game, which was obtained based on the following linear interpolation:

$$p_A = 2.5 - 0.5c_A + 2c_B, \quad -1 \leq c_A, c_B \leq 1,$$

where p_A is the payoff to player A, given that c_A and c_B are the cooperation levels of the choices that players A and B make, respectively. Fogel [Harrald and Fogel (1996)] also considered a similar interpolation process. However, we considered multiple, discrete levels of cooperation. For example, we used the *four*-choice IPD game, where the four cooperation levels are represented as $+1$ (full cooperation), $+1/3$, $-1/3$, and -1 (full defection). These choices can be used with the linear interpolation equation shown above to obtain the payoff. Figure 3.3 illustrates the payoff matrix of a *four*-choice IPD game that was used [Chong and Yao (2005)].

Note that in generating the payoff matrix for a n-choice IPD game, the following conditions must be satisfied [Chong and Yao (2005)]:

		PLAYER B			
		$+1$	$+\frac{1}{3}$	$-\frac{1}{3}$	-1
	$+1$	4	$2\frac{2}{3}$	$1\frac{1}{3}$	0
PLAYER	$+\frac{1}{3}$	$4\frac{1}{3}$	3	$1\frac{2}{3}$	$\frac{1}{3}$
A	$-\frac{1}{3}$	$4\frac{2}{3}$	$3\frac{1}{3}$	2	$\frac{2}{3}$
	-1	5	$3\frac{2}{3}$	$2\frac{1}{3}$	1

Fig. 3.3. The payoff matrix for the two-player $four$-choice IPD used in [Chong and Yao (2005)]. Each element of the matrix gives the payoff for Player A.

(1) For $c_A < c'_A$ and constant c_B: $p_A(c_A, c_B) > p_A(c'_A, c_B)$,
(2) For $c_A \leq c'_A$ and $c_B < c'_B$: $p_A(c_A, c_B) < p_A(c'_A, c'_B)$, and
(3) For $c_A < c'_A$ and $c_B < c'_B$: $p_A(c'_A, c'_B) > \frac{1}{2}(p_A(c_A, c'_B) + p_A(c'_A, c_B))$.

These conditions are analogous to those for the classical IPD's. The first condition ensures that defection always pays more. The second condition ensures that mutual cooperation has a higher payoff than mutual defection. The third condition ensures that alternating between cooperation and defection does not pay in comparison to just playing cooperation.

We investigated two strategy representation: neural networks and direct look-up table. We considered these two strategy representations because they allow the investigation on the impact of strategy representation on the introduction and maintenance of variations of behavioral responses in the population of IPD strategies. On the one hand, the neural network indirectly represents the input-output response mappings of IPD strategies, with possibilities of many-to-one mappings between representations and actual behavioral responses [Fogel (1994b); Atmar (1994)]. On the other hand, the direct look-up table directly represents the input-output response mappings of IPD strategies. We hypothesized that a more direct representation of IPD strategies will allow more behavioral variations to be introduced and maintained in the population through co-evolution.

For the neural network representation, we used a fixed-architecture feed-forward multilayer perceptron (MLP) [Chong and Yao (2005)]. Specifically, the neural network consists of an input layer, a single hidden layer of ten nodes, and an output node. The network is fully connected and strictly layered (i.e., no short-cut connection from the input layer to the output node. The transfer (activation) function used for all nodes is the hyperbolic tangent function, $tanh(x)$. The input layer consists of the following four input nodes:

(1) The neural network's previous choice, i.e., level of cooperation, in [−1, +1].
(2) The opponent's previous level of cooperation.
(3) An input of +1 if the opponent played a lower cooperation level compared to the neural network, and 0 otherwise.
(4) An input of +1 if the neural network played a lower cooperation level compared to the opponent, and 0 otherwise.

The input layer is a function of two variables (e.g., neural network's previous choice and the opponent's previous choice) since the last two inputs are derived from the first two inputs. These additional inputs are to facilitate learning the recognition of being exploited and exploiting. Given the inputs, the neural network's output determines the choice for its next move. The output is a real value between +1 and −1 that is discretized to either +1, +1/3, −1/3 or −1, depending on which discrete value the neural network output is closest to.

We considered self-adaptive mutation for variation operators for the real-valued representation of neural networks that we used [Chong and Yao (2005)]. This approach associates a neural network with a self-adaptive parameter vector $[\sigma_i(j)]$ that controls the mutation step size of the respective weights and biases of the neural network $[w_i(j)]$. Offspring neural networks ($[w'_i(j)]$ and $[\sigma'_i(j)]$) are generated from parent neural networks ($[w_i(j)]$ and $[\sigma_i(j)]$) through mutations. Two different mutations based on Gaussian and Cauchy distributions were used in order to further investigate the impact of indirect strategy representation on variation operators that could increase genetic diversity but not necessarily lead to increase in behavioral diversity.

For the self-adaptive Gaussian mutation, offspring neural networks are generated according to the following equations:

$$\sigma'_i(j) = \sigma_i(j) * exp(\tau * N_j(0,1)); \quad i = 1 \ldots 15, \ j = 1, \ldots, N_w,$$

$$w'_i(j) = w_i(j) + \sigma'_i(j) * N_j(0,1); \quad i = 1 \ldots 15, \ j = 1, \ldots, N_w,$$

where $N_w = 63$, $\tau = (2(N_w)^{0.5})^{-0.5} = 0.251$, and $N_j(0,1)$ is a Gaussian random variable (zero mean and standard deviation of one) resampled for every j. N_w is the total number of weights, biases, and the pre-game inputs required for an IPD strategy based on memory length of one.

For the self-adaptive Cauchy mutation that is known to provide bigger changes to the neural network weights (i.e., provide more genetic diversity) [Yao *et al.* (1999)], the following equations are used:

$$\sigma'_i(j) = \sigma_i(j) * exp(\tau * N_j(0,1)); \quad i = 1 \ldots 15; j = 1, \ldots, N_w,$$

$$w'_i(j) = w_i(j) + \sigma'_i(j) * C_j(0,1); \quad i = 1 \ldots 15; j = 1, \ldots, N_w,$$

where $C_j(0,1)$ is a Cauchy random variable (centered at zero and with a scale parameter of 1) resampled for every j. All other variables remain the same as those in the self-adaptive Gaussian mutation.

For the direct look-up table representation, the details can be illustrated by figure 3.4 [Chong and Yao (2005)], which shows the behavioral response of a *four*-choice IPD strategy. m_{ij} specifies the choice to be made, given the inputs i (player's own previous choice) and j (opponent's previous choice). Rather than using pre-game inputs (two for memory length one strategies), the first move is specified independently. Each of the table elements can take any of the possible four choices (+1, +1/3, −1/3, −1).

		Opponent's Previous Move			
		$+1$	$+\frac{1}{3}$	$-\frac{1}{3}$	-1
	$+1$	m_{11}	m_{12}	m_{13}	m_{14}
Player's	$+\frac{1}{3}$	m_{21}	m_{22}	m_{23}	m_{24}
Previous Move	$-\frac{1}{3}$	m_{31}	m_{32}	m_{33}	m_{34}
	-1	m_{41}	m_{42}	m_{43}	m_{44}

Fig. 3.4. The look-up table representation for the two-player IPD with four choices and memory length one [Chong and Yao (2005)].

A simple mutation operator was used to generate offspring. Mutation replaces the original element, m_{ij}, by one of the other three possible choices with an equal probability. For example, if mutation occurs at $m_{13} = +1/3$, then the mutated element m'_{13} can take either +1, −1/3, or −1 with an equal probability. Each table element has a fixed probability, p_m, of being replaced by one of the remaining three choices. The value p_m is not optimized. Crossover is not used in any of the experiments. With a direct representation of IPD strategy behaviors, a simple mutation is more than sufficient to provide behavioral diversity in the population.

The following co-evolutionary procedure was used [Chong and Yao (2005)]:

(1) Generation step, $t = 0$:
Initialize $N/2$ parent strategies, $P_i, i = 1, 2, ..., N/2$, randomly.
(2) Generate $N/2$ offspring, $O_i, i = 1, 2, ..., N/2$, from $N/2$ parents using a variation.
(3) All pairs of strategies compete, including the pair where a strategy plays itself (i.e., round-robin tournament). For N strategies in a population, every strategy competes a total of N games.
(4) Select the best $N/2$ strategies based on total payoffs of all games played. Increment generation step, $t = t + 1$.
(5) Step 2 to 4 are repeated until termination criterion (i.e., a fixed number of generation) is met.

In particular, we used $N = 30$, and repeated the co-evolutionary process for 600 generations (which is sufficiently long to observe an evolutionary outcome, e.g., persistent cooperation). A fixed game length of 150 iterations is used for all games. Experiments are repeated for 30 independent runs. Note that additional steps were taken to ensure that the initial population has sufficient behavioral diversity in addition to genotypic diversity [Darwen and Yao (2000)] to avoid early convergence of results. All details are available in [Chong and Yao (2005)]. The procedure involves setting particular parameters for specific strategy representation and resampling for new strategies to make sure that the frequency at which each of the four choices ($+1$, $+1/3$, $-1/3$, -1) is played is approximately similar so that there is no bias to play a particular choice early in the evolution.

Results showed that there were fewer number of runs where the population evolved to play mutual cooperation in experiments that used neural network representations [Chong and Yao (2005)]. For example, some runs had intermediate outcomes while a few had defection outcomes (Fig. 3.5). This is quite different from the case for classical IPD games [Axelrod (1987); Darwen and Yao (1995)] where each run converged to mutual cooperation quite consistently and quickly. Increasing genetic diversity (e.g., using self-adaptive Cauchy mutation) do not necessarily lead to more behavioral diversity in the population since some runs still evolved to intermediate or defection outcomes (Fig. 3.6). The results further illustrates that more choices have made cooperation more difficult to evolve.

However, when direct look-up table representation was used, results

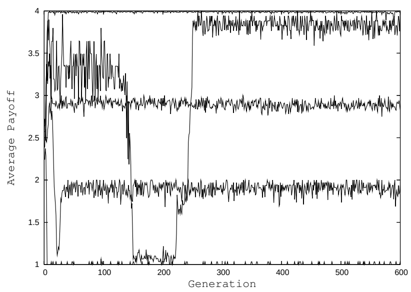

Fig. 3.5. Five sample runs of a co-evolutionary learning system that used neural network representation with a self-adaptive Gaussian mutation in the *four*-choice IPD [Chong and Yao (2005)].

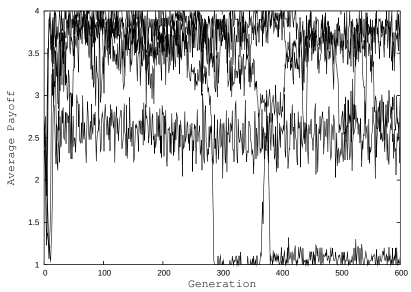

Fig. 3.6. Five sample runs of a co-evolutionary learning system that used neural network representation with a self-adaptive Cauchy mutation in the *four*-choice IPD [Chong and Yao (2005)].

showed that the evolution to cooperation was not difficult [Chong and Yao (2005)]. For example, results showed that even when a simple mutation with a low probability of mutation (e.g., $p_m = 0.05$) was used, no run evolved to mutual defection even though intermediate outcomes were obtained (Fig. 3.7). However, increasing the probability of mutation resulted with all populations in all runs evolving to mutual cooperation play. The results showed that the choice of strategy representation can have an impact on the evolution of cooperation if it allows for greater behavioral diversity in the population.

3.3.2. *IPD with Noise*

A natural extension to the classical IPD is to consider the impact of noisy interactions on the evolution of certain behaviors. Axelrod noted two types

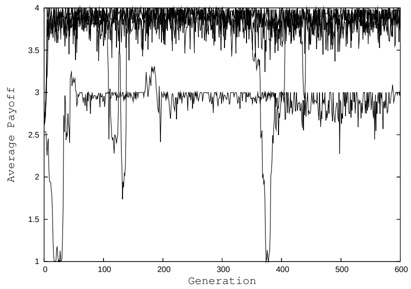

Fig. 3.7. Five sample runs of a co-evolutionary learning system that used direct look-up table representation with a simple mutation at $p_m = 0.05$ in the *four*-choice IPD [Chong and Yao (2005)].

of noise, i.e., misimplementation and misperception, that can affect a strategy's response to the opponent's choice of play [Axelrod and Dion (1988)]. With misimplementation, the strategy knows a mistaken play but the opponent does not know. With misperception, one or both interacting strategies may not know that a different choice was made. The main motivation for this extension is to study the impact of noise on the learning of certain behaviors through co-evolution when interactions can be noisy. In particular, one issue that can be considered is whether cooperative strategies based on reciprocity (such as tit for tat) can still perform well when noise, which affects strategy behavioral response based on previous moves, is present.

Julstrom [Julstrom (1997)] investigated the effects of noise in the *two*-choice IPD through a co-evolutionary learning system. In particular, noise was modelled as mistakes. That is, there is a probability that the choice played by a strategy is changed to the other choice (e.g., defection is played instead of the original cooperation, and vice versa). Results from the experiments showed that noise (starting around 2%) can reduce the level of cooperation in the population.

Recently, we further extended the IPD game with more choices by introducing noise and used a co-evolutionary learning system as a model for investigations [Chong and Yao (2005)], which we have detailed in the earlier subsection. We also modelled noise as mistakes that a player makes. For the *four*-choice IPD game, there is a certain probability of occurrence, p_n, and is fixed throughout a game where a strategy intends to play a particular choice but ends up with a different choice instead. For example, with $p_n = 0.05$, there will be a 0.05 probability that if 1/3 is intended to be played, one of the other three possible cooperation levels, i.e., $+1$, $-1/3$, and −1, will be chosen uniformly at random.

Results from experiments again showed the importance of behavioral diversity for the evolution of cooperation for noisy IPD games with more choices. For noise introduced at very low probabilities (less than 1.5% or $p_n = 0.0015$), evolution to cooperation is more likely than the case when noise was not introduced. Strategies were observed to be more forgiving, confirming the predictions of other studies noted in [Axelrod and Dion (1988); Wu and Axelrod (1995)]. However, when noise was introduced at high probabilities (starting around 5% or $p_n = 0.05$), evolution to cooperation was more difficult. The population was more likely to evolve to defection.

Despite this, if the co-evolutionary learning system has sufficient behavioral diversity (e.g., using direct look-up table representation that allows for behavioral diversity to be introduced and maintained more easily and effectively), evolution of cooperation is not greatly affected [Chong and Yao (2005)]. Evolved strategies still played high levels of cooperation even when there are more choices to play and that the interactions can be noisy, both which can contribute to more difficulty of evolving cooperative behaviors. For example, table 3.1 compares different co-evolutionary learning system with different levels of behavioral diversity, e.g., C-CEP (neural network and self-adaptive Gaussian mutation), C-FEP (neural network and self-adaptive Cauchy mutation), C-PM05 (direct look-up table and mutation at $p_m = 0.05$) for different noise levels (%) [Chong and Yao (2005)]. Results show the number of runs for each experiment that evolved to mutual defection, e.g., average payoff less than 1.5. The table showed that no runs evolved to mutual defection when direct look-up table representation was used in the co-evolutionary learning system [Chong and Yao (2005)].

Table 3.1. Comparison of results for three different co-evolutionary learning systems.

Noise (%)	C-CEP	C-FEP	C-PM05
0	4	1	0
5	4	9	0
10	7	11	0
15	8	17	0
20	18	26	0

It should be noted that although both mutation and noise can be considered as sources of behavioral variations in models that encourage cooperation [Mcnamara *et al.* (2004)], they produce behavioral diversity differently. Mutation introduces strategies with different behaviors into the population. Noise allows other parts of a strategy's behavior that are not played otherwise in a noiseless IPD game to be accessed. Our results [Chong and Yao (2005)] showed that noise does not necessarily promote behavioral diversity in the population that lead to a stable evolution to cooperation, although noise at low levels does help. With higher levels of noise, closer inspection of evolved strategies showed the population to overspecialize to a specific behavior that is vulnerable to invasion, leading to cyclic dynamics in the evolutionary process between cooperation and defection.

In particular, noise and mutation have different impacts on the evolutionary process [Chong and Yao (2005)]. For example, increasingly higher levels of noise lead to mutual defection outcomes. Given a very noisy environment, strategies overspecialized to play defection only. This was not observed in the noiseless case of the IPD with increasingly more mutations. For example, increasingly higher mutation rates in the co-evolutionary learning system that used direct look-up table representation did not lead to mutual defection outcomes. Strategies were not observed to overspecialized to play defection, or any specific play.

3.3.3. *N-Player IPD*

Real-world interactions may involve more than two players. One famous example is the "tragedy of the commons" [Hardin (1968)], which illustrates the problem of self-interested actions of players for a particular public goods for initial rewards leading to a situation where everyone loses out in the end. For the case of the IPD, N-player interactions can be extended to the original formulation of two-player game [Axelrod and Dion (1988)]. This allows for the study of whether cooperative behaviors are possible when interactions involve more than two players since strategies that are effective for the two-player case may not be effective (or worse, fail) in large group interactions [Glance and Huberman (1994)].

One of us formulated an N-player IPD or NIPD game for investigations using the co-evolutionary learning approach [Yao and Darwen (1994)] (other studies include [Bankes (1994); Lindgren and Johansson (2001)]). The NIPD game is defined by the following three properties [Colman (1982)] (page 159):

- Each player faces two choices between cooperation and defection.

- Defection is dominant for each player, i.e., each player is better off defecting than cooperating regardless of how many other players that cooperate.
- The dominant defection strategies intersect in a deficit equilibrium. In particular, the outcome if all players choose their non-dominant cooperation strategies is preferable from every player's point of view to the one in which everyone chooses defection, but no one is motivated to deviate unilaterally from defection.

The payoff matrix (Fig. 3.8) for the NIPD game can then be constructed based on the following conditions that must be satisfied [Yao and Darwen (1994)]:

- $D_i > C_i$ for $0 \leq i \leq n-1$.
- $D_{i+1} > D_i$ and $C_{i+1} > C_i$ for $0 \leq i \leq n-1$.
- $C_i > (D_i + C_{i-1})/2$ for $0 \leq i \leq n-1$.

A large number values satisfy these conditions. For the study in [Yao and Darwen (1994)], the values are chosen such that if n_c is the number of cooperators in the NIPD game, then the payoff for cooperation is $2n_c - 2$ and the payoff for defection is $2n_c + 1$ (Fig. 3.9). For this payoff matrix, the average per-move payoff a can be calculated as follows if N_c cooperative moves are made out of N moves:

$$a = 1 + \frac{N_c}{N}(2n - 3),$$

which will allow the measurement of how common cooperation was by examining the average per-round payoff.

Number of cooperators among the remaining n-1 players

Player A		0	1	2		n-1
	C	C_0	C_1	C_2	...	C_{n-1}
	D	D_0	D_1	D_2	...	D_{n-1}

Fig. 3.8. The payoff matrix for the NIPD game. The value in the table gives the payoff to the player based on its choice of play [Yao and Darwen (1994)].

Number of cooperators among the remaining n-1 players

Player A		0	1	2		n-1
	C	*0*	*2*	*4*	...	*2(n-1)*
	D	*1*	*3*	*5*	...	*2(n-1)+1*

Fig. 3.9. An example of the payoff matrix for the NIPD game [Yao and Darwen (1994)].

NIPD game interactions were in the form of a large number of random selection of groups of N players with replacement (e.g., 1000 NIPD games for a population of 100 strategies). Results from the experiments in [Yao and Darwen (1994)] showed the group size (i.e., the value of N in the NIPD game) has a negative impact on the evolution of cooperation. As N increases, there are fewer number of runs where the population evolved to play cooperation. For example, in the case of memory two strategies, only one out of 20 runs had defection outcomes for 3IPD. However, the number of runs with defection outcomes increased to nine for 6IPD. Increasing N to 16 (i.e., 16IPD) resulted with all runs evolved to defection outcomes [Yao and Darwen (1994)].

3.3.4. *Other Extensions*

There are many other extensions to the classical IPD game, or even further extensions to already extended IPD games (such as the NIPD) that can be studied through a co-evolutionary learning approach. For example, we examined the impact of localized interactions of the NIPD games in [Seo *et al.* (1999, 2000)]. The earlier study for the NIPD [Yao and Darwen (1994)] showed that the evolution of cooperation is more difficult to achieve through a co-evolutionary learning process as N increases. However, in some real-world interactions, it is unlikely that a player interacts with everybody (or that it has equal probability of interacting with anyone in the population). Instead, a player might interact with other specific players (e.g., neighbours, relatives, or at the workplace). Such localized interactions may involve spatial models [Nowak and May (1992); Ishibuchi and Namikawa (2005)]. In particular, localized interactions can have a positive impact on the evolution of cooperation in the NIPD game. That is, population structured in a spatial model is more likely to evolve cooperation [Seo *et al.* (1999); Lindgren and Johansson (2001)].

Another extension that can be considered is to incorporate indirect interactions to the IPD game that originally only considers direct interactions between strategies. Most of the previous studies have focused on modelling direct interactions (e.g., cooperative behaviors through direct reciprocity that involves repeated encounters, i.e., IPD games [Axelrod (1984)]) or indirect interactions (e.g., cooperative behaviors through mechanisms of indirect reciprocity such as reputation where an individual receives cooperation from third parties due to the individual's cooperative behaviors to others in the case of indirect reciprocity [Nowak and Sigmund (1998b)]). However, it has been suggested that complex real-world interactions involve both direct and indirect interations (although for simplicity for modelling and analysis, only one of the interactions is considered at one time) [Nowak and Sigmund (1998a)]. For this aspect, we have investigated a model with both direct and indirect interactions [Yao and Darwen (1999)]. In particular, each strategy is tagged with a reputation score, which is calculated based on payoffs received from a small random sample of pre-games. A co-evolutionary approach to show that with the addition of reputation, cooperative outcomes are possible and more likely even for the case of the IPD with more choices and shorter game durations [Yao and Darwen (1999)].

In addition to that, another extension will be to consider the adaptation of payoff matrices. We recently conducted a preliminary study on evolving strategy payoff matrices, and how such an adaptation process can affect the learning of strategy behaviors [Chong and Yao (2006)]. The motivation for the study is to relax the assumption of having fixed, symmetric payoff matrix for all evolving strategies. This assumption may not be realistic, considering that not all players are similar in real-world interactions. We focus specifically on an adaptation process of payoff matrix based on past behavioral interactions. In particular, a simple update rule that provides a reinforcement feedback process between strategy behaviors and payoff matrices during the co-evolutionary process is used. Results from experiments [Chong and Yao (2006)] showed that the evolutionary outcome is dependent on the adaptation process of both behaviors (i.e., strategy behavioral responses) and utility expectations that determine how behaviors are rewarded (i.e., strategy payoff matrices). Defection outcomes are more likely to be obtained if IPD-like update rules that favor the exploitation of opponents are used. However, cooperative outcomes can be easily obtained when mutualism-like update rules that favor mutual cooperation are used.

3.4. Conclusion and Future Directions

The greatest advantage and the most important feature of co-evolutionary learning is that of the process of adaptation on representation that is dependent on the interactions between members of the population. In this aspect, the co-evolutionary learning approach is well-suited to solving the problem of IPD games in two contexts. First, co-evolutionary learning can be used as a search algorithm for effective strategies without requiring human knowledge. All that is required is the rules of the game. Second, the adaptation process of strategy behaviors based on interactions in co-evolution provides a natural way to investigate conditions that lead to the evolution of certain behaviors. In both of these contexts, the advantage of co-evolutionary learning to other approaches is that strategy behaviors are not fixed or predefined. Instead, co-evolutionary learning provides a means to realize strategy behavioral responses that are not necessarily bounded by expert human knowledge, thus providing new insight to the problem.

Since the first study of co-evolutionary learning on the classical IPD by Axelrod [Axelrod (1987)], there had been a wide-range of studies that further extended the classical IPD game with additional features such as, but not limited to, continuous or multiple levels of cooperation, noisy interactions, N-player interactions, spatial interactions, and indirect interactions. The motivation in all of these studies is to bridge the gap between the abstract IPD interactions with the complex real-world interactions. As such, by understanding the specific conditions that lead to the evolution of specific IPD strategy behaviors, these studies have further helped to provide a more in-depth view on complex real-world interactions such as those found in the human society.

There are still much more that can be explored using the co-evolutionary learning approach. One direction will be to further extend the more complex IPD games and investigate the impact of the additional extension. This is important because the extensions might interact with one another in some unknown and nonlinear fashion. Understanding these interactions will help to further unravel complex human interactions. Another direction will be to investigate a more rigorous approach to determine the robustness of evolved strategy behaviors. In this particular aspect, the notion of generalization might provide a more natural approach for co-evolutionary learning in addition to classical evolutionary game theory approach of the evolutionarily stable strategies.

References

Atmar, W. (1994). Notes on the simulation of evolution, *IEEE Transactions on Neural Networks* **5**, 1, pp. 130–147.

Axelrod, R. (1980a). Effective choice in the prisoner's dilemma, *The Journal of Conflict Resolution* **24**, 1, pp. 3–25.

Axelrod, R. (1980b). More effective choice in the prisoner's dilemma, *The Journal of Conflict Resolution* **24**, 3, pp. 379–403.

Axelrod, R. (1984). *The Evolution of Cooperation* (Basic Books, New York).

Axelrod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma, in L. D. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, chap. 3 (Morgan Kaufmann, New York), pp. 32–41.

Axelrod, R. and Dion, D. (1988). The further evolution of cooperation, *Science* **242**, 4884, pp. 1385–1390.

Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation, *Science* **211**, pp. 1390–1396.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice* (Oxford University Press, New York).

Bäck, T., Hammel, U. and Schwefel, H. P. (1997). Evolutionary computation: Comments on the history and current state, *IEEE Transactions on Evolutionary Computation* **1**, 1, pp. 3–17.

Bankes, S. (1994). Exploring the foundations of artificial societies: Experiments in evolving solutions to iterated n-player prisoner's dilemma, in R. Brookes and P. Maes (eds.), *Artificial Life IV* (Addison-Wesley), pp. 337–342.

Chellapilla, K. and Fogel, D. B. (1999). Evolution, neural networks, games, and intelligence, *Proc. IEEE* **87**, 9, pp. 1471–1496.

Chong, S. Y. and Yao, X. (2005). Behavioral diversity, choices, and noise in the iterated prisoner's dilemma, *IEEE Transactions on Evolutionary Computation* **9**, 6, pp. 540–551.

Chong, S. Y. and Yao, X. (2006). Self-adaptive payoff matrices in repeated interactions, in *2006 IEEE Symposium on Computational Intelligence and Games (CIG'06)* (IEEE Press, Piscataway, NJ), pp. 103–110.

Colman, A. M. (1982). *Game Theory and Experimental Games* (Pergamon Press, Oxford).

Darwen, P. and Yao, X. (1995). On evolving robust strategies for iterated prisoner's dilemma, in *Progress in Evolutionary Computation*, *Lecture Notes in Artificial Intelligence*, Vol. 956, pp. 276–292.

Darwen, P. and Yao, X. (2000). Does extra genetic diversity maintain escalation in a co-evolutionary arms race, *International Journal of Knowledge-Based Intelligent Engineering Systems* **4**, 3, pp. 191–200.

Darwen, P. and Yao, X. (2001). Why more choices cause less cooperation in iterated prisoner's dilemma, in *Proc. 2001 Congress on Evolutionary Computation (CEC'01)* (IEEE Press, Piscataway, NJ), pp. 987–994.

Darwen, P. and Yao, X. (2002). Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense, *International Journal of Computational Intelligence and Applications* **2**, 1, pp.

83–107.

Darwen, P. J. (1996). *Co-evolutionary Learning by Automatic Modularization with Speciation*, Ph.D. thesis, University of New South Wales, Sydney, Australia.

Darwen, P. J. and Yao, X. (1997). Speciation as automatic categorical modularization, *IEEE Transactions on Evolutionary Computation* **1**, 2, pp. 101–108.

Fogel, D. B. (1991). The evolution of intelligent decision making in gaming, *Cybernetics and Systems: An International Journal* **22**, pp. 223–236.

Fogel, D. B. (1993). Evolving behaviors in the iterated prisoner's dilemma, *Evolutionary Computation* **1**, 1, pp. 77–97.

Fogel, D. B. (1994a). An introduction to simulated evolutionary optimization, *IEEE Transactions on Neural Networks* **5**, 1, pp. 3–14.

Fogel, D. B. (1994b). An introduction to simulated evolutionary optimization, *IEEE Transactions on Neural Networks* **5**, 1, pp. 3–14.

Fogel, D. B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (IEEE Press, Piscataway, NJ).

Fogel, D. B. (1996). On the relationship between the duration of an encouter and the evolution of cooperation in the iterated prisoner's dilemma, *Evolutionary Computation* **3**, 3, pp. 349–363.

Franken, N. and Engelbrecht, A. P. (2005). Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma, *IEEE Transactions on Evolutionary Computation* **9**, 6, pp. 562–579.

Glance, N. S. and Huberman, B. A. (1994). The dynamics of social dilemmas, *Scientific American* , pp. 58–63.

Hardin, G. (1968). The tragedy of the commons, *Science* **162**, pp. 1243–1248.

Harrald, P. G. and Fogel, D. B. (1996). Evolving continuous behaviors in the iterated prisoner's dilemma, *BioSystems: Special Issue on the Prisoner's Dilemma* **37**, pp. 135–145.

Ishibuchi, H. and Namikawa, N. (2005). Evolution of iterated prisoner's dilemma game strategies in structured demes under random pairing in game playing, *IEEE Transactions on Evolutionary Computation* **9**, 6, pp. 552–561.

Julstrom, B. A. (1997). Effects of contest length and noise on reciprocal altruism, cooperation, and payoffs in the iterated prisoner's dilemma, in *Proc. 7th International Conf. on Genetic Algorithms (ICGA'97)* (Morgan Kauffman, San Francisco, CA), pp. 386–392.

Lindgren, K. (1991). Evolutionary phenomena in simple dynamics, in C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen (eds.), *Artificial Life II* (Addison-Wesley), pp. 295–312.

Lindgren, K. and Johansson, J. (2001). Coevolution of strategies in n-person prisoner's dilemma, in J. Crutchfield and P. Schuster (eds.), *Evolutionary Dynamics - Exploring the Interplay of Selection, Neutrality, Accident, and Function* (Addison-Wesley).

Mcnamara, J. M., Barta, Z. and Houston, A. I. (2004). Variation in behaviour promotes cooperation in the prisoner's dilemma, *Nature* **428**, pp. 745–748.

Miller, J. (1989). The coevolution of automata in the iterated prisoner's dilemma, Tech. Rep. 89-003, Santa Fe Institute Report.

Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos, *Nature* **355**, pp. 250–253.

Nowak, M. A. and Sigmund, K. (1998a). The dynamics of indirect reciprocity, *Journal of Theoretical Biology* **194**, pp. 561–574.

Nowak, M. A. and Sigmund, K. (1998b). Evolution of indirect reciprocity by image scoring, *Nature* **393**, pp. 573–577.

Seo, Y. G., Cho, S. B. and Yao, X. (1999). Emergence of cooperative coalition in nipd game with localization of interaction and learning, in *Proc. IEEE 1999 Congress on Evolutionary Computation (CEC'99)* (IEEE Press, Piscataway, NJ), pp. 877–884.

Seo, Y. G., Cho, S. B. and Yao, X. (2000). Exploiting coalition in co-evolutionary learning, in *Proc. IEEE 2000 Congress on Evolutionary Computation (CEC'00)* (IEEE Press, Piscataway, NJ), pp. 1268–1275.

Stanley, E. A., Ashlock, D. and Smucker, M. D. (1995). Prisoner's dilemma with choice and refusal of partners: Evolutionary results, in *Proc. Third European Conf. on Advances in Artificial Life*, pp. 490–502.

Wu, J. and Axelrod, R. (1995). How to cope with noise in the iterated prisoner's dilemma, *The Journal of Conflict Resolution* **39**, 1, pp. 183–189.

Yao, X. (1994). Introduction, *Informatica (Special Issue on Evolutionary Computation)* **18**, pp. 375–376.

Yao, X. (1999). Evolving artificial neural networks, *Proc. IEEE* **87**, 9, pp. 1423–1447.

Yao, X. and Darwen, P. (1999). How important is your reputation in a multi-agent environment, in *Proc. 1999 Conf. on Systems, Man, and Cybernetics (SMC'99)* (IEEE Press, Piscataway, NJ), pp. 575–580.

Yao, X. and Darwen, P. J. (1994). An experimental study of n-person iterated prisoner's dilemma games, *Informatica* **18**, pp. 435–450.

Yao, X., Liu, Y. and Darwen, P. J. (1996). How to make best use of evolutionary learning, in R. Stocker, H. Jelinck, B. Burnota and T. Bossomaier (eds.), *Complex Systems - From Local Interactions to Global Phenomena* (IOS Press, Amsterdam), pp. 229–242.

Yao, X., Liu, Y. and Lin, G. (1999). Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation* **3**, 2, pp. 82–102.