# Simulation of Self-Control through Precommitment Behaviour in an Evolutionary System

**Gaye Delphine Banfield**

**A thesis submitted to the**
**University of London**
**for the Degree of**
**Doctor of Philosophy**

**School of Computer Science**
**and Information Systems,**
**Birkbeck College,**
**University of London**

**Year 2006**

**Abstract**

The purpose of this thesis is to determine how evolution has resulted in self-control through precommitment behaviour. Empirical data in psychology suggest that we recognize we have self-control problems and attempt to overcome them by exercising precommmitment, which bias our future choices to a larger, later reward. The behavioral model of self-control as an internal process is taken from psychology and implemented, using a top-down approach, as a computational model of the human brain. This is a novel approach to modeling the brain, but is appropriate given the complexity of the behaviour. The higher and lower brain systems, represented by two Artificial Neural Networks (ANNs) using reinforcement learning, are viewed as cooperating for the benefit of the organism. This is a departure from the classical view of the higher brain, associated with planning and control, overriding the lower brain. The ANNs are implemented as two players, learning simultaneously, but independently, competing in games, where the payoffs are neither wholly adverse nor wholly competitive. This departs from the traditional framework for multi-agent reinforcement learning, removing the limitation of centralized learning and widening the scope for the learner's behaviour. Psychological studies suggest that the structure of the self-control problem can be likened to the Iterated Prisoner's Dilemma game in that cooperation is to defection what self-control is to impulsiveness. In this thesis it is proposed that increasing precommitment increases the probability of cooperating with oneself in the future. To this aim, a bias towards future rewards is implemented. The results suggest that this bias enhances

cooperation, which could be interpreted as precommitment. The model is then subjected to simulation of evolutionary adaptation using genetic algorithms. The results show an evolutionary basis for this complex behaviour and suggest that evolutionary factors, as opposed to learning alone, play a crucial role in its formation.

*It's the way a man chooses to limit himself that determines his character.*

Luke Rhinehart, The **Dice Man**

**Acknowledgements**

Many people helped enormously in writing this thesis and in its very existence. A special debt is owed to Dr. Chris Christodoulou for his personal support and confidence in this research. Many thanks to Dr. Peter Sozou, whose ideas and suggestions have been a constant source of guidance and inspiration. I would like to thank Dr. Tom Westerdale who has been on hand to lead me through the process of writing, which has not been the easiest of tasks. I would also like to thank my family and close friends who have been, and continue to be an oasis of support. Getting here would not have been possible without the encouragement of my parents and for that I am forever grateful. Last and certainly not least, my thanks to Al for all his efforts, who I am sure is the happiest to see this finished, and most importantly, I am indebted to my son Fred who, for too long, has had to cope with a harridan as a mum, for his unconditional love I thank him.

## Contents

## List of Figures

## List of Symbols

| Symbol | |
|---|---|
| $\sum x_j$ | The sum $x_1 + x_2 + \ldots + x_n$ |
| $\alpha$ | Step-size in TD learning |
| $\gamma$ | Discount rate in TD learning |
| $\lambda$ | Decay rate for eligibility trace e.g. $\lambda=0$ means only that one state preceding the current one is changed by the TD error. |
| $\delta$ | Temporal difference error |
| $\psi$ | Differential bias |
| $d$ | The delay of the reward |
| $k$ | The degree of discounting of future rewards, i.e., $k=0$ implies there is no discounting |
| $v$ | The current discounted value of the reward |
| $V$ | The undiscounted value of the reward |
| $\eta$ | Learning rate in the Selective Bootstrap learning rule |

**List of Abbreviations**

| | |
|---|---|
| ANN | Artificial Neural Network |
| CR | Conditioned Response |
| CS | Conditioned Stimulus |
| DP | Dynamic Programming |
| EA | Evolutionary Algorithms |
| EARL | Evolutionary Algorithms for Reinforcement Learning |
| EP | Evolutionary Programming |
| ES | Evolutionary Strategies |
| EVR | The patient referred to as Elliot by Damasio (1994) |
| FFWD | Feed Forward |
| GA | Genetic Algorithms |
| IPD | Iterated Prisoner's Dilemma |
| LMP | Local Minimum Problem |
| LMS | Least-Mean-Square weight update rule otherwise known as the Widrow-Hoff rule after its authors Widrow and Hoff (1960) |
| MARL | Multi-Agent Reinforcement Learning |
| MAS | Multi-Agent System |
| MDP | Markov Decision Process |
| MRI | Magnetic Resonance Imaging |
| PD | Prisoner's Dilemma |
| RBG | Rubinstein's Bargaining Game |
| RL | Reinforcement Learning |

| | |
|---|---|
| SANE | Symbiotic Adaptive Neuro-Evolution (Moriarty and Mikkulainen, 1996) |
| SARL | Single Agent Reinforcement Learning |
| SRN | Simple Recurrent Network |
| TD | Temporal Difference Learning |
| TE | Trial and Error learning |
| UR | Unconditioned Response |
| US | Unconditioned Stimulus |

# Chapter 1

## 1 Introduction

### 1.1 Overview

In this thesis we investigate how evolution has resulted in self-control such that people must use precommitment behaviour to control their future actions. We begin by attempting to gain a greater understanding of self-control through precommitment through simulation of the brain as a functionally decomposed parallel system. Unlike a serial system, a functionally decomposed parallel system can have internal conflicts, as different processes exhibit different behaviours. Self-control through precommitment behaviour is an example of such an internal conflict. People exercise self-control when they choose a larger, but later reward over a smaller, but sooner reward. Precommitment is a mechanism for doing this, by making a choice now that will make it impossible, or at least difficult to change our minds later, and if we do change our minds the change is costly. If individuals were fully rational, precommitment would be unnecessary, as any later temptation that would jeopardise their true preference would be rejected (Nesse, 2001).

The research begins by simulating self-control through precommitment behaviour in a computational model, the architecture of which comprises of two ANNs, one representing the higher brain functions and the other the lower brain functions. The higher brain functions are associated with rational thought and the lower brain functions are associated with instinctive behaviour. The model is representative of the higher *versus* lower brain

functions. This is based on the theoretical premise that the human brain is a modular system, and that the higher and the lower systems of the brain are largely independent and are competing for control of the organism. This thesis presents a novel view of the higher and lower brain regions cooperating, i.e., working together, as opposed to the classical view of the higher brain as the controller overriding the lower brain. In this thesis the new model of the higher and lower brains cooperating is referred to as the *Cooperating Model* and the traditional view of the higher brain assuming the role of controller is referred to as the *Control Model*. In the Cooperating Model of this thesis (presented in Chapter 3), the higher and lower brain functions have to learn to collaborate, leading to cooperation. In the traditional Control model the higher brain system, associated with planning and control, overrides the lower brain system. In the Cooperating model the higher and lower brain systems work together for the benefit of the organism. In this thesis, the word "cooperation" has a non-conventional meaning: only cooperate for the larger later reward. Cooperation as defined in the Oxford dictionary, is to "work together for a common end" in which case if both the higher and lower brain defect, i.e., go for the smaller sooner reward, this could be viewed as cooperation. In this thesis this is not exactly the case; here cooperation means cooperating in order to gain the larger later payoff, hence the situation where both players defect is not seen as cooperation.

This model of the neural cognitive system of self-control through precommitment behaviour presented in Chapter 3 is simple, but also sufficiently detailed to explain in computational terms how the brain generates

the apparently inconsistent behaviour of self-control through precommitment. The model encompasses a cognitive architecture that provides a general explanation of self-control. The ANN representing the higher brain region is implemented with a weight update rule that simulates far-sightedness, i.e., placing greater value on future rewards. Similarly the ANN representing the lower brain region is implemented with a weight update rule that simulates myopia. This thesis is concerned with the evolutionary basis of self-control through precommitment behaviour, but not to the exclusion of learning. For example what is the role of learning and the effect of learning on this complex behaviour? For this reason the model also explores the role of learning during an individual's lifetime on self-control behaviour, by incorporating a continuous learning process and by interacting with the environment through its sensors, i.e., it has minimal pre-programmed knowledge. Learning from interaction with the environment is the fundamental idea underlying reinforcement learning and for this reason an ANN with reinforcement learning is used. In this thesis, a number of experiments are conducted to examine what interdependencies exist, and what internal conflicts exist in this functionally decomposed neural system. The model takes into consideration the complexity of the environment. The variables defining the ANN are parameterised to enable control of the model. These include the form of learning, the learning rate and the number of hidden neurons in each ANN. The model is compared to other models of related behaviour, for example, Carver's and Scheier's model of self-regulation (1998).

The resulting model is developed and tested by playing games, in particular games that have a real world application. For example, pollution can be regarded as a game of Prisoner's Dilemma (Hamburger, 1979); war can be regarded as another dilemma game called *Chicken* (Binmore, 1992), where two players compete for a piece of territory. If one player chickens out he looses, if both players chicken out the situation remains the same, and if neither player chickens out the consequences are unpleasant for both. The two artificial neural networks, representing the higher and lower centres of the brain, compete against each other in two general-sum games: the Rubinstein's Bargaining game (Rubinstein, 1982), and the Iterated Prisoner's Dilemma game (Axelrod and Hamilton, 1981). A general-sum game is where the players' payoff are neither totally positively correlated nor totally negatively correlated (refer to section 2.3 for a more detailed explanation). The Rubinstein's Bargaining Game (RBG) is viewed as a simple general-sum game and has many of the characteristics of the self-control problem, i.e., discounting of future rewards, myopic *versus* far-sighted behaviour, and learning to cooperate with the other player. For these reasons the RBG is an appropriate game to use in the early stages of developing and testing the model. The Iterated Prisoner's Dilemma (IPD) game is appropriate since it has been suggested there is a relationship between self-control and social cooperation (Brown and Rachlin, 1999). Brown and Rachlin (1999) used a version of the IPD game in experiments on self-control with human subjects. The empirical results of these experiments are used to validate the neural model. The effect of reward and punishment on the behaviour of the two neural networks is observed in various game-theoretical situations. The results

of these tournaments are compared to the available data in psychology and economics on how people play games.

In the final stage of this thesis the two artificial neural network system undergoes evolutionary adaptation. The parameters defining the networks are subjected to simulated genetic evolution using genetic algorithms (Holland, 1992). The aim of this phase of the study is to find an explanation for the way self-control through precommitment evolved. The investigation focuses on the functional decomposition of the brain and attempts to explain such behaviour as a by-product of some internal mechanism. The role of reinforcement and reinforcement history is also examined to explain variances in an individual's behaviour. Finally, the simulation explores the theory that such behaviour is adaptive. Following on from Sozou (2003), possible evolutionary explanations for the existence of self-control through precommitment behaviour that are explored in this thesis are:

1. The behaviour results from an internal conflict, between the higher and lower centres of the brain, which may be due to: (i) the animal evolving optimal low-level (i.e., instinctive) behaviours in response to certain cues in an ancestral environment; (ii) the animal being moved to a novel environment where these low-level behaviours are inappropriate to its higher goals; (iii) the animal learning cognitively in the "higher" part of the brain that the low-level behaviours are inappropriate; (iv) the animal trying to devise a way in the higher centre of the brain to bypass the low-level behaviours. As such, standard psychological self-control problems can be understood as part of a spectrum of phenomena involving

overcoming behaviours, which cognition can directly control only partially or not at all. In this explanation the theoretical premise is made that the functions associated with the higher brain system (i.e., rational thought) and the functions associated with the lower brain system (i.e., instinctive behaviour), are locked in some form of internal conflict for control of the organism and therefore its behaviour.

2. The behaviour is a side effect of the evolution of commitment mechanisms for game-theoretical situations where precommitment is useful, e.g., anger and self-deception (Nesse, 2001).

3. The behaviour results from a best evolutionary compromise to environmental complexity and variability. It is not feasible for evolution to program the brain with a direct hard-wired response to every situation it could meet. Instead, there is goal-directed behaviour and a capacity for learning. However, these goals cannot perfectly correspond to fitness. Hence, natural selection has allowed low-level behaviours to effectively take control when cues are strong enough to reliably reflect fitness consequences. This gives rise to the multiple personalities theory; for example, the person that wakes up in the morning is different from the person who went to sleep the previous night. This theory is supported by Trivers (2000) in the evolution of self-deception and Samuelson and Swinkels (2002).

These explanations are neither mutually exclusive nor exhaustive. In summary, the model aims to answer the question whether self-control through

precommitment behaviour is (i) an internal conflict, (ii) a biological by-product, or (iii) an adaptation to enhance the survival of the species.

Although there has been much research in psychology and economics in the area of self-control, to the best of our knowledge this is the first time that self-control through precommitment behaviour is simulated in a computational neural model, in a competitive interaction. For verification of the model, the results from the simulation are compared with the empirical data of self-control and precommitment in psychology and economics, and are found to compare favourably. This contributes to bridging the gap between the modelling community and the experimentalists. In addition, the extent to which reinforcement learning can be realised in games that model real life situations is of considerable interest to game theorists and economists and it is currently the subject of intense research activity (Kaebling et al., 1996; Littman, 2001; Shoham et al., 2003).

## 1.2 Outline of this Thesis

The thesis is divided in to two parts. The first part (Chapters 2 – 5) provides the groundwork for the latter. It begins with a critical literature review and analysis of the ideas for the neural modeling of the behaviour self-control through precommitment. The computational model is then presented and compared with alternative models. Finally this section concludes with the principles relevant for the evolutionary adaptation of the resulting neural model. The second part of the thesis (Chapters 6 - 8) is concerned with the development of the evolutionary system through simulations and empirical analysis. The first phase is concerned with building and testing of the neural

model. In the final phase the resulting neural model is subject to evolutionary adaptation using Genetic Algorithms. Table 1.1 lists the experiments that were carried out and the motivation for doing that experiment in order of appearance. In summary, experiments 1 to 5 are concerned with developing and testing the neural model of Figure 3.3, experiments 6 to 8 investigate the effect of implementing a bias towards future rewards on the behaviour of the neural model and experiments 9 and 10 subject the neural model to evolutionary adaptation through Genetic Algorithms.

| Ref. | Description | Scenario of Simulation | Section |
|---|---|---|---|
| 1. | Selective Bootstrap feed forward network (SB-FFWD) playing an Artificial Opponent in the RBG | Simulation of the behaviours associated with the lower brain functions modeled as a feed forward MLP network implemented with the Selective Bootstrap weight update rule | 6.4.1 |
| 2. | Temporal Difference feed forward network (TD-FFWD) playing an Artificial Opponent in the RBG | Simulation of the behaviours associated with the higher brain functions modeled as a feed forward MLP network implemented with the Temporal Difference weight update rule | 6.4.2 |
| 3. | 2-ANNs (TD-FFWD vs. SB-FFWD) playing the RBG | Simulation of the novel computational model of self-control presented in Section 3.3 of the higher *versus* lower brain in a game theoretical situation | 6.4.3 |
| 4. | 2-ANNs (TD-FFWD vs. SB-FFWD) playing the IPD game with local rewards | Simulation of the computational model of self-control presented in Section 3.3 with both networks implemented as feed forward MLP networks with the novel view of the higher and lower brain cooperating, i.e., working together as opposed to the classical view of the higher brain as the controller overriding the lower brain. In this case each ANN receives an individual reward | 6.5.1 |
| 5 | 2-ANNs (TD-FFWD vs. SB-FFWD) playing the IPD game with global rewards | Simulation of the computational model of self-control presented in Section 3.3 with both ANNs implemented as feed forward MLP networks with the novel view of the higher and lower brain cooperating, i.e., working together as opposed to the classical view of the higher brain as the controller overriding the lower brain where | 6.5.2 |

| | | | |
|---|---|---|---|
| | | both ANNs receive the same reward | |
| 6 | 2-ANNs (TD-FFWD vs. SB-FFWD) playing the IPD game with bias towards future reward implemented by varying the bias to the ANN between a value between 0 and 1. This is referred to as a *variable bias*. | Simulation of the computational model of self-control as in experiment 4, but with bias towards future rewards implemented as a variable bias in place of the ANN's bias | 6.6.1 |
| 7 | 2-ANNs (TD-FFWD vs. SB-FFWD) playing the IPD game with a bias towards future reward implemented as an *extra node* to the ANN | Simulation of the computational model of self-control as in experiment 5, but with bias towards future rewards added. Modeled as an extra input to the ANN | 6.6.2 |
| 8 | 2-ANNs (TD-FFWD vs. SB-FFWD) playing the IPD game with a bias towards future reward implemented as a *differential bias* applied to the payoff matrix | Simulation of the computational model of self-control as in experiment 5, but with a *differential bias* added to the global reward payoff matrix as differential bias | 6.6.3 |
| 9 | Evolutionary adaptation of the 2-ANNs model with a bias towards future rewards modeled as the *differential bias* from experiment 8 | Simulation of the computational model implemented as in experiment 8 subject to evolutionary adaptation of the bias through Genetic Algorithms to investigate what value for the bias works best when | 7.2 |
| 10 | Evolutionary adaptation of the 2-ANNs model with bias towards future rewards modeled as a *differential bias* and with evolution of the learning parameters | Simulation of the computational model implemented as in experiment 8 subject to evolutionary adaptation of the number of hidden nodes and learning rules for the best bias | 7.3 |

**Figure 1.1 List of experiments carried out in this thesis in order of appearance.** Experiments 1 to 5 are concerned with developing and testing the neural model of Figure 3.3. Experiments 6 to 8 investigate the effect of implementing a bias towards future rewards on the behaviour of the 2-ANNs model. Experiments 9 and 10 subject the neural model to evolutionary adaptation through Genetic Algorithms.

The remainder of this thesis is organized as follows. **Chapter 2** - *Literature Review of Self-Control and Games* reviews the literature on self-control through precommitment behaviour from the perspective of psychology and economics. It then examines self-control through precommitment in the context of games, specifically games that model real-world applications. It also includes a critical review of the literature on games with no clear winner (general-sum games) and concludes by examining the role of nature and

learning in self-control through precommitment behaviour. **Chapter 3** - *Review of the Concepts for the Neural Modelling of Self-control through Precommitment* begins by a review and analysis of relevant neural models, i.e., models of related behaviours. A cognitive model of self-control through precommitment is presented with an explanation of how this translates into a computational neural model. **Chapter 4** - *Review of Reinforcement Learning in the context of Self-Control* introduces reinforcement learning in the context of self-control and reviews the literature on reinforcement learning methods including a description of reinforcement learning as applied to Artificial Neural Networks. The chapter concludes by examining what role reinforcement learning and Artificial Neural Networks play in this thesis. **Chapter 5** - *Review of the Concepts for Evolutionary Adaptation of the Neural Model* starts with a critical review of the main evolutionary computation techniques of Genetic Algorithms, Evolutionary Programming and Evolutionary Strategies. The chapter concludes with an explanation why Genetic Algorithms are the evolutionary computation technique of choice for this thesis. In **Chapter 6** - *Explaining Self-Control by Playing Games* the computational neural model is developed and tested by running experiments using general-sum games, which model real-world situations. In the first set of experiments the general-sum game Rubinstein's Bargaining game (1982) is used. In the second set of experiments the results of the Rubinstein's Bargaining Game are verified in a similar neural architecture with the Iterated Prisoner's Dilemma game. The results are presented and discussed. **Chapter 7** - *Evolutionary Adaptation of the Neural Model* is concerned with the design and testing of the evolutionary system. It concludes with a presentation of the

results and discussion. **Chapter 8** - *Can Self-Control through Precommitment be explained by evolutionary game theory?* critically examines the work of this thesis and, in particular, it considers whether a plausible explanation for the evolution of self-control through precommitment behaviour has been found. It discusses the contribution of this thesis to computational neuroscience and related fields most notably reinforcement learning, evolutionary computation and game theory. The chapter concludes with possible future directions for this work.

# Chapter 2

# 2 Literature Review of Self-Control and Games

## 2.1 Chapter Outline

This chapter builds the theoretical foundation on which this thesis is based. It starts with a review of the literature on self-control through precommitment from psychology, economics and neuroscience. It then discusses self-control in the context of games specifically the Iterated Prisoner's Dilemma's game. A rudimentary review of the necessary game theory concepts is given. The chapter concludes by examining the question whether we are born with the capacity to precommit or whether we learn self-control through precommitment as part of socialization?

## 2.2 Self-Control through Precommitment: the problem and its importance

### 2.2.1 Defining self-control behaviour

Self-control arises out of a desire to control one's behaviour. Taken literally self-control is to control one's self, where self, as defined by the Oxford Dictionary, is one's "nature and individuality", which is displayed to the world by our behaviour (actions). Behaviour is just one aspect of the human situation. We have feelings, skills, and conscious and unconscious thoughts, but whatever thoughts or feelings we have, it is on our behaviour that we are judged. In psychology, to exercise self-control is to inhibit an impulse to engage in a behaviour that violates a moral standard (Morgan et al., 1979). Problems in exercising self-control occur when there is a lack of willingness

or motivation to carry out this inhibition. This suggests a cognitive *versus* a motivational conflict. Problems in exercising self-control suggest a motivational problem: we know what is good for us (cognition), but we do not do it (motivation). There is often a discrepancy between our verbal and non-verbal behaviour. The distinction between cognition and motivation has been likened to the distinction between the higher and lower brain functions. This suggests that self-control involves a conflict between cognition and motivation (Rachlin, 1995), a far-sighted planner and a myopic doer (Shefrin and Thaler, 1981), reason and passion, and is not just a case of changing tastes. Self-control problems stem from a conflict at any single point in time of the choices we have available now and our future choices, and occur because our preferences for available choices are inconsistent across time (Ainslie, 1975; Loewenstein, 1996).

Figure 2.1 illustrates the view of self-control, where there is a choice between a smaller-sooner (*SS*) reward, available at time $t_2$ and a larger-later (*LL*) reward, available at $t_3$. The lines of the graph in Figure 2.1 show the *discounted value functions* over time for the rewards *SS* and *LL*. Studies in self-control have found that increasing the delay of the reward, referred to as the *delay of gratification*, decreases the discounted value of the reward (Mischel et al., 1989). *Delay of gratification* is defined in psychology as waiting for a more appropriate time and place to gain a reward. In Figure 2.1 at time $t_1$ the discounted value of *LL* is greater than the discounted value of *SS,* so the person prefers the reward *LL* over the reward SS. At a later time just before $t_2$, when the reward *SS* is imminent, the discounted value of *SS* is

greater than the discounted value of *LL*, so the person prefers the reward *SS* over the reward *LL* thus we have *reversal of preferences*.

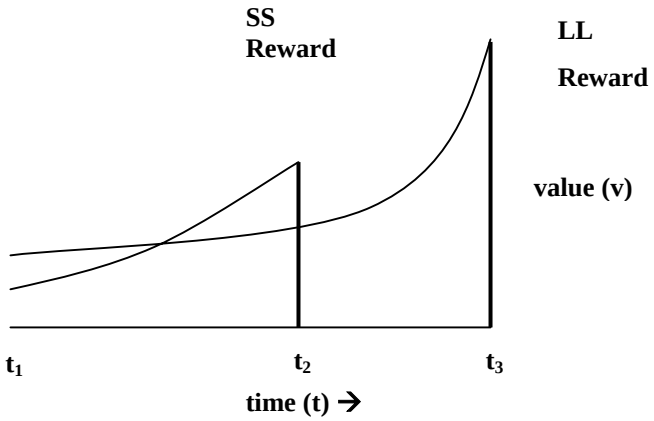



**Figure 2.1 Illustration of Self-Control as a choice between a smaller, sooner reward and a larger, later reward.**
Illustration of self-control defined as the choice of a larger, but later reward (*LL*) available at $t_3$ over a sooner, but smaller reward (*SS*) at time $t_2$. The lines represent the discounted value function of the reward indicating how these rewards decrease in value with increasing delay. At an earlier time $t_1$ the value of the larger-later reward exceeds that of the smaller-sooner reward. The crossing of the lines indicates a reversal of preferences when the value of the reward *SS* overtakes the value of the reward *LL* (adapted from Rachlin, 1995).

In self-control problems the conflict arises out of this reversal of preferences between those choices available immediately (*SS*) and those available at some time later (*LL*). The crossing of the discounted value functions for *LL* and *SS* depicts this reversal of preferences. Reversal of preferences is seen in experiments on human subjects (Solnick et al., 1980; Millar and Navarick 1984). Note that the lines of the graph get steeper as the time approaches the time of the reward (Rachlin, personal communication, 2003). The curves of the graphs in Figure 2.1 have roughly the shape of a hyperbolic function (Eq. 2.1) and also roughly the shape of an exponential function (Eq. 2.2). Eq. 2.1

and Eq. 2.2 give the current discounted value *v*, which is the height of the graph at any point in time:

$$v = \frac{V}{(1 + kd)} \qquad (2.1)$$

$$v = Ve^{-kd} \qquad (2.2)$$

In these equations *V* is the undiscounted value of the reward (the height of the lines *SS* or *LL*), *d* is the delay of the reward (the difference between the time of reward and the time now), and *k* is the discount rate representing the degree of discounting of future rewards, (for example, *k=0* implies there is no discounting of future rewards). Two hyperbolic discount functions with the same discount rate (*k*) may cross, but two exponential discount functions with the same discount rate cannot cross. As Mazur (1987) suggests, assuming that a person uses the same discounted value function (with the same discount rate) for all the person's rewards, if the discounted value function is exponential then there can be no reversal of preferences, but if it is hyperbolic then there can be.

Rubinstein (2003) offers an alternative to hyperbolic discounting for the explanation of the crossing of the discounted value functions. Rubinstein's method (2003) is procedural. The approach contains a set of rules, which examine the available choices for dominance and similarities, and attempts to rank them. If this set of rules fails to rank the available choices, then another set of criteria is applied. In this thesis, the hyperbolic discount function as shown in Eq. 2.1 is the preferred explanation as it is simple to understand,

simple to apply and encapsulates the key psychological phenomena of the self-control problem in that the present is given preferential treatment. It is also well supported in the literature (Ainslie and Haslam, 1992; Rachlin, 1995; Laibson, 1997). However, in this thesis, hyperbolic discounting is not implemented and the effect of alternative discounted value functions are explored.

### 2.2.2 Exercising Self-Control

To give an example where self-control behaviour is exercised, with reference to Figure 2.1, consider the student, let the *LL* represent obtaining good grades and *SS* going to the pub. Let $t_1$ indicate the start of an academic year. At this time for most students the value of getting good grades exceeds that of going to the pub. When invited to the pub at $t_2$ however, the value of *SS* is higher than their long term goal of getting good grades (*LL*). If the student exercises self-control then he or she will choose study (*LL*) over the pub (*SS*). Self-control behaviour encompasses a resistance to temptation, in this case to go to the pub (*SS*).

Figure 2.1 illustrates the traditional view of self-control in that the *SS* reward resulted from a single act (a single choice). In more complicated self-control problems the *LL* reward could result from a pattern of acts (a pattern of choices) and the *LL* reward could be distributed, part of it received after each act. For example, forgoing the cupcake once does not mean that the dieter will immediately wakeup with a supermodel body. In order that the dieter succeeds in her diet, she needs to forgo several cupcakes. The illustration in

Figure 2.1 of self-control has been criticized as being a too simplistic representation of self-control in real life, as it models the situation only where the rewards are mutually exclusive and discrete (Mele, 1995; Plaud, 1995). For example, in the case of the recovering addict there is a long-term pattern to stay clean *versus* a smaller-sooner reward of a fix, which is better explained in terms of a pattern of behaviour (to stay clean) *versus* a single act (a fix). Ainslie (1992) explains the self-control problem of an addict by thinking in terms of a series of choices, seen as a pattern of acts, as opposed to a single act. To illustrate what is meant by a pattern of acts, Rachlin (1995) says that an act is to a pattern, as a note is to a song. From this viewpoint of self-control, if we exercise self-control we are choosing a pattern of behaviour, which is composed of one or more acts, over an alternative single act. Within the literature on self-control this view is either supported unequivocally (Rachlin, 1995; Eisenberger, 1995), or dismissed entirely (Kanekar, 1995), or accepted, but with reservation (Kane, 1995; Plaud, 1995). Kane (1995) suggests that the definition of a pattern has two meanings: a pattern can either be a form of behaviour (habit), or an internal plan. Kane suggests that Rachlin's view of self-control is appropriate only to the habit type of pattern.

***Critical Observation.*** In this thesis we are not going to discuss the complications of self-control from the viewpoint of an act *versus* a pattern of acts, but we will assume that the person will have a single choice to make, as in Figure 2.1. Figure 2.1 illustrates the simple self-control problems where there is a clear preference for one alternative to another. Even the more complicated self-control problems such as the addict can still be viewed in the

context of Figure 2.1, if we consider it in terms of a larger, more delayed reinforcer (staying clean – the *LL* in Figure 2.1), over a smaller, less delayed reinforcer (a fix – the *SS* in Figure 2.1). Figure 2.1 encapsulates the key things that are relevant to this thesis: (i) the discounted values increase with time, (ii) initially the larger-later reward (*LL*) is preferred over that of the smaller-sooner reward (*SS*), and (iii) at some time before *SS* there is a reversal of preferences, with the value of *SS* overtaking and exceeding that of the *LL* reward. In this thesis we limit s*elf-control* behaviour to choosing a large delayed reward over a small immediate reward (Rachlin, 1995) as illustrated in Figure 2.1. The situation of refraining from going to the pub (the small immediate reward) in order to study to obtain good grades (the large delayed reward), and the situation of not stealing a bike (as an example of a small immediate reward) for loss of face or reputation (the larger delayed reward), are both self-control problems, and in this thesis are dealt with in the same way.

### 2.2.3 Defining Precommitment behaviour

Research by Ariely (2002) and Rachlin (2000) suggests that we recognize that we have self-control problems and try to solve them by *precommitment* behaviour. Precommitment behaviour can be seen as a desire for people to protect themselves against a future lack of willpower. Results by Ariely (2002) from a series of experiments on college students, showed that we recognize that we have self-control problems, and attempt to control them by setting costly deadlines. These deadlines help to control procrastination, but are not as effective as externally imposed deadlines, i.e., deadlines imposed by others. *Precommitment* is defined as making a choice with the specific aim of

denying oneself future choices (Rachlin, 1995). Schelling (1992) used the term *binding* to define precommitment behaviour. Examples are:

1. putting an alarm clock away from your bed, to force you to get up to turn it off.
2. saving part of your monthly pay cheque into an investment fund, to prevent you from spending it.
3. removing chocolate biscuits from your house to prevent late-night binges.
4. disconnecting the internet connection to your computer when you have a deadline.

All of the above are examples of the self-imposition of a behavioural constraint that limits a future choice. With reference to Figure 2.1, note that time $t_1$ is the ideal time to exercise self-control through precommitment, as the discounted value of the *SS* reward is low relative to the discounted value of the *LL* reward. Exercising self-control denies one's self the choice of the *SS* at time $t_2$. Precommitment is illustrated in Figure 2.2, which is based on experiments by Rachlin and Green (1972) on precommitment behaviour by pigeons. Self-control through precommitment behaviour can be illustrated by Figure 2.1 and Figure 2.2 in the following way. The discounted values of *SS* and *LL* in Figure 2.2. are the same as those shown in Figure 2.1. If we do not precommit, then in Figure 2.1 at time $t_2$, we can choose between a small immediate reward, *SS*, and a large delayed reward, *LL,* this situation is represented by the upper arm in Figure 2.2. If we precommit to *LL* at a prior point in time while the discounted value of *SS* is still low relative to *LL,* i.e.,

any time between time $t_1$ and $t_2$, then that restricts choosing *SS* at a later time $t_2$, this situation is represented by the lower arm in Figure 2.2.

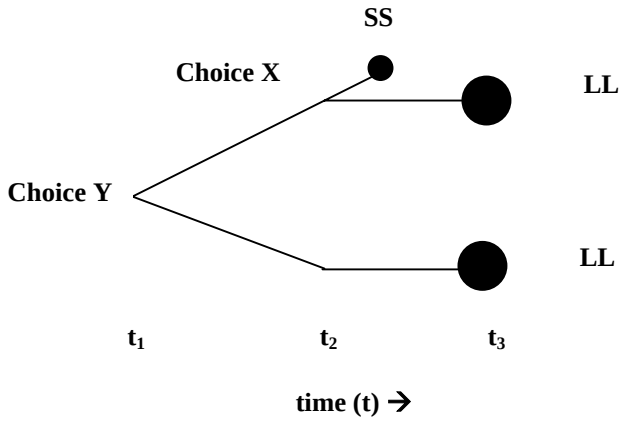



**Figure 2.2 Illustration of precommitment behaviour**
At choice *X* the smaller-sooner reward (*SS*) is preferred over the larger-later reward (*LL*). If precommitment behaviour is exercised, then at choice *Y* an alternative is preferred that restricts the choice to *LL* only, i.e., the lower arm (adapted from Rachlin, 1995).

Figure 2.2 illustrates the available choices. Exercising precommmitment at Choice *Y* means taking the lower arm, which precludes the *SS* at Choice *X,* (i.e., there are no chocolate biscuits in the house, since we removed them at Choice *Y*).

There are different levels of precommitment, which determine how successful the precommitment will be. According to Nesse (2001) precommitment is either (i) *conditional*, e.g., a threat, or (ii) *unconditional*, e.g., a promise. As he states, the carrying out of precommitment or not depends on how it is enforced. If the precommitment behaviour is *secured*, i.e., is enforced by the situation or a third party, then there is a greater degree of certainty that the behaviour will be carried out. If the precommitment behaviour is *unsecured*,

i.e., it depends on the individual's emotion or reputation, then it is less certain that the precommitment behaviour will be carried out.

Precommitment behaviour by physical restraint is less often adaptable to a changing environment and is difficult to implement. In real life precommitment is more often enforced by some sort of punishment (*P*). Schelling (1992) gives an example where addicts are invited to write a self-incriminating letter confessing their addiction. This letter will be sent to the addressee if any evidence of drug taking is found. At Choice *Y*, while the value of *SS* is still low relative to *LL*, the addict puts into place a punishment contingency, (e.g., the self-incriminating letter), for choosing *SS* at time $t_2$ that reduces the value of *SS*. The value of *SS* is now the original value of *SS* less the cost of the punishment *P*. This brings the net value of *SS* below that of the discounted value of *LL* at Choice *X*. Another example of precommitment by punishment is the case of alcoholics willingly taking the drug Antabuse before a social event where alcohol is available. The drug Antabuse causes severe pain if alcohol is consumed after taking it. Even though the drug does not stop alcoholics from drinking, it does act as a strong deterrent.

It is not necessary to enforce this level of precommitment for self-control in everyday life. In fact, most problems of self-control are managed without the explicit examples of precommitment as described above. Mischel at al. (1989) showed that as we grow older we become better at controlling our behaviour, i.e., we become better at choosing the *LL* reward over the *SS* reward. One explanation is that precommitment is internalized as we get older and our

emotions, such as guilt, function as an internal precommitment for exercising self-control. There is much support for this argument (Hoch and Loewenstein 1991; Frank, 1995; Muraven and Baumeister 2000). However, not all emotions make us better in choosing the *LL* over the *SS*. Baumeister (1995) believes emotions such as anger reduce higher-level cognitive processing and produce myopia, (i.e., the discounted value of *LL* is reduced). Alcohol has a similar effect. Baumeister (1995) likens emotions to a muscle that gets stronger with use, which could explain the development of self-control, as we get older. Rachlin (1995) opposes the view that precommitment can be internalized. He argues that if precommitment is internalized then how do we internalize punishment so that it has the same effect as the precommitment by external punishment described earlier? Rachlin (1995) suggests that people achieve self-control by restructuring behaviour to create a pattern of behaviour rather than by internalizing precommitment. The longer a pattern of behaviour continues, the more costly it becomes to stop (Rachlin, 2000). This suggests that self-control arises through the development of patterns of overt behaviour, which becomes a habit. Frank (1995) agrees that this habit forming is a means to greater self-control. Mosterin (1995) criticises Rachlin's view of the act *versus* pattern of behaviour, as being insufficient to explain the increase of self-control with age.

***Critical Observation.*** In my view, guilt provides a powerful mechanism for self-punishment. Loss of reputation or loss of respect from significant others is also a powerful incentive. Baumeister (1995) supports this view citing guilt as a sufficient motivation to change one's behaviour. Frank (1995) also

believes that guilt acts as a form of self-punishment by focusing our attention on the future. It may be difficult to identify internal precommitment. If a person refuses a cake, is it because she or he is not hungry, or through guilt because she or he is on a diet? In this thesis it is argued that although it may be difficult to measure a behaviour, this does not mean that the behaviour did not happen. For example, to the dieter, the action (refusing the cake – the *SS* in Figure 2.1), results in the same outcome (the dieter consumes less calories in line with the long term reward, i.e., the supermodel body – the *LL* in Figure 2.1). In this thesis, although the reason may be different (not hungry or on a diet), the fact that the action resulted in the right outcome is the important factor in self-control problems. What seems to be lacking from Rachlin's model of self-control is an explanation of the internal mechanism that motivates one to choose the *LL* over the *SS*. One of the aims of this thesis is to address this.

### 2.2.4 Precommitment behaviour and games

Precommitment behaviour features both in game theory and psychology. In game theory precommitment is simply called commitment. Von Neumann and Morgenstern (1944) mention commitment in their text, which established game theory. Although in game theory it is assumed that people interact in a rational manner (Binmore, 1992), it is still a powerful tool for studying human behaviour through simulation. The differences between game theory and what happens in the real world can be highlighted by comparing what people really do, with what is defined in game theory as the *optimal* thing to do. For example, in the Iterated Prisoner's Dilemma game, game theory states that the optimal thing to do is to defect; this is not actually what people do.

Experiments by Axelrod (1984) suggest that most people will cooperate early on, and will stop cooperating only if the other player defects, or when an end point is near. Another example is the Ultimatum game (Nesse, 2001), where one player proposes to divide a resource and the second player then accepts or turns all the resource back to the game. In game theory the optimal thing to do is to accept anything. Roth et al. (1991) showed that if the offer is not close to 50:50, then the offer is rejected.

***Critical Observation.*** One difference between precommitment, as defined in game theory, and what in this thesis is referred to as personal precommitment, is that personal precommitment may only be an announcement of plans (Nesse, 2001). Another difference is that in personal precommitment one has moral and other dilemmas. For the purpose of this thesis precommitment is defined in the same way as in game theory, i.e., once precommitment is announced, it must be irrevocable.

## 2.3 Explaining Self-control through games

Game theory (von Neumann and Morgenstern, 1944) is a powerful tool for studying human behaviour, if it is assumed that humans behave rationally, i.e., that a player is playing to win, which in real life is not always the case. In game theory there are two types of games: (i) *zero-sum* games are strictly competitive games where one player's payoff is the negative of the other, hence the payoff functions sum up to zero and (ii) *general-sum* games that look for a pattern of coalition consistent with rational behaviour and do not have any restriction on the players' payoff (Binmore, 1992). Zero-sum games can be viewed as a subset of general-sum games. Zero-sum games are simpler

in that they converge to a unique *Nash equilibrium* (Nash, 1950a), where each player's strategy choice is the player's best response to the strategy choice of other players (Nash, 1950a). This is not necessarily the case with general-sum games. General-sum games might have many equilibria (Nash, 1950a). A *strategy* or *policy* can be defined as the decision-making function, which specifies what *action* a player will take in any situation (*state*). An *action* is the behaviour that is performed. The state-action relationship is referred to in psychology as a stimulus-response. An *optimal strategy* is defined in terms of *Nash's Equilibrium* (Nash, 1950a). A game is said to be *symmetric* when the players start in the same situation, have the same choices of strategies and the same payoffs[1].

The *Prisoner's Dilemma* (PD) game is a symmetric general-sum game that has been used to model cooperation behaviour (Axelrod, and Hamilton, 1981). There has been much research on the PD game. Many researchers feel this simple game holds the key to problems ranging from strategic defence planning, to an explanation for animal behaviour in biology. Some biologists feel that many animals and plants are engaged in ceaseless games of PD played out in evolution (Dawkins, 1989). In the PD game there is a banker and two players. Each player has two cards one labelled *cooperate* (*C*) and one labelled *defect* (*D*). Each player chooses one of the cards and lays it face down without the other player knowing what choice he or she has made. The banker turns over the cards and rewards the players based on a payoff matrix

[1] The games deployed in this thesis are all symmetric general-sum games.

like the one in Figure 2.3. The payoffs to the row player are listed first. A negative payoff implies that the player pays the banker.

| | | **Column Player** | |
|---|---|---|---|
| | | **Cooperate** | **Defect** |
| **Row Player** | **Cooperate** | 1,1 | -1,2 |
| | **Defect** | 2, -1 | 0,0 |

**Figure 2.3 A payoff matrix for the Prisoner's Dilemma game**
The payoffs to the row player are listed first. It pays a player to defect whatever his opponent does; yet both players would be better off if they both cooperate.

The Prisoner's Dilemma game is defined in Figure 2.4.

| | **Cooperate** | **Defect** |
|---|---|---|
| **Cooperate** | R | S |
| **Defect** | T | P |

**Rules:**
1. T>R>P>S
2. 2R>T+S

**Figure 2.4 The Prisoner's Dilemma Game**
Defined by: Temptation to Defect (*T*) must be better than the Reward for Mutual Cooperation (*R*), which must be better than the Punishment for Mutual Defection (*P*), which must be better than the Sucker's payoff (*S*) (Rule 1: T>R>P>S); the average of the Temptation to Defect (*T*) and the Sucker's Payoffs (*S*) must not exceed the Reward for Mutual Cooperation (*R*) (Rule 2: 2R>T+S). The rewards are shown for the row player.

For the game to qualify as a PD it must satisfy two rules: (i) there must exist a rank order in the play, (i.e., Temptation to defect (*T*) must be better than the Reward for mutual cooperation (*R*); the Reward for mutual cooperation (*R*) must be better than the Punishment for mutual defection (*P*); the Punishment for mutual defection (*P*) must be better than the Sucker's payoff (*S*)) (Rule 1: T>R>P>S), and (ii) the average of the Temptation to defect (*T*) and the Sucker's payoff (*S*) must not exceed the Reward for mutual cooperation (*R*)

(Rule 2: 2R>T+S). Nesse (2001) suggests the four situations defined by the four boxes in Figure 2.3 have shaped emotions specific to the situation (trust and friendship, suspicion and anger, anxiety and guilt, and rejection). The dilemma in the Prisoner's Dilemma game is derived from trying to guess what card the other player has played. Each player has two choices: to defect (*D*) or to cooperate (*C*). With reference to Figures 2.3 and 2.4, if the other player played a *D,* the best card to play would also be a *D* and the player would receive a Punishment for mutual defection (*P*); to play a *C* would mean the player would receive a Sucker's payoff (*S*), which is even worse. If the other player played a *C*, then to play a *C* as well would mean a Reward for mutual cooperation (*R*); to play a *D* would give the player an even higher score for Temptation to defect (*T*). Therefore, whatever the other player does, the best move is always to play a *D* and thus the player's best response is to defect. Yet both players know that if they had both played a *C* for cooperation, they would both have benefited by the higher reward for mutual cooperation (*R*), instead of the punishment for mutual defection (*P*) and hence the dilemma.

Research into the original human version of PD has its origins in North America (Dawkins, 1989). The name Prisoner's Dilemma originates from the scenario of two prisoners who are in jail suspected of collaborating in a crime. The police have already got them on a lesser charge. Each one is invited to betray the other (defect). The outcome depends on what both prisoners do, and neither knows whether the other prisoner has remained silent (cooperate) or blabbed (defect). If the first prisoner blames the other (defects), whilst the second prisoner remains silent, cooperating with his treacherous friend, then

the second prisoner receives a heavy jail sentence and the first prisoner walks away free, having yielded to the temptation to defect. If each prisoner betrays the other both are convicted, but their sentence is reduced for mutual defection. If both refuse to speak to the authorities, they both get off with the lesser charge, i.e., they receive a reward for mutual cooperation. As neither prisoner can determine the action of the other, the prisoner's best response would be to betray the other and defect.

The Iterated Prisoner's Dilemma game (IPD) is more complicated. IPD is simply the PD game repeated an indefinite number of times with the same players. The successive rounds of the game give the players the opportunity to build up trust or mistrust, to reciprocate or placate, forgive or avenge. It is possible for a player to guess from the other player's past moves whether the other player is to be trusted. The iterated game offers plenty of strategic scope. In the standard iterated game neither player knows when the game will end. In the case of the IPD game, a *strategy* can be defined as a decision rule for each scenario of the game. A strategy could be a program whose input is comprised of the history of all moves up to the present and whose output is the move to make now. For example, a program could have a simple strategy such as Tit-for-Tat. The Tit-for-Tat strategy cooperates on the first move and thereafter copies the previous move of the other player.

Axelrod, working partly in collaboration with Hamilton (Axelrod and Hamilton, 1981), devised an evolutionary system to breed strategies for the IPD game. The Axelrod and Hamilton (1981) experiment is discussed in

Chapter 7. A brief discussion of Axelrod's findings is given here. In Axelrod and Hamilton (1981) evolutionary system, the success of a strategy is a measurement of how well it fares against the other strategies; they found that most of the strategies that evolved resembled the Tit-for-Tat strategy, which cooperates on the first move and then copies its opponent on the preceding move. Thus the Tit-for-Tat is a strategy of cooperation based on reciprocation. Their results showed that in evolution there is a conflict between the immediate reward of defection and the larger delayed reward of mutual cooperation that can be resolved by the probability of reciprocation. Defection yields the highest immediate reward (the *SS* reward in Figure 2.1). However, in the IPD game there is scope to build up trust between the players leading to the higher long term reward of mutual cooperation (the *LL* reward in Figure 2.1). There is also the conflict of exploration *versus* exploitation. If only the most successful individuals are allowed to breed, i.e., exploiting what is known, then this might be the best in the immediate future, by maximizing the performance of the next generation, but not necessarily the best in the long term. To achieve the optimal solution other possibilities need to be explored.

Rachlin (2000) suggests that the structure of the behaviour self-control can be likened to the IPD game in that cooperation is to defection what self-control is to impulsiveness. To illustrate this, consider the real world example of the self-control problem of the student faced with the temptation of going to the pub, discussed in Section 2.2.2. With reference to Figure 2.1, the *LL* represents getting good grades. At some point later in time the student receives an invitation to go to the pub and it is at this time that his temptation

(*SS*) becomes known, i.e. going to the pub. When invited to the pub, the student is faced with the self-control problem of staying at home and studying (the *LL* reward) or going to the pub and socializing (the *SS* reward). In Figure 2.3, the (*C,C*) is the *LL* reward of staying at home and studying leading to good grades and the (*D,D*) is the *SS* reward of going to the pub and socializing. If it is assumed that *C* is staying at home and *D* is going to the pub, then (*C,D*) could represent the middling situation of when asked to the pub you decide to stay at home, but do not study as effectively because you wish you had gone to the pub, and (*D,C*) could represent the situation of going to the pub, but having a miserable time because you feel guilty about not studying.

An experiment by Brown and Rachlin (1999) explored the relationship between self-control and cooperation, using human subjects playing a version of the IPD game. A game was played either by a single player to simulate self-control, or by a pair of players to simulate cooperation. In the experiment they had four trays to represent the four cells of the payoff matrix for the PD game. For the purpose of this illustration let us assume that each cell is in fact a box and the boxes are made of glass. The player knows what is in each box. The payoff matrix is shown in Figure 2.5. Let us assume each box has a door, which is opened by a red or green key. Inside the box there are some nickels (the reward) and another green or red key. There are two boxes with red doors and two boxes with green doors. The red key can open a red door, and the green key a green door. Each box has a number of nickels. After each round the box is refilled.

| | |
|---|---|
| **Red Door**<br>3<br>Red Key | **Red Door**<br>4<br>Green Key |
| **Green Door**<br>1<br>Red Key | **Green Door**<br>2<br>Green Key |

**Figure 2.5 The payoff matrix for the game of self-control or social cooperation**
The payoff matrix used in the self-control game and the social-cooperation game. A red or green door opens a box. In the box there is another red or green key, and a number representing the number of nickels (the reward). This is a game of self-control as the player has to choose between defecting and choosing the higher current reward, (either of the right hand boxes with 2 or 4), or cooperating and choosing the long term reward, (either of the left hand boxes with 3 or 1), (adapted from Brown and Rachlin, 1999)

In the *Self-Control* game there is just one player. The rules for the self-control game are: at the start of the game the player is given a red key, he or she can open one of the red doors and then surrender the key and take the nickels (the reward) and the key in the box. On the next round the key is used to open the box with the same colour. The aim is to maximize the number of nickels. If the player selects the top right hand box, then a green key is used on the next trial to open either one of the green doors resulting in a smaller reward (the punishment for taking the *SS* on the previous round). To maximize the total payoff, the strategy is to always choose the top left hand box and get three nickels, and only choose the top right hand box if it is the last round. The game is a self-control problem, as choosing the top right hand box has the highest immediate reward (*SS*), but it conflicts with the behaviour that maximizes the accumulated payoff in the long term (*LL*), i.e., choosing the top left hand box. The current choice is dependent upon the degree on which the next trial the higher future reward (top right hand box) is discounted. This is

referred to as *probabilistic discounting*. Rachlin (2000) defines *probabilistic discounting* as the case when:

"a player may currently discount higher future reward by the probability that she herself will fail to choose the lower reward on subsequent trials" (Rachlin, 2000, p.171)

For example, if the dieter by past experience thinks that it is highly improbable that she can resist calorific food tomorrow or the next day, why then resist it today.

In the *Cooperation* game there are two players. The rules for the Cooperation game are the same as with the Self-Control game except that each player took alternate trials. One player uses the key and then passes it to the next one. Hence, which key the player received, depended upon the other player's choice.

The results showed that the distinguishing feature between cooperation and self-control is the *probability of reciprocation*. Rachlin (2000) defines the *probability of reciprocation*, in terms of one's current action based upon what one believes one will do in the future:

"The important question is not, Will others cooperate (or will I cooperate) in the future? But If I cooperate now, will others cooperate (or will I cooperate) in the future?" (Rachlin, 2000, p.179)

Thus, with reference to Figure 2.5, this means the conditional probability that the player will continue to select the top left hand box (cooperate). This is

expressed as the number of times the top left hand box is selected over the total number times that both the top left hand box (cooperate) and the top right hand box (defect) are selected. When exercising self-control, the question becomes *if I cooperate with myself now, will I continue to cooperate with myself in the future?* For example, for the dieter the question is "if I refuse that cake now, will I continue to refuse it tomorrow?" The probability of reciprocation may be high in self-control situations where one is dealing with oneself.

An experiment by Baker (2001) showed that there is a direct relationship between the probability of reciprocation and cooperation. In Baker's experiment the game was played on a computer, with the computer taking the place of the second player. On the computer screen there was a diagram with four boxes as in Figure 2.5. In Baker's experiment the probability of reciprocation (*pr*) was explicitly stated. If the first player cooperated on the previous trial, then the computer on the next trial would cooperate with a probability of *pr* and defect with a probability of (*1-pr*). If the first player defected on the first trial then the computer on the next trial would defect with a probability of *pr* and cooperate with a probability of (*1-pr*). Thus, the probability of reciprocation of cooperation is signaled by the behaviour of the player on the first trial. The first player's move is known in psychology as the *discriminative stimuli.* This move signals when a reinforcer, in this case cooperation, is likely to be forthcoming. The results of Baker (2001) are summarized in Figure 2.6.

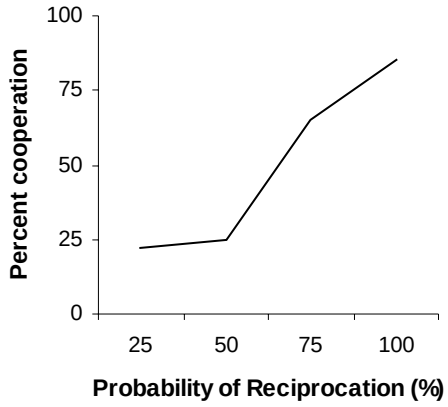



**Figure 2.6 Baker's results (2001) showing cooperation as a function of reciprocation**
The average results of the last fifteen of one hundred trials of Baker's experiment (2001) showing that the tendency to cooperate is directly related to the probability of reciprocation. The self-control problem can be stated as: *if I cooperate with myself now, will I continue to cooperate with myself in the future?* (Rachlin, 2000).

The results show a positive correlation between the probability of reciprocation and the tendency to cooperate, i.e., as the probability of reciprocation increases so does the percentage of cooperation. When problems of self-control occur, the probability of reciprocation decreases (Rachlin, 2000). The smoker, who has stopped several times, must have started several times. There is no reason to cooperate today, if the smoker believes that he or she will begin again tomorrow.

There is much support in the literature for this link between self-control and cooperation. The philosopher Plato wrote in his piece Phaedon (written in 360 B.C) of a metaphor for the human soul as a two-horse chariot. The charioteer represents the higher brain functions such as planning and reason, and the horses represent immediate gratification. The horses are myopic and the

charioteer is far-sighted, because the charioteer has control of the horses. More recently, Schelling (1971), Platt (1973) and Ainslie (1992) have also claimed that there is a relationship between self-control and cooperation. It is only when inconsistencies between the short term (the smaller-sooner reward) and the long term (the larger-later reward) arise, that conformity to the preferred long-term is labeled self-control.

***What is new and overview?*** In this thesis it is proposed that increasing the level of precommitment increases the probability of cooperating with oneself in the future, i.e., the probability of reciprocation. Hence, in this thesis the theoretical premise is made that as the level of precommitment increases, so to does the tendency to cooperate, as suggested by the results of Baker (2001) and Brown and Rachlin (1999). The word "cooperation" has a non-conventional meaning in this thesis (only cooperate for the high-payoff decision). Cooperation as defined in the Oxford dictionary, is to "work together for a common end", in which case (*D,D*) could be viewed as cooperation. In this thesis and also in the IPD game, cooperation means cooperating in order to gain the larger later payoff, hence the situation where both players defect (*D,D*) is not seen as cooperation.

## 2.4 Physiological evidence for Self-Control

It is has been suggested that there is already a neurobiological explanation for self-control (Hughes and Churchland, 1995). However, from a review of the literature, it would seem that there is still much to be learned about the physiological basis for self-control. It follows that if consensus has not been

reached on a definitive functional model, then how can there exist a neurobiological model encapsulating the psychological phenomena?

The earlier neurobiological data on self-control was observed from brain lesion patients. Brain lesion data is problematic, as it is difficult to determine the exact role of the lesion in the breakdown of the behaviour (Churchland and Sejnowski, 1992). It is only recently that brain-imaging data has been made available. Early explanations for the neurobiological basis for self-control were based on data from studies on brain-lesion patients such as the patient, referred to simply as, EVR (Damasio et al., 1990). Damasio (1994) found that lesions in the orbital and lower medial frontal lesions resulted in deterioration of social conduct and judgment. Further experiments showed that the frontal ventromedial has a role in the decision-making process for long term costs and benefits. Recent studies by O'Doherty et al. (2002), using brain-imaging techniques, have confirmed this. MRI studies have located the area that anticipates reward. This is believed to be the ventral striatum centred at the base of the brain (Knutson et al., 2001). A MRI study carried out by Bjork et al. (2004) on adolescents and young adults found that there was less activity in this area for adolescents as compared to adults when faced with the anticipation of future gain. They also found that the anticipation of a potential reward showed activity in right insula, dorsal thalamus and dorsal midbrain.

***Critical Observation.*** From a review of the literature, summarized above, it would seem that there is still much to be gleamed from a detail analysis of the physiological basis for self-control behaviour. What is needed is a model that

provides both a neurobiological and a behavioral explanation to avoid past mistakes, e.g., the procedure for frontal lobotomies and medical blunders such as the patient EVR. This research aims to bridge the gap, between the neurobiological and behavioral data.

## 2.5 Evolution of Self-Control through Precommitment

Precommitment behaviour can be viewed as an indicator of the internal conflicts that arise in our brain (Nesse, 2001). If we were truly rational, our preferences would not change over time: if it is in our interest to get up when the alarm clock goes off, then we should not want to go back to sleep when it wakes us up (Samuelson and Swinkels, 2002). Are we born with this capacity? A survey of the literature would seem to suggest that there is an evolutionary basis for this complex behaviour, and that our genes have been shaped through natural evolution to provide us with this capacity of precommitment even when there is a cost (Samuelson and Swinkels, 2002). Experiments on college students by Ariely (2002) show that we commit to less than optimum deadlines. Gibbard (1990) proposes that natural selection has evolved our emotions to create a "normative control system" to achieve our long term goals through constraint of the immediate reward. Frank (1988) also suggests that our emotions have evolved to act as an internal self-control mechanism, which kick in when required. This suggests a conscious *versus* unconscious mind or, as suggested in the literature, a multiple selves theorem (Schelling, 1992; Trivers, 2000; Samuelson and Swinkels, 2002). In the two-self model, the current self, concerned with the present, might wish to restrict the choice of the future self, even though it makes its future self unhappy. It does this, because it is aware that it will make other future selves happier.

Nesse (2001) suggests that precommitment is so important to our survival that evolution has evolved "specialized capacities" to do this. Cosmides and Tooby (1989) support this view. They propose that evolution has given us a "cheater detection module", which enables us to inhibit short term pleasure in order to fulfill commitments in pursuit of long term goals. Have we through evolution, as suggested by Burnham and Phelan (2000), tamed our "primal instincts" to give us this capacity to precommit, which overrules our short term pleasures (primal desires) to achieve our long term goals? Metcalfe and Mischel (1999) suggest that a higher process curtails our intrinsic, reactive behaviour. Metcalfe and Mischel (1999) account for the differences in the development of self-control in individuals by suggesting that there are two processing systems which govern our development of self-control. They propose a "hot" system that is emotive and reactive, and a "cool" system that is cognitive and reflective. The interconnection of these two systems determines the development of self-control. The prefrontal cortex is the newest part of our brain, and is the part of our brain, which has been subjected to the majority of changes under natural evolution (Greenfield, 1997). From studies of brain-damaged patients such as Phineas Gage and the brain-lesion patient known simply as EVR (Damasio, 1994), the prefrontal cortex has been shown to be crucial in performing a cost-benefit analysis of the short-term choices *versus* the long-term choices.

Alternatively, there may be no evolutionary basis for self-control through precommitment and this behaviour is learned as part of socialization alone. A review of the psychological literature on how to achieve greater self-control

would seem to support this theory. For example, the way operant conditioning is applied to human behaviour to achieve greater self-control, suggests that once a target behaviour is selected then one should plan a course of action and make a record of each occurrence of the target behaviour; then he or she should change the environment and use positive reinforcement each time temptation is avoided. Strotz (1956) proposes that a strategy of precommitment techniques is learned from an external self-control device to manage the internal conflict of short-term pleasure *versus* long-term gain. The rules learned then become habits. As a child, a parental voice might suffice as an external self-control device. This is supported by the results of Kochanska et al. (2000), which demonstrated that children of mothers, who were more sensitive and supportive, were more advanced in their development of self-control. However by early adolescence, parental influences make way for peers, films, computer games etc.; do these influences act as self-control devices?

Figure 2.7 shows the key milestones in the development of self-control (Morgan et al., 1979). The child begins by complying, i.e., the child voluntarily obeys requests and commands. By the age of two, the child has an ability to wait for a reward. At the age of 6 the child has acquired the cognitive ability to inhibit the pleasure of short term rewards, e.g., to think of marshmallows as clouds (Mischel and Mischel, 1983) and has the capacity for moral self-regulation, i.e., the ability to monitor one's conduct. At this stage the child is beginning to learn to think in an abstract way. To achieve these milestones a child must have the necessary cognitive processes in place. A

child must have memory skills to recall a mother's directive in order to apply it to its own behaviour. For a child to apply self-regulation she or he must be able to think of him or her self as separate beings that can control their own actions and emotions. The child must have cognitive inhibition in order to be able to inhibit short term pleasures (Mischel et al., 1989).

| Age | Self -Control Behaviour |
|---|---|
| 12-18mths | Beginning of compliance, i.e., voluntary obedience to request and commands |
| 18-30mths<br>To 5yrs | Ability to wait increases, i.e., delay of gratification<br>Compliance and delay of gratification improve |
| 6 to 11 yrs | Strategies for self-control expand<br>Awareness of ideas to gain rewards<br>Capacity for moral self regulation, i.e., the ability to monitor one's conduct |
| 12-20yrs | Moral self regulation improves |

**Figure 2.7 Milestones in the development of self-control**
These milestones represent overall age trends; individual differences exist in the precise age grouping (adapted from Morgan et al., 1979).

***Critical Observation.*** The review of the literature on self-control behaviour suggests a dual-process system. Gibbard (1990) suggests a "controller", Frank (1988) suggests an emotive system that kicks in when required and Metcalfe and Mischel (1999) suggest a hot and cold system. In this thesis it is proposed that the interconnection of these two processes alluded to by Gibbard (1990), Frank (1988), Cosmides and Tooby (1989) and Metcalfe and Mischel (1999), is representative of the competition between the higher centre of the brain, (i.e., rational thought) and the low-level centre, (i.e., instinctive behaviour). If this is the case, is it that self-control stems from a genetic conflict between the maternal genes congregated in the neocortex *versus* the paternal genes congregated in the hypothalamus as suggested by Haig (1997) and Trivers and

Burt (1999)? If self-control through precommitment has been shaped by natural evolution, then there must be a fitness benefit for the gene or it will be eliminated through natural selection. Natural evolution works at the gene level: it benefits genes and their phenotypes, not the group or species (Maynard Smith 1982; Dawkin, 1989). Is it the case that the capacity for precommitment has increased fitness, which in turn has shaped higher intelligence, which has then reinforced precommitment behaviour?[2]

The evidence would seem to imply that as our brains have evolved, so too has our capacity to precommit. The empirical and theoretical data on self-control implies that there is an evolutionary heritage to self-control through precommitment behaviour. When our environment was less predictable and our basic needs were less likely to be met, then the present was weighted more heavily. As nature has evolved however, so too did our capacity to precommit. Fantino (1995) summarises this in the title of his article on the evolutionary reasons for impulsiveness, "The future is uncertain, eat desert first".

## 2.6 Summary

For the purpose of this thesis, *self-control* is defined as choosing a larger-later reward over a smaller-sooner reward as shown in Figure 2.1. The key points in Figure 2.1 that are relevant to this thesis are: (i) the discounted values of both rewards increases with time, (ii) initially the value of the larger-later reward (*LL*) exceeds that of the smaller-sooner reward (*SS*) allowing us to precommit to *LL*, and (iii) at some time before *SS* there is a reversal of

[2] In this thesis, an answer to this question will be investigated in the context of the explanation (2) in Chapter 1, Section 1.1 (p. 23).

preferences, with the value of *SS* reward overtaking and exceeding that of the *LL* reward. Precommitment is a mechanism for managing self-control problems. *Precommitment* is defined as making a choice with the specific aim of denying or limiting ones future choices. When *LL* is the preferred option at $t_1$ we precommit to *LL* by carrying out an action that limits our later choice to *LL* only. Exercising self-control can be viewed as the tendency to cooperate with one's self (Brown and Rachlin, 1999). In this thesis the word *cooperation* has a non-conventional meaning; more specifically it means cooperating in order to gain the larger later payoff, hence the situation where both players defect is, in this thesis, not considered to be cooperation. The problem of self-control then becomes: "if I cooperate now with myself will I continue to cooperate with myself in the future?" This can be interpreted as in the *probability of reciprocation*. Baker (2001) showed increasing the probability of reciprocation increases the tendency to cooperate. The premise made in this thesis is that by increasing the level of precommitment, the probability of reciprocation increases, and as a result the tendency to cooperate.

For there to be an evolutionary basis for self-control through precommitment behaviour, there has to be a fitness benefit. The literature seems to suggest that this is the case, but studies of children on delay of gratification and internalization of precommitment would suggest that learning plays some role in the development of self-control. In this thesis, the role of both learning and evolution in self-control through precommitment behaviour is investigated.

From a review of the literature there is much support for a multiple-self model of self-control. Cosmides and Tooby (1989) propose a two-process model where there is a cheater module. Metcalfe and Mischel (1999) propose a hot and cold two-process system. In the next chapter, a simple neural model for self-control through precommitment is introduced. In this simple neural model, which is developed and tested in this thesis, the higher and lower brain regions are represented as two ANNs, locked in competition for control of the organism.

# Chapter 3

# 3 Review of the concepts for the Neural Modelling of Self-Control through Precommitment

## 3.1 Chapter Outline

This chapter is a critical literature review of relevant neural models of self-control and related behaviours. Each model is discussed in the context of self-control behaviour and the pros and cons highlighted. The chapter concludes by presenting the computational model of self-control that is developed and tested in this thesis.

## 3.2 Support for the dual-process model

We have described self-control as a dilemma between a future self, concerned with long-term benefits, and the present self, concerned with immediate gratification, see Chapter 2, Section 2.2.1. Support for this conflict goes as far back to Plato and his metaphor for the human soul, a two-horse chariot. The charioteer is reason and the horses stand for immediate pleasure. Adam Smith's 2-self model in his book "Theory of moral sentiments" (1759) talks about a conflict between reason and passion for control of the moral sentiment of the person. Recent research also supports this conflict within one's self. Thaler and Shefrin (1981) propose a two-self model of myopic doer *versus* far-sighted planner. Smolensky (1988) suggests a top-level conscious processor for effortful reasoning and an intuitive processor for heuristics, intuitive problem solving. The common theme in these approaches is that influences can come from both top downward and bottom upward. The

religious view of self-control is illustrated in Figure 3.1. It encapsulates this view of two processes, which are largely independent locked in some form of internal conflict for optimal control of the organism. The circle represents the body of a person. The thick black arrow represents the person's emergent behaviour, which is the result of a continuous battle between the good (the angel) and the bad (the devil). Self-control is concerned with keeping that arrow facing upward; temptation or impulsiveness is concerned in keeping the arrow facing downward. In this religious model a person exercises self-control as a result of good external influences, for example, parents, school, and the church.

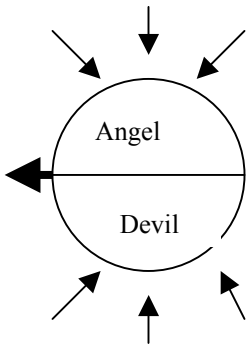



**Figure 3.1 The religious view of self-control**
The arrows entering the devil represent temptation and the arrows entering the angel are good social influences, e.g., parents, school, the church (adapted from Rachlin, 2000)

***Critical Observation.*** This religious model of self-control focuses on the reasons for *why* we behave as we do. It ignores the biological mechanisms underlying self-control. The religious model of Figure 3.1 separates the operations of the mind from the biological organism. The suggestion that reasoning and moral judgment, and the pleasure and pain that ensues, can be separated from the biological mechanisms that underlie the behaviour, paves

the way for dualism, where the mind and brain are viewed as radically different kinds of things.

### 3.3 Alternative abstract models

Carvier and Scheier (1998) propose a model of self-control based on self-regulation of behaviour through feedback. Self-control in this model is seen as maintaining conformity to a standard. The model is shown in Figure 3.2.

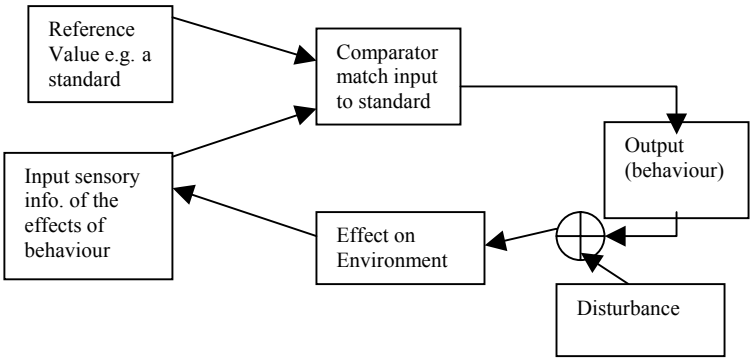



**Figure 3.2 A model of self-control that uses self-regulation**
An abstract behavioural model of self-regulation, which can be used to exercise self-control. In this model self-control can break down in several ways: (i) when there does not exist a standard for a behaviour, e.g., being on time for work; (ii) we fail to carry out any monitoring on our behaviour, e.g., setting the alarm clock; (iii) we simply fail to follow through, e.g., we switch the alarm clock off and go back to sleep (adapted from Carver and Scheier, 1998).

In this model problems of self-control can occur where there is a break down through, a lack of standard for a behaviour, or a lack of monitoring, or a failure to follow through on a behaviour. This model can be illustrated in terms of a thermostat control system. The standard is the thermostat; the sensory information is the thermostat control. The comparator matches the temperature to some desired level and the temperature is adjusted accordingly. Disturbances may be the sun, the wind, or the number of people in the room. This model explains self-control behaviour in terms of maintaining some

standard be it moral or otherwise. This standard has got to be stable. It is important that the reference value is held rigid, as fluctuations in this will lead to erratic behaviour. Error detection compares the standard and behaviour. The level of error detection determines self-regulation hence self-control. For example, a low error detection would result in sloppy or careless behaviour.

***Critical Observation.*** In this abstract model, behaviour is explained in the context of personality and social psychology. It does not attempt to explain how messages are passed between our brains and our bodies, and where in our brains messages go to or come from. In a sense it fails in the same way as the religious model of Figure 3.1, as it ignores the biophysical mechanisms that underlie the behaviour, i.e., the *how*. In the next section neurobiological models will be presented, which attempt to address this issue.

## 3.4 Neurological support for the dual-process model

From a review of the neurophysiological literature on the structure and function of the brain, it seems that there is much support for a dual-process model (Bjork et al., 2004; Frank et al., 2001; Sporns et al., 2000; Beiser and Houk, 1998). The dual-process model has a planner type function, located in the prefrontal cortex, and a doer type function, located elsewhere in the brain possibly the midbrain or upper brain stem. Frank et al. (2001), describe the division of labour between prefrontal cortex and basal ganglia in the context of reinforcement learning. The basal ganglia (more specifically the ventral limbic and the striatum regions) is involved in the assignment of a reinforcer signal to a sequence of actions and the prefrontal cortex acts as the critic in the

creation of the reinforcement signal. Beiser and Houk, (1998) support this in their model of serial ordering of events. The ordering of events is an important cognitive function of self-control problems. Beiser and Houk (1998) show that messages pass via a loop, starting from the prefrontal cortex through to the basal ganglia and thalamus, and then back to the prefrontal cortex. This model is a departure from the traditional role of the basal ganglia. Traditionally the basal ganglia has been associated with motor skills. Sporns et al., (2000) in their discussion of the processes needed when facing complex, dynamic environments identify two opposing functions: extraction and response. These two functions are dealt with in two seemingly independent, but cooperating areas of the brain. The first is located in the cortical area, and the second within and across the cortical area. More recently Bjork et al. (2004) have shown the division of motivation and gain in reward-directed behaviour can be mapped to the limbic and frontal cortex regions respectively. Finally, Damasio (1994) proposes a neural basis of self, as two representations: (i) the individual's autobiography, based on memory and (ii) an imagery of a future self, which is subjective.

***Critical Observation.*** These models all attempt to explain how self-control behaviour occurs in terms of biological processes (to the best of my knowledge, precommitment has not been explained in a neurological model). However, (with the possible exception of Damasio) the models ignore *why* this behaviour occurs, i.e., the motivation for the behaviour that was addressed by both the religious model of self-control (Figure 3.1) and the abstract model of self-control (Figure 3.2).

## 3.5 A model of self-control

In the viewpoint of modern cognitive neuroscience, self-control as an internal process can be represented in a highly schematic way as in Figure 3.3 (Rachlin, 2000). Arrow **1** in Figure 3.3 denotes information coming into the cognitive system located in the higher centre of the brain, which represents the frontal lobes associated with rational behaviour such as planning and control. This information combines with messages from the lower brain, representing the limbic system (including memory from the hippocampus) that is associated with emotion and action selection (O'Reilly and Munakata, 2000; Rachlin, 2000). This travels back down to the lower brain and finally results in behaviour (arrow **2** in Figure 3.3), which is rewarded or punished by stimuli entering the lower brain (arrow **3** in Figure 3.3).

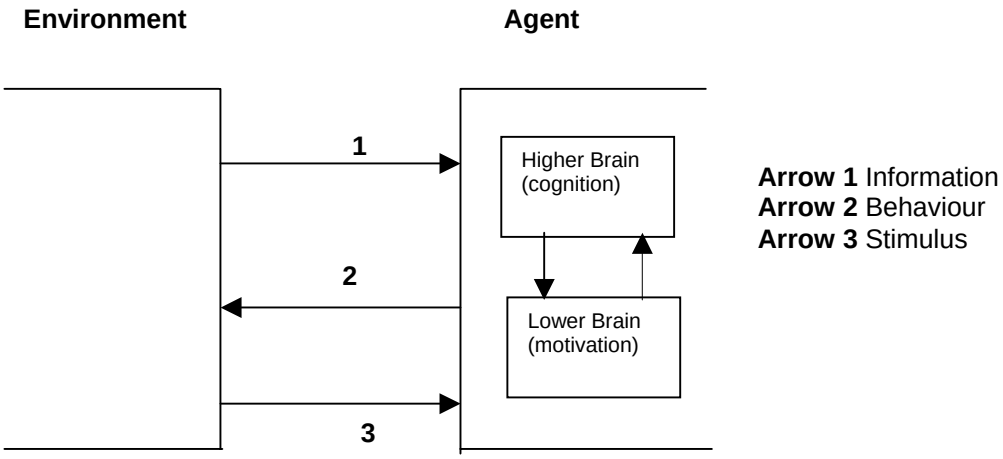



**Figure 3.3 A model of self-control behaviour as an internal process**
Self-control as an internal process, from the viewpoint of modern cognitive neuroscience. Arrow 1 summarizes information both past and current about the environment, Arrow 2 is the emergent behaviour of the Agent, and Arrow 3 is the reinforcement signal, which can either be a global reward or penalty as a response to Arrow 2 (based upon Rachlin, 2000).

The model, as depicted in Figure 3.3, is accepted in modern physiological and cognitive psychology as a model of self-control. To illustrate this, with reference Figure 2.1, self-control is defined as the choice of a larger, but later reward (*LL*) available at $t_3$ over a sooner, but smaller reward (*SS*) at time $t_2$. Information about an earlier time $t_1$, i.e., the value of the larger-later reward is stored away elsewhere in our brain possibly memory this combines with information coming into our brain about the immediate environment, i.e., the sooner-smaller reward at $t_2$ represented by Arrow 1. This results in an action denoted by Arrow 2, either to defect and succumb to temptation and receive the immediate reward at $t_2$ denoted by Arrow 3, i.e., *SS*, or to cooperate and to not give way to temptation and receive the larger later reward at $t_3$ denoted again by Arrow 3. The model explains where in the brain messages go to and come from and how the behaviour emerges. With reference to Figure 2.1, we are made aware of the temptations (the *SS* in Figure 2.1) by information coming into the cognitive system (Arrow **1** in Figure 3.3) This information combines with messages from the lower brain, and other information stored elsewhere (possibly memory) of our long term goals (the *LL* in Figure 2.1). A choice is made, either the *LL* or the *SS,* which finally results in behaviour (Arrow **2** in Figure 3.3). We are then rewarded with the *SS* or *LL* (Arrow **3** in Figure 3.3).

***Critical Observation.*** The model of self-control in Figure 3.3 aims to explain the biological mechanisms underlying self-control behaviour by showing at an abstract level where in our brain messages originate and are sent to. It also attempts to explain *why* self-control behaviour occurs, but not how greater

self-control can be achieved, which both the religious model of Figure 3.1 and the abstract model of Figure 3.2 attempted to address.

### 3.5.1 What is new and overview?

In this thesis, the simple model of Figure 3.3 is implemented as an architecture of two interacting networks of neurons. This follows on from the ideas proposed in Section 3.1 and supported by the neurophysiological data in Section 3.2 that the two hemispheres of the brain engage in a competition for control of the organism. In this thesis we make the theoretical premise that the higher and lower brain functions cooperate, i.e. work together, which is in contrast to the traditional view of the higher brain functioning as a controller overriding the lower brain. From this viewpoint, a computational model of the neural cognitive system of self-control behaviour is developed. The schematic model of Figure 3.3 is implemented as two Artificial Neural Networks (ANNs) simulating two players, representing the higher and lower centers of the brain, competing against each other in *general-sum* games using reinforcement-learning. It is a network architecture of two networks exhibiting different behaviours to represent the higher *versus* lower cognitive functions, as depicted in Figure 3.3. The *State* (corresponding to arrow **1** in Figure 3.3) summarizes information both past and current about the environment; the *Action* (corresponding to arrow **2** in Figure 3.3) is the emergent behaviour of the combined networks and the *reinforcer* (corresponding to arrow **3** in Figure 3.3) is a global reward or penalty signal as appropriate to the action. From this model of self-control behaviour, precommitment behaviour can be viewed as resolving some internal conflict between the functions of the lower and the higher centres of the brain by

restricting or denying future choices and hence can be thought of as resolving an internal conflict by prevention. It does this by biasing future choices to the larger, but later reward (the lower arm leading to *LL* in Figure 2.2). This is simulated in this computational model in one of three ways: (i) as a variable bias in place of the ANN's bias, (ii) as an extra input implemented on one or both ANNs or (iii) as a differential bias applied to the payoff matrix.

The computational model implemented in this thesis explains in computational terms how the brain generates self-control behaviour, based on the known neurophysiology of the brain, from a top-down modelling approach. Complex processes like self-control cannot be understood simply by the operations of individual neurons, it requires an understanding of the interaction of multiple components, i.e., networks of neurons responsible for specific functions (Fodor, 1983; Jacobs, 1999). Current research indicates that the higher cognitive functions are not based on the action of individual neurons in a limited area, but are based on the outcome of the integrated action of the brain as a whole (O'Reilly and Munakata, 2000). For this reason, a holistic approach to modeling the brain as a functionally decomposed system from a top down perspective is adopted, which is appropriate given the complexity and scope of the behaviour. In this thesis, the model explores the neural competition between modules (Jacobs, 1999). The model also takes into consideration the complexity of the environment as well as behaviour. The variables that define the ANN are parameterized to enable control of the model. These include the form of learning, the learning rate and the number of neurons in the each module.

## 3.6 Concluding Remarks

In this chapter we considered three models. The first is a model from the spiritual viewpoint of self-control; this model provided an explanation of why we may achieve self-control, but it did not provide an explanation of what is happening in our brain when we engage in self-control behaviour. The second model by Carver and Scheier (1988) provides an alternative explanation of the reasons why we exercise self-control and attempts to explain the how, but it fails to explain where in our brain messages go to and come from. The computational model developed in this thesis, depicted in Figure 3.3, bridges this gap by providing an explanation of both the *how* and the *why*. It does this by translating abstract purpose (modelled as a bias to *LL* or *SS*) into a specific action (precommitment), for self-control through precommitment behaviour.

In the next chapter reinforcement learning is introduced, firstly in the context of self-control, followed by an explanation of how reinforcement learning is to be used in this thesis.

# Chapter 4

# 4 Review of Reinforcement learning in the context of Self-Control

## 4.1 Chapter Outline

This chapter introduces reinforcement learning in the context of self-control. It presents an overview of reinforcement learning and how reinforcement learning is implemented in an artificial neural network. The chapter concludes by explaining how reinforcement learning and artificial neural networks are used in this thesis.

## 4.2 A novel approach to the self-control problem

In this thesis the aim is to achieve a greater understanding of self-control through precommitment behaviour by simulating the processes in the brain that are executed when we exercise this behaviour. Reinforcement learning (RL) supports this aim, as it is concerned with algorithms and processes going on inside an agent as it learns. In RL, the agent's goal is to maximize its rewards in the long term. This means being able to represent the value of future rewards now. In order to do this, there is an additional concept of discounting. The discount rate determines the present value of future rewards. A discount rate of zero is myopic. Myopic refers to maximizing the immediate reward. As the discount rate approaches one, the agent becomes more far-sighted and takes future rewards into account more strongly. The discount rate can be viewed as the relative value of a delayed reward *versus* an immediate reward. In many life situations we apply the concept of discounting where we prefer the reward sooner rather than later. We have defined s*elf-control* as the

dilemma of choosing a large delayed reward over a small immediate reward (Rachlin, 1995). Increasing the delay of the later reward, increases our discounting of the reward (Mischel et al., 1989). The greater the discounting of the larger-later reward, the sooner its current value sinks below the value of the smaller-sooner reward. As we become more far-sighted we discount future rewards less; this is modelled as an increase in the discount rate in RL.

## 4.3 A Brief History of Reinforcement Learning

Reinforcement Learning (RL) was born from three separate, but related areas of research. The first has its roots in psychology "Learning by *Trial and Error*" (TE), which can be summarized by Edward Thorndike's Law of Effect:

"Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond" (Thorndike, 1911, p.244).

From this we can gleam two mental processes: selecting an appropriate action for a situation (search) and then associating that action with that situation to determine when and what actions work best (memory). These are the basic processes of RL. The second area of research that contributes to RL is *Temporal Difference* learning (TD). TD is linked with classical (or Pavlovian)

conditioning in psychology (Sutton and Barto, 1987). TD works on the notion of a secondary reinforcer that takes on similar reinforcing properties as a primary reinforcer. The main contributors to TD are Minsky (1954) who saw the relevance of this form of learning to artificial intelligence, and Samuel (1959) in his Checkers program, which is the first known implementation of the TD method. The final research area to contribute to RL is that of Optimal Control founded on the work by Bellman (1956). This has been the basis for *dynamic programming* (Bellman, 1957a) and the mathematical framework for single agent RL, the *Markov decision process* (Bellman, 1957b).

The 1960s and 1970s were the Dark Ages for RL. RL was confused with Supervised learning. There were some exceptions like Narendra and Thathachar (1974) who developed a learning algorithm for solving non-associative problems such as slot machines, and Widrow, Gupta and Maitra (1973) who modified the least mean squares algorithm (LMS training rule) of Widrow and Hoff (1960) to produce the *Selective Bootstrap* adaptation rule otherwise known as "learning with a critic". The 1980s saw a renewed interest in RL with psychological models of classical conditioning learning, which combined the learning approaches of TD and TE (Sutton and Barto, 1987; Klopf, 1988). During this period Holland (1986) developed classifier systems, which combined TE learning with Genetic Algorithms. The Actor-Critic architecture (Barto et al., 1983) and TD($\lambda$) (Sutton, 1988) combined TE and TD. Watkins (1989) with Q-learning finally brought together the three research areas: TD, TE and DP. Q-learning is the RL algorithm of choice of many of the researchers in this field today.

## 4.4 Elements of Reinforcement Learning

Reinforcement learning is learning how to behave in any given situation. Reinforcement learning can be described as *learning by experience* as the learner is not told which actions to take but, instead must discover which actions yield the most reward by trying them. This learning by experience may be over the entire lifetime of the learner. The learner or decision-maker, e.g., the player or robot, is defined as the *agent*. An *action* defines what the learner can do in a given situation, e.g., what moves to play on a Chess board. A *play* or an *episode* can be defined as an instance of selecting an action. The *goal* is to maximize the expected total reward over some time period, for example, to gain the maximum payoff over 100 action selections, i.e., plays. The agent cannot change or influence the goal. For example, in a game of Chess the goal is to check mate the other player and win the game. Whether the agent achieves this, is determined by how well or badly the other player plays. Each action has an expected reward if selected, which is called the *value* of that action. There are many ways to estimate the value of an action. In RL the aim is to learn to estimate or to predict the value of an action accurately. The learning can be described as *nonassociative,* defined as learning how to act in one situation, or *associative* in which there is a requirement to associate different actions with different situations. *Action-selection* is defined as the problem of selecting the action that is appropriate for the present situation.

One of the distinguishing features of RL is the dilemma of when to *exploit* current knowledge of actions that will gain an immediate reward, and when to

*explore* new actions, that might gain a reward. The *greedy action* is the action whose estimated value is the greatest at any one time. To select the *greedy* action is to *exploit* current knowledge. To select a *nongreedy* action is to *explore* new actions. Exploitation will maximize the immediate reward, but exploration may produce the greater total reward in the long term. There are many methods concerned with the problem of action-selection. *Greedy selection* exploits current knowledge and always selects the action with the highest value. The *near-greedy selection* performs the greedy selection in the majority of cases, but will select a non-greedy action at random. A variation on this is the *softmax* action-selection, where all actions are ranked according to their estimated value. The above describes *action-value* methods for action selection, where an estimate of the value of a particular action is maintained. Alternatively the *comparison* methods for action-selection maintain an overall rating of an action as compared to other actions. The choice of method depends on the task. Where the true values of the actions change little over time, there is no need to explore further than just trying each action once. Where the true values of the actions change over time for each situation, it would take more exploration to find the optimal action at any one time. The RL framework for a single agent is summarized in Figure 4.1. The *environment* is everything outside the agent. The limit between agent and environment can be fuzzy. For example, the sensory input connections on a robot would be considered as part of the environment. The agent is not defined by the limit of its physical body, but anything that it cannot change randomly. That is to say the agent is defined by the limit of its control, but not of its knowledge.

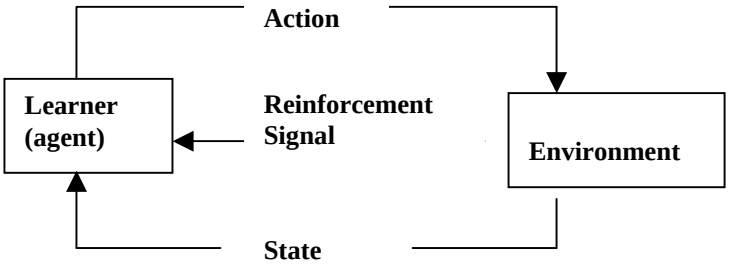



**Figure 4.1 A model for a single agent reinforcement learning.**
The single agent-environment interaction in reinforcement learning implemented in the Markov decision process framework. An *Environment* contains everything external to the agent. The *State* is a summary of past behaviour that is needed to determine future behaviour. The *Agent* is the learner, e.g., the player or ANN. An *Action* is what the agent can do, e.g., a board move, selecting a lever. There is a *Reinforcement Signal* to evaluate the current action, but it does not tell the learner what is the best action to take (Bellman, 1957b).

The *agent-environment interface* comprises of: (i) the process of the agent selecting the action, (ii) the environment reinforcing the selected action, (iii) updating the environment and (iv) presenting the new situation to the agent. The agent and the environment interact at a sequence of time steps, which may be discrete or continuous. The *state* is the representation of the environment. It provides the basis on what choices are made, but it does not need to inform the agent of everything about the environment. For example, in a game of cards the learner does not know the next one in the deck before it is dealt. If the state summarizes, yet retains relevant information on past states it is said to be *Markov* (Bellman, 1957b). Having the *Markov property* means that the agent can predict the next state and the expected reward from the current state. This is important in RL because decisions and values are functions of the current state. In RL this may not always be the case, but it is sufficient that the state is a good enough approximation. A RL task that satisfies the Markov property is called a *Markov Decision Process (MDP)*. If

state and action sets are finite, then the RL task is a finite MDP. The MDP framework provides a model for *Single Agent Reinforcement Learning* (SARL) as shown in Figure 4.1. The *policy* is the mapping from a given state to an action. It is a stochastic rule by which an agent selects an action as a function of a state (*action-selection*). The learner or agent is searching for the optimal policy where the expected return is greatest. There may be more than one optimal policy. The basic RL algorithm is given in Figure 4.2.

```
Create initial state set; Estimate the value of each state;
Repeat (for each episode)
        Initialize current state;
        Repeat (for each time step t of the episode)
                Select an action;
                Take action;
                Observe reward;
                Update the estimated value of the state;
                Retrieve next state;
        Until t is terminal
Until end of episode
```

**Figure 4.2 The basic reinforcement learning algorithm learning by experience**
The first step is to determine which state to begin in. At every time step, the agent selects an action based on the current state; then takes the action and receives a reward. Rewards are immediate and are given directly by the environment. What is good in the long run, is specified by the value of a state. Roughly speaking, the value of a state is the total amount of reward an agent can expect starting from that state. The values of the states are updated from observations an agent makes over its lifetime, hence an agent learns from experience. Finally the next state is selected (adapted from the text of Sutton and Barto, 1998).

The agent's action is rewarded at each time step with a reinforcement signal by the *reward function*. The reward function is the process that generates an immediate reinforcement signal to each action the agent takes from each state. This function is out of the agent's control, hence exists outside the agent. The agent is rewarded in terms of its goal. The reinforcement signal defines the

agent's goal, hence the reward must indicate what is to be accomplished, not how to achieve it. In RL the agent's goal is to maximize its rewards in the long term. This means being able to represent the value of future rewards now. As discussed in Section 4.2, RL uses the *discount rate* to do this. A discount rate of zero means that the agent is myopic, i.e., the agent chooses an action at time $t$ to maximize the reward at time $t$+1. As the discount rate approaches 1, the agent becomes more far-sighted, i.e., the agent chooses an action at time $t$ to maximize future rewards at some later time step $t+n$ where $n>1$. The discount rate can be viewed as the relative value of delayed *versus* immediate rewards. Self-control is one of many life situations where we apply the concept of discounting. The expected return of discounted future rewards $R$ is given in Eq. 4.1:

$$R = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \qquad (4.1)$$

where $\gamma$ is the discount rate, $r$ is the reward at each time step and $T$ is the terminal time step or infinity. The expected return of future rewards is used in the *Value Function* to enable the agent to estimate or predict what action to take. There are two types of Value Functions:

1. *V(S)* is the state-value function for a policy, i.e., it gives the expected return when starting in a state whilst following a given policy.
2. *Q(s,a)* is the action-value function for a policy, i.e., it is the value of taking an action $a$ in state $s$ whilst following a given policy.

## 4.5 Reinforcement Learning methods

RL methods update the agent's policy as a result of its experience. There are three main classes of methods: Dynamic programming (DP), Monte Carlo and Temporal Difference learning (TD). Each have its own strengths and weaknesses, however they all have the basic processes of: *policy evaluation*, which involves estimation of the Value Function and backing up the values of actual or possible states; and *policy improvement,* which updates the Value Function and improves the agent's policy.

In DP the goal is to compute the optimal policies (action-selection) given a perfect model of the environment as an MDP. A model can be defined as a representation of the environment, such that from a given state and action the learner or agent can predict the resultant next state and next reward. A model could be *stochastic*. A stochastic model has several possible states and rewards, each with some probability of occurring. The DP method uses Value Functions to organize the search space for good policies. The basic idea is to use the Bellman equations (1957a) as update rules for approximating the desired Value Functions. DP does this through *policy evaluation* and *policy improvement*. The aim of the *policy evaluation,* otherwise known as the prediction problem, is how to compute the state-value function (*V(S)*). In RL it is the iterative policy evaluation, i.e., how to produce a successive approximation of the state-value function, which is important. The state-value function is approximated by bootstrapping, i.e., estimation of the values of states based on estimates of successor states. In the DP method this means keeping backups of every state, which is computationally expensive. Usually the terminating condition is when the difference between the current state-

value function and the previous state-value function, is small. In the *policy improvement* process, the aim is to determine how to obtain a better policy that improves on the original policy, i.e., having a greater expected return. The *policy iteration* combines evaluation and improvement by truncating the two separate processes to make it less computationally expensive. The DP method is well developed, as it has existed since the late 1950s. The main disadvantage of the DP method is that it requires a complete and accurate model of the environment, which is not always available. In addition, the DP method operates over the entire state set and action set, which despite the fact that it can be improved by generalization techniques, is still computationally expensive when compared to other RL methods. It is also limited by the curse of dimensionality (Bellman, 1957a). This is where the mainly finite number of states grows exponentially with the number of state variables. Littman et al. (1995) provide an excellent summary on DP methods for RL.

The Monte Carlo method is model free, i.e., there is no need to maintain a complete knowledge of the environment. It learns from simulated experience based on episodes rather than discrete time steps of a task. As with the DP method, the Monte Carlo method is based on estimating the Value Function. In the Monte Carlo method the *policy evaluation* averages the returns observed after visits to a given state; over time the average for a state should converge, i.e., not change. The Monto Carlo method does not bootstrap as in the DP method. The main advantage of the Monte Carlo method, as compared to the DP method, is that the Monte Carlo method learns directly from the environment and hence it does not need an exact model of the environment. It

does this by sampling or approximating the states, as opposed to maintaining a complete set of states. This makes it computationally less demanding than the DP method. In order to continue exploring actions, as opposed to exploiting known actions that yield rewards, the Monte Carlo method uses either an *on-policy* algorithm, which evaluates or improves the policy whilst using it, or an *off-policy* algorithm, which separates the policy into an estimation policy for improvement and a behaviour policy to perform the improvement. Monte Carlo methods for RL are still in their infancy. The main criticism arises from the fact that they operate in terms of episodes rather than time steps and hence they must wait till the end of an episode, which may last many time steps, before updating their Value Function.

The TD method brings together the advantages of the Monte Carlo method and the DP method. It is model free as with Monte Carlo method and it bootstraps like DP, i.e., it bases estimates on previous learned experiences. The policy evaluation uses experience to update the estimates of the Value Function as shown in Eq. 4.2 (Sutton and Barto, 1998):

$$V(S_p) = V(S_p) + \alpha\left[r_c + \gamma V(S_c) - V(S_p)\right] \qquad (4.2)$$

where $V(S_p)$ on the right hand side is the state-value function of the state for the previous time step, $\alpha$ is the step-size parameter, which is sufficiently small, $r_c$ is the reward received at the current time step, $\gamma$ is the discount rate, and $V(S_c)$ is the state-value function of the state for the current time step. The value of the previous state is updated based on the current state, i.e., it bootstraps and it needs only to wait till the next time step to update the Value Function, whereas the Monte Carlo method has to wait till end of an episode.

There are various TD methods. The Q-learning *off-policy* algorithm approximates the optimal action-value function (*Q(s,a)*) without using the policy. Specifically Q-learning learns about the greedy policy while following a policy that explores, i.e., it selects nongreedy actions. The actor-critic method is an *on-policy* TD method. The policy is the action-selection process, and the critic criticizes the actions of the actor. It is an *on-policy* method, as the critic must learn the reinforcement signal. It is also a reinforcement learning comparison method, as each action is ranked. Before the development of Q-learning, the actor-critic method was the TD method of choice for much of the research in reinforcement learning. The TD method is the most widely used reinforcement learning method as it is simple, it operates time step by time step and it is computationally less expensive than other RL methods.

The basis of all TD methods is how to distribute credit or blame to the actions, which have produced the eventual reward. This is defined as the *Temporal Credit Assignment* problem, i.e., how to distribute credit (or blame) for success among the many decisions that may have been involved in producing it (Minsky, 1961). In TD methods, the *TD error* $\delta_t$ is calculated according to Eq. 4.3 and then assigned to the states responsible:

$$\delta_t = r_{t+1} + \gamma V(S_{t+1}) - V(S_t) \qquad (4.3)$$

where time $t$ is the current time step, $r_{t+1}$ is the reinforcement signal at time $t$+1, $\gamma$ is the discount rate of future rewards, $V(S_{t+1})$ is the state-value function at time $t$+1 and $V(S_t)$ the state-value function at time $t$. The basic mechanism

for temporal credit assignment in RL is the eligibility trace $\lambda$. The basic idea is that when a TD error occurs, only the eligible states are assigned credit or blame for it; earlier states are given less credit for the TD error. An eligibility trace can be accumulating in that it accumulates each time a state is visited and then fades away gradually when the state is not visited. Eq. 4.4 calculates the eligibility trace for a state $e_t(s)$. If the state has been visited, then it is incremented by a value of 1, otherwise its eligibility decays over time:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & s \neq s_t \\ \gamma\lambda e_{t-1}(s) + 1 & s = s_t \end{cases} \qquad (4.4)$$

where $\gamma$ is the discount rate, $\lambda$ is the trace-decay parameter with a value between zero (representing pure bootstrapping) to 1 (representing pure nonbootstrapping), $e_t(s)$ is the eligibility trace for state $s$ at time $t$. Eq. 4.5 calculates the change to the state-value function $V(s)$ for recently visited states:

$$\Delta V(s) = \alpha \delta_t e_t(s) \qquad (4.5)$$

where $\alpha$ is the step-size parameter, $\delta_t$ is the TD error given by Eq. 4.3 and $e_t(s)$ is the eligibility trace for the state $s$ at time $t$ given by Eq. 4.4. Eq. 4.2 is the Value Function for a special case of TD(λ), TD(0), where only one state preceding the current one is changed by the TD error in contrast to TD(λ), which selects all eligible states to be changed by the TD error. From the basic RL algorithm presented in Figure 4.2, we derive the complete algorithm for TD(λ) as given in Figure 4.3. Any TD method can use the eligibility trace $\lambda$, although it has been found that the eligibility trace is not as effective with Q-learning. This is because cutting off traces at exploratory points, as opposed to

the end of an episode, reduces Q-learning's effectiveness (Rummery, 1995). Using eligibility traces requires more computational resources, but has proven to result in faster training especially in assigning delayed reward (Sutton, 1988).

For all states; For all time steps $t$
    Initialize $V(s)$ and $e_t(s) = 0$
Repeat (for each episode)
    Initialize $s_t$;
    Repeat (for each time step $t$ of the episode)
        Select action;
        Take action;
        Observe reward;
        Next state $s_{t+1}$;
        $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$
        $e_t(s_t) = e_t(s_t) + 1$
        For all $s$:
            $V(s) = V(s) + \alpha\, \delta_t\, e_t(s)$
            $e_t(s) = \gamma\, \lambda\, e_t(s)$
        $s_t = s_{t+1}$;
    Until $s$ is terminal
Until end of episode

**Figure 4.3 The TD(λ) reinforcement learning algorithm**
Pseudo-code for TD(λ) RL algorithm. Increments are performed on each time step until the end of the task making it an online algorithm. The first step determines which state to begin in. At every time step, the agent selects an action based on the current state; then takes the action, receives an award, and determines the next state $s_{t+1}$. The TD error is calculated using the state-value function of the previous state $V(s_t)$ and the state-value function of the current state $V(s_{t+1})$. The eligibility trace for the current state is incremented by 1 to denote that this state has been visited in this step. For this time step we need to assign the TD error to each previous state denoted by the state's eligibility trace. This is done by iterating through all states and the TD-error is then used to update all recently visited states denoted by their nonzero eligibility trace $e_t(s)$. $\alpha$ is the step-size parameter, $\delta_t$ is the TD error given by Eq. 4.3 and $e_t(s)$ is the eligibility trace for the state at time $t$ given by Eq. 4.4. The eligibility trace for all states decay by $\gamma$ multiplied by $\lambda$. Finally the next state $s_{t+1}$ becomes the current state $s_t$, and the agent moves onto the next time step. (adapted from Sutton and Barto, 1998).

## 4.6 Reinforcement Learning and Function Approximation Techniques

With some tasks/problems it is not feasible to have a table with an entry for each state (*V(s)*), or state-action pair (*Q(s,a)*). The problems are that firstly more memory is required to hold the large tables, secondly more processing time is required, and finally the data becomes noisy. It may be impossible to have a complete description of all the possible states, or alternatively one may have just partially observed information about the states, which can be modified through learning over time. *Function approximation* is a technique used to overcome these problems. The function approximation method used with RL has to be able to deal with the distinguishing features of RL. These are: the delayed reward, the temporal credit assignment problem, the exploration *versus* exploitation problem, the possibility of partially observable states, and life long learning. The aim of function approximation for RL is to teach a policy to output an action from a given state. Learning in RL is in real time, i.e., the function must be able to change during the lifetime of the task. In addition, the function needs to learn incrementally. For example, in RL the agent is learning the optimal policy while the policy is changing. Even if the policy stays the same, the target values of the state may change with bootstrapping as in the RL methods TD and DP.

## 4.7 Gradient Descent and Artificial Neuron Learning

The *gradient descent method* is possibly the most widely used of all function approximation methods and is technique well suited to RL due to the fact it learns incrementally after each input. The aim of the gradient descent method

is to minimize the error between the actual observed output and the target (or desired) output, over a set of inputs. The implementation of the gradient descent method requires the definition of an error function. The sum of squared errors $\varepsilon$ (sometimes referred to as mean-squared-error) is usually used, given in Eq. 4.6:

$$\varepsilon = \sum_{i \in S} (t_i - a_i)^2 \qquad (4.6)$$

In RL the set of inputs is the set of states $S$, $t_i$ and $a_i$ are respectively the target and actual output for input state $i$.

An *Artificial Neural Network* (ANN) is well suited to problems with noisy (errors), complex and incomplete data. An ANN is based on a simplified version of a biological neuron and its basic functionality is based on the workings of neurons in the human brain. The processing is highly parallel with distributed representation, i.e., information is distributed over many units. Each unit of the ANN receives inputs, which are adjusted by the numerical weights connecting them to the other units. The weights represent the knowledge of the ANN. These are usually generated randomly and are then adjusted during learning. An important feature of the ANN is that it can apply what it has learnt to previously unseen examples. This ability is known as *generalisation* and occurs as the ANN detects features of the input that it has learnt to be significant and therefore represented in the weights. The ability of ANNs to learn through experience means they can generalise to situations or experiences, which are similar, but not necessarily identical.

**An ANN's Topology.** Depending on the nature of the problem, the ANN is organized in different arrangements (topologies). Figure 4.4 shows the topology to be used in this thesis. Figure 4.4 shows a three layered feed-forward structure. A typical multi-layer feed forward ANN is composed of interconnected nonlinear units called *nodes*, i.e., the unit's output is a nonlinear function of its input. Typically multi-layer feed forward ANNs are interconnected in layers to form a direct acyclic graph with a layer of input nodes and one or more layers of hidden nodes, and a layer of output nodes.

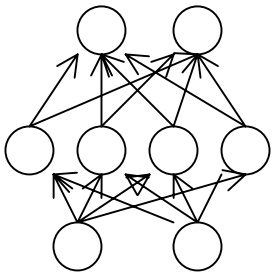


**Figure 4.4 Topology of ANNs used in this thesis**
This ANN is a three layered feed-forward structure with one input layer of two nodes, one hidden layer of 4 nodes and an output layer of two nodes. Trainable connections are represented as solid lines.

The hidden nodes function as feature detectors, recording the basic inputs so that the ANN can learn the required features appropriate to the task. This recording or *internal representation* is critical to the functioning of the ANN. The function of the hidden nodes is to form the decision boundaries. With enough hidden nodes it should be possible to form internal representations of any input pattern so that the output nodes are able to produce an appropriate response from a specific input. However, there are no clear rules governing the network topology. The number of inputs and outputs features of the task

determines the number of input nodes and output nodes respectively. There is no simple solution, as to the number of hidden nodes and how they should be arranged in layers. It is up to the user to experiment in order to find the optimal configuration. The *Kolmogorov theorem* (Kolmogorov, 1957) as discussed in Kurkova (1992) proves that two hidden layers are theoretically capable of separating any classes, but the topology of hidden nodes within those layers must be decided through experimentation and analysis, although evolutionary techniques have proved useful in optimizing the topology of ANN (refer to Chapter 5 for further details).

**Activation Functions.** An artificial neuron implements a nonlinear mapping from a set of input values to its output. Each input to the neuron is associated a *weight* to strengthen or deplete that input. The artificial neuron computes a linear function of its inputs and then uses an *activation function* to compute its output. The output is further influenced by a threshold value, also referred to as the *bias* ($\theta$). Every node in each layer calculates its activation as the sum of each input multiplied by the weights from the nodes to which it is connected. The net input for the node *j* is given in Eq. 4.7 (McCulloch and Pitts, 1943):

$$net_j = \sum_{i=0}^{n} w_i o_i \qquad (4.7)$$

where $w_i$ is the weight between nodes *i* and *j*, and $o_i$ is the output from node *i* to node *j* (or the input to node *j*). The activation function determines the output for the node from the net input and the bias. There are different types of activation functions that can be used. The activation function used in this thesis is the *Sigmoid function.* The total weighted inputs ($net_j$) are fed through

the Sigmoid function in Eq. 4.8 to give the Sigmoid output for node *j* (Rumelhart et al., 1986):

$$o_j = \frac{1}{1 + e^{-\lambda (net_j - \theta)}} \qquad (4.8)$$

where $\lambda$ is the slope parameter that controls the steepness of the Sigmoid function and is normally given the value of 1. Here the bias $\theta$ can be considered as having an input of –1 with a weight of $\theta$. Figure 4.5 shows the Sigmoid function for a bias ($\theta$) of zero.

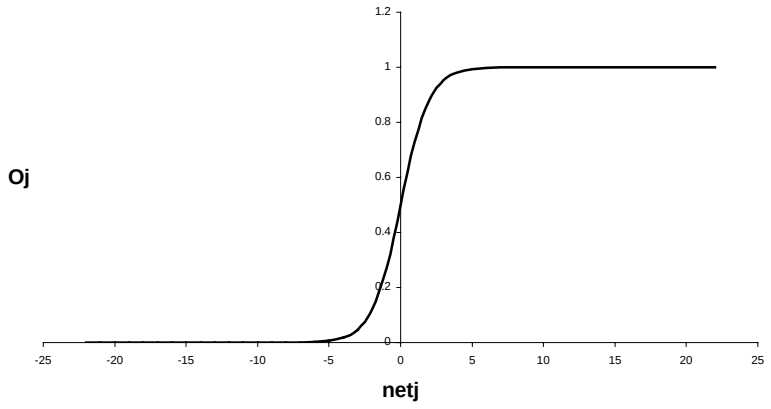



**Figure 4.5 Sigmoid Function for a bias of zero**
The Sigmoid output, given by Eq. 4.8, for a node *j* with a bias of zero

A schematic diagram of an artificial neuron equivalent of a biological neuron is shown in Figure 4.6. The artificial neuron computes a weighted sum of its inputs from other neurons and then adds its bias. Whether the neuron fires depends upon whether the sum is above or below the bias as discussed above.

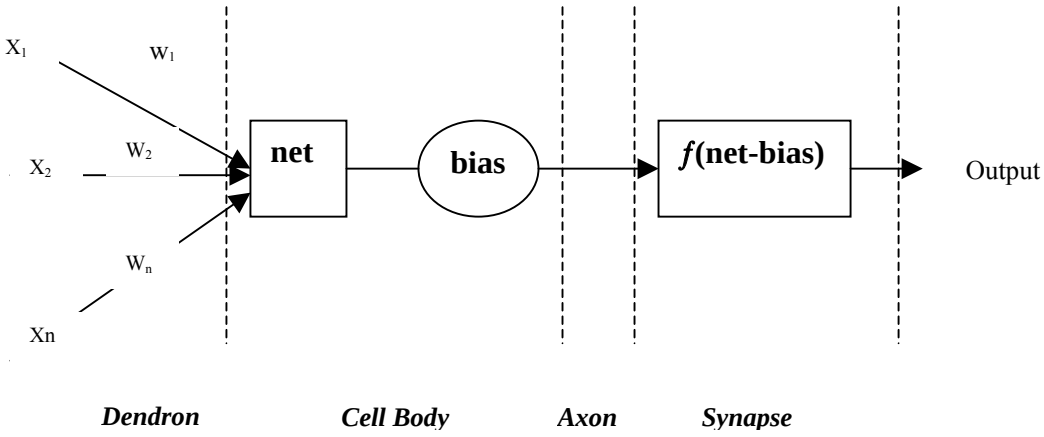



**Figure 4.6 Schematic diagram of an artificial neuron equivalent of a biological neuron**
The artificial neuron computes a weighted sum of its inputs from other neurons and then adds its bias. The neuron fires depending on whether the sum is above or below the bias (based on Figure 14.3 p.418 Konar, 2000).

The Sigmoid function satisfies the requirements for gradient descent in that the output is a non-linear function of its input and is differentiable. All the outputs are then fed into the nodes in the subsequent layer or just output if it is the output layer.

Gradient descent is possibly the most widely used method for training Artificial Neural Networks. The error is calculated by the *delta rule* or the *least-mean-square* (LMS) rule, otherwise known as the Widrow-Hoff rule (Widrow and Hoff, 1960), given in Eq. 4.9:

$$\Delta w_i = \eta \left( o_t - o_a \right) x_i \qquad (4.9)$$

where $\eta$ is the learning rate (the step size in the negative direction of the gradient of the error $\varepsilon$ Eq. 4.6); the optimal value is usually found by experiment, $o_t$ is the target output, $o_a$ is the actual output and $x_i$ is the $i$th input.

The gradient-descent algorithm in Figure 4.7 uses the stochastic gradient-descent rule. In the stochastic gradient-descent algorithm the weights are updated incrementally after each input as opposed to the end of the complete set of inputs.

Initialize each weight $w_i$ to some small random value
Repeat
    Initialize $\Delta w_i$ to zero
    For each input do
        Present input $x_i$
        Compute actual output $o_a$
        $w_i = w_i + \eta(o_t - o_a)x_i$
until termination condition is met

**Figure 4.7 The gradient descent algorithm**
The stochastic or incremental gradient descent algorithm updates weights after every input as opposed to the end of the input set; $\eta$ is the learning rate, $o_t$ is the target output, $o_a$ is the actual output and $x_i$ is the $i$th input. The weights are updated using the Widrow-Hoff rule (Widrow and Hoff, 1960).

The aim of the gradient descent method is to adjust the weights by a small amount in the direction that would most reduce the error (ε in Eq. 4.6), i.e. the global minimum, as opposed to a local minimum. This is illustrated in Figure 4.8.

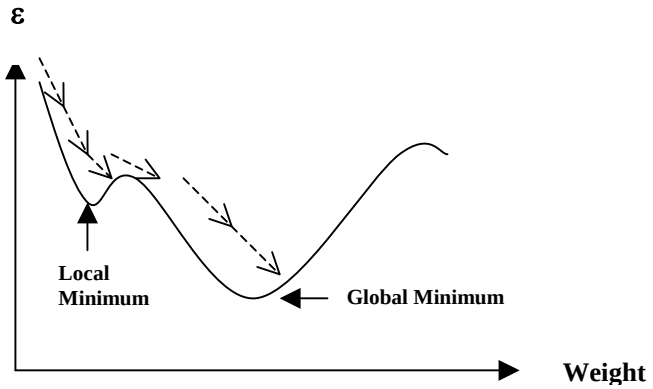



**Figure 4.8 Gradient Descent method illustrated**
The aim of the gradient descent method is to find the weight values that minimize the error $\varepsilon$ (Eq. 4.6), i.e., to find the global minimum, as opposed to the local minimum shown above.

The values for the weights $w_i$ and the bias ($\theta$)[3] are derived through learning. The backpropagation is a non-linear gradient-descent method, which trains the multi-layer perceptron artificial neural network. The *backpropagation* learning algorithm (Werbos, 1994; Rumelhart et al., 1986) is the most common ANN learning technique. Backpropagation, as used in training a multi-layer neural network, consists of two passes through the whole network. In the forward pass, a set of input data is presented to the network and the output of the neurons calculated, from input layer to output layer, without changing the weights. The resulting output is compared to an expected output and an error value calculated. This is used in the second pass, the backward pass, which sends information about the error back through the same neuron connections that sent the activation forward. Figure 4.9 shows the stochastic gradient-descent version of the backpropagation algorithm for a feed forward network, as described above with one layer of hidden nodes. Following steps 1 to 5 in Figure 4.9 is called an *epoch*. A number of epochs are required for *training* the ANN. The ANN is said to be trained when some performance criteria is met. The backpropagation learning algorithm (Rumelhart et al., 1986), as used in supervised learning, requires a "teaching output" that has to be supplied by an external "teacher". An error is calculated based on the difference between the actual output and the teaching output referred to as the target output. This is the propagated backward through the ANN as per Figure 4.9.

[3] The bias could be implemented as an extra input having always an input of –1 and a weight *w* that is modified like all the others

Create a FeedForward network with $n_i$ inputs, $n_h$ hidden units, and $n_o$ output units;

Initialize all networks weights to some small random value, where the weight $w_{ij}$ denotes the weight from unit $i$ to unit $j$;

Repeat

For each input $x$ do

1. Present input $x$ to the network
2. Compute actual output $o_u$ of every unit $u$ in the network
3. For each output unit $k$ calculate its error term $\delta_k$ using the target output $t_k$

$$\delta_k = o_k(1- o_k)(t_k- o_k)$$

4. For each hidden unit $h$ calculate its error term $\delta_h$ as there is no target output use the error term $\delta_o$ of each output unit $h$ feeds into

$$\delta_h = o_h(1- o_h)\Sigma w_k \delta_k$$

5. Update each network weight $w_{ji} = w_{ji} + \eta\delta_j x_i$

until termination condition is met

**Figure 4.9 Backpropagation with the gradient descent algorithm**
The stochastic gradient-descent version of the backpropagation algorithm for a feedforward network. An integer is assigned to each unit of the network. $x_i$ denotes the input from node $i$ to node $j$, $w_{ji}$ denotes the weight from unit $i$ to unit $j$. $\delta_j$ denotes the error term associated with the input unit $j$, $\delta_h$ denotes the error term associated with unit $h$, $\delta_o$ denotes the error term associated with the output unit $o$. Steps 1 and 2 propagate the input forward, steps 3, 4, and 5 propagate the error backwards through the network (adapted from Rumelhart, 1989).

The fact that RL is goal directed with no explicit target output would seem to limit the applicability of the backpropagation algorithm as a training method for RL. However in Chapter 6, which is concerned with implementing RL in an ANN, we see how the error signal is constructed using the reinforcement signal.

## 4.8 Reinforcement Learning and Neuroscience

Reinforcement Learning is not a term used in psychology. The terms of "reinforcement" and "reinforcement learning" were first introduced in the engineering literature in the 1960s (Waltz and Fu, 1965; Mendel, 1966). Klopf

(1972, 1975, 1988) would seem to be the first to link the ideas in RL such as trial and error learning with animal learning in psychology. There have been many recent neuroscience models that use the actor-critic TD method in RL as a psychological model for classical conditioning learning in animals (Barto, 1995; Suri and Schultz, 1999; Holroyd and Cole, 2002). Some researchers argue that the actor-critic model is too simplistic (Brown et al., 1999). Barto (1995) himself warns of relating abstract systems to the animal nervous system. The criticism aside, much of the concepts in the actor-critic model can be related to conditioning in animal learning. For example, the state-value function (*V(s)*) provides a mechanism for predicting future rewards, and the TD error refines this prediction. In order to be able to calculate *V(s)* and the TD error, the model uses approximation; these are then refined and updated with each time step of the trial. Research to date suggests that this is exactly what happens in classical conditioning learning in animals. For example, Cohen et al. (2002) have found that dopamine activity serves as reinforcement signal that indicates a mismatch between prediction and the delivery. This then updates the associative (Hebbian) synaptic connections to reduce subsequent prediction errors.

Before describing RL in the context of animal learning, it is relevant at this point that we review some definitions from animal learning in psychology (based on Morgan et al., 1979). There are two types of conditioning learning: *classical (pavlovian) conditioning* where the temporal reinforcers are delivered independently of the animal's actions and *instrumental* conditioning where the actions of the animal determines the reward or punishment. The

essential operation in classical conditioning is the pairing of the two stimuli. One stimulus is called the *conditioned stimulus* (CS). This is also referred to as the neutral stimulus because it does not specifically produce a response in the animal. The second stimulus is called the *unconditioned stimulus* (US). This stimulus produces a response known as the *unconditioned response* (UR). The time between the CS and the US is known as the *interstimulus interval*. As a result of being paired with the US a number of times and with the right interstimulus interval, the CS produces a response similar to the UR, which is called the *conditioned response* (CR). The experiments of Ivan Pavlov formed the basis of classical conditioning learning. Pavlov (1927) designed an experiment for measuring how much a dog's mouth waters (salivates) in response to food. The dog was placed in a soundproof room. Pavlov sounded a bell (the CS), and then shortly afterwards presented the food (the US), and the amount of saliva secreted (the UR) by the dog was measured. After presenting the food and bell a few more times he then presented just the bell. The saliva secreted (CR) increased as more conditioning took place (Morgan et al., 1979). TD learning explains the association of the CS with the reward (US) by the expected return of future rewards (Eq. 4.1). The effect of the *interstimulus interval* (the delay) on the association of the CS to the US is modelled by the discount rate in TD learning. The TD error in RL is associated with the activity of dopamine cells in the midbrain (Doya, 2000; Cohen et al, 2002; Dayan and Abbot, 2002; Dayan and Balleine, 2002). This is verified by evidence from studies on drug addiction and self-stimulation experiments on rats. A controversial area of research is the exact region of the brain affected by dopamine, and the role

dopamine plays in animal conditioning learning. Dayan and Balleine (2002) suggest that too much emphasis has been placed on dopamine and the TD error and research needs to focus attention elsewhere, if we are to understand what biological mechanisms are involved in classical conditioning learning. In addition, the role dopamine plays in motivation and habit behaviour is unclear. It has been suggested that dopamine is concerned with motivation behaviour and not with enjoyment (Brown et al., 1999). Motivation and habit both play a part in sequential decision-making, but to what extent is still disputable (Cohen et al., 2002; Dayan and Balleine, 2002; Brown et al., 1999). Figure 4.10, adapted from Doya (2000), summarizes what is currently known in relating RL to the central nervous system. Figure 4.10 highlights the fact that the Basal Ganglia, previously associated with motor control, is now believed to play a part in reinforcement learning, as suggested in studies by Holroyd and Coles (2002), Cohen et al. (2002) and Beiser and Houk (1998).

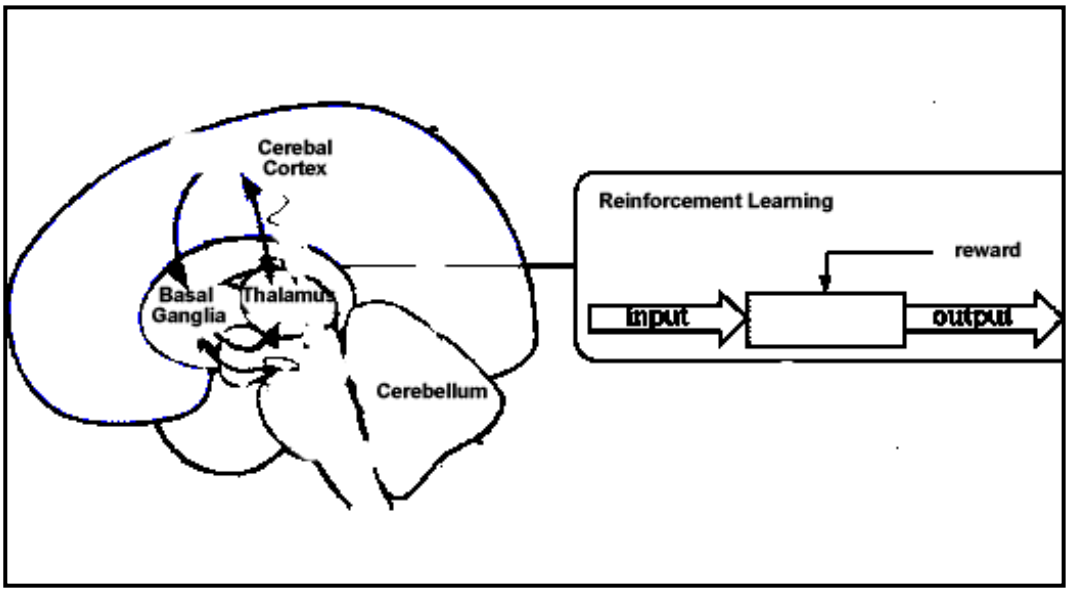



**Figure 4.10 Neural model for reinforcement learning**
Illustrating the role of the Basal Ganglia, (previously associated with motor control), in *reinforcement learning.* There is no explicit target output, but a reward signal that notifies how good or bad the output, (or a sequence of outputs) is (adapted from Doya, 2000).

Figure 4.10 also serves to suggest that further progress still needs to be made if a more comprehensive and neurologically plausible model is to be formulated.

### 4.8.1 Summary

Although RL is an engineering term, it has more recently been linked with classical conditioning learning in psychology. Specifically the actor-critic model, a TD method, has been used as an abstract model of the concepts in classical conditioning learning, which stems from the research by Pavlov (1927). In summary, a conditioned stimuli (the bell in Pavlov's experiment) produces a conditioned response similar to the unconditioned stimuli (food). Relating this abstract model to the neurological mechanisms that drive these responses is an active area of research in neuroscience with general consensus, that the TD error is associated with the dopamine activity in the midbrain, but the exact level and the role dopamine plays in animal conditioning learning is still questionable. What we do know is that dopamine activity plays a role in motivation and habit, both associated with self-control behaviour. RL and self-control have the same characteristics of goal directed behaviour and discounting. For example, in self-control greater discounting of future rewards implies that one becomes more myopic, which is reflected in a discount rate approaching zero in RL.

## 4.9 Concluding Remarks

The RL method to be used in this thesis is TD learning as it combines the benefits of both DP and TE learning. TD does not require a complete model of

the environment. TD deals with discounting of future rewards, by using the discount rate, and can assign blame to pass actions, i.e., *the temporal credit assignment problem* using the eligibility trace TD(λ). In this thesis, TD is implemented as TD(0), as only one state preceding the current one is changed by the TD error. The general-sum games used in this thesis are particularly simple and each state and action pair can be represented in a lookup table. ANNs are used in this thesis to learn the action-selection function. ANNs are appropriate as they learn throughout the lifetime of the task and learn incrementally, i.e., they reward or penalize at each time step (round). ANNs can generalize to new situations similar to states presented previously. ANNs in this thesis are implemented as a multi-layer perceptron-like, i.e., non-linear nodes, feed forward networks. The ANNs are trained with backpropagation, which traditionally is used in supervised learning with a target or desired output. In RL, the reinforcement signal is used to construct this desired output. On a final note, RL as presented in this chapter, falls neatly into the framework of single agent reinforcement learning. In Chapter 6 we will extend RL to multi-agent learning, which is the framework for our 2-ANNs model presented in Figure 3.3 in Chapter 3. The next chapter presents Genetic Algorithms, which is the final technique to be used in this thesis.

# Chapter 5

# 5 Review of the Concepts for Evolutionary Adaptation of the Neural Model

## 5.1 Chapter Outline

This chapter brings together the three main techniques: Reinforcement Learning, Artificial Neural Networks and Genetic Algorithms. It starts with a review of evolutionary computation techniques. It then goes on to explain why the Genetic Algorithm is the evolutionary computation technique of choice for this thesis. It concludes by combining all three techniques and describes how they will be used in this thesis.

## 5.2 An Overview of Evolutionary Computation

Evolutionary Computation, sometimes referred to as Evolutionary Algorithms, consists of three main techniques: Genetic Algorithms, Evolutionary Programming and Evolutionary Strategies. Each method emphasizes a different facet of natural evolution. *Genetic Algorithms* emphasize the genetic changes to the individual. In Genetic Algorithms the individual is typically represented as a bit string. *Evolutionary Programming* focuses on the processes that yield behavioural changes within a group. *Evolutionary Strategies* focus on the behavioral changes of the individual. All of the techniques begin with a population of individuals. In natural evolution, the DNA provides a set of instructions on how to make an individual. The DNA can be thought of as a string of genes. This genetic information is called the *genotype* of the individual in contrast to the *phenotype*, which is the

physical manifestation (physical embodiment) of the individual. The genotype sets the individual apart from other individuals. The process of natural evolution frequently involves sexual reproduction, which is basically a mixing and shuffling of genes. Before this, some sort of selection process must have occurred. In evolutionary computation selection is much simpler than sexual selection. Usually an evolutionary computational method weeds out the worse performing algorithms by selecting the fittest for further breeding, where fitness is defined by how successfully the individual performs a particular task, e.g., finding the optimal value of a function. The fittest individuals are selected to become parents of offspring that form a new generation through recombination (the exchange of genes). *Recombination* is the process that takes the genetic information from the parents for the offspring.

All Evolutionary Computation methods have two critical design decisions: (i) how to represent the individuals in the population, and (ii) what evolutionary process to use?

The *Genetic Algorithm* (GA) is an evolutionary algorithm technique developed by Holland (1975), which is modelled on genetic evolution. It works on a population of individuals represented as strings of genes. A gene can be represented as a bit string. Selection of those individuals to go on to the next generation is by a *fitness function*. Conceptually, how well an individual performs in a task can be thought of as the *fitness function*. Selection of individuals for reproduction is based on the individual's fitness. Generally the individuals that perform better produce more offspring. The offspring for the next generation are produced when two individuals of the population come

together. Reproduction involves taking bits from each parent to form a new individual (generally referred to as *crossover*). *Mutation* is then applied to the resulting population. This is infrequent; one in one thousand individuals will be *mutated,* i.e., one of his genes has been randomly altered. The combined effects of crossover and mutation mean that GAs can produce offspring that are very different from their parents. Because each offspring is different this makes for a diverse population and hence a diverse generation. This diversity in the populations and generations reduces the likelihood of the usual sort of problems associated with premature convergence such as the *local minima problem (LMP)*, where a minimum error is not necessarily the global minimum.

*Evolutionary Programming* (EP) was developed by Fogel et al. (1966). The aim is to solve a problem through simulated phenotypic evolution. The representation of the individual is problem domain dependent. For example, in the travelling salesman problem, the individuals of the population are ordered lists and for optimisation problems the individuals of the population are real values. Thus, in EP it is easy to see how the representation of the individual links to the behaviour of the individual, as opposed to GAs. To create the next generation in EP, both offspring and parents solve the problem. The resulting solution is evaluated against a set of possible solutions. They get a score, which is the individual's *fitness*. The fittest individuals form the next generation. This can be described as an *elitist cull*. The fitness is sometimes calculated as the behavioural error. The *behavioural error* is the difference between the optimal behaviour (specified by the set of possible solutions) and

the actual observed behaviour. A specified number of those individuals with the lowest behavioural error go onto the next generation. EP has a heavy reliance on mutation, as there is no recombination operator, for which EP has been criticized (Goldberg, 1989). The basic algorithm for EP is given in Figure 5.1:

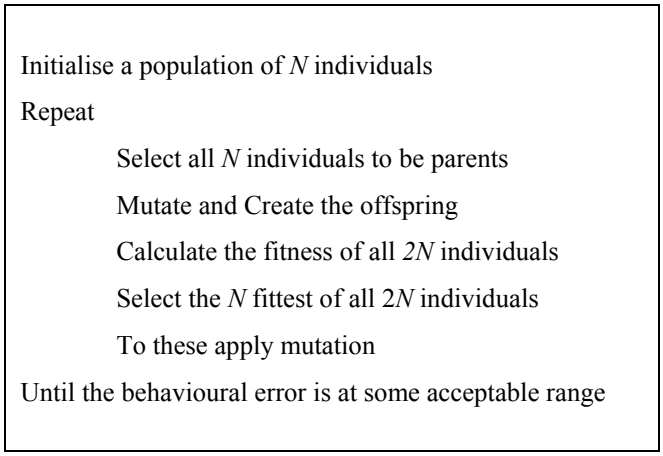



**Figure 5.1 The basic algorithm for evolutionary programming**
All of the individuals in the initial population are selected to be parents. From the resulting population only the fittest are selected for mutation. The resulting population is then evaluated. The fitness function measures the behavioural error, defined as the difference between the optimal behaviour and the actual observed behaviour.

*Evolutionary Strategies* (ES) were developed by Rechenberg (1973) and expanded to include more than one individual in the population by Schwefel (1981). Individuals are generally represented as strings of real numbers. A new population is created by selecting the best solutions, be it parents and their offspring, or just the offspring, and applying recombination. In ES the offspring can be produced from more than two parents. ES place a heavier reliance on mutation than GAs. Mutation is applied to the resulting new population. The change to the genotype, as a result of mutation, is very small, as it alters the gene, which is represented as a real number, only a little. GAs

in contrast, invert the whole value of the gene. This means that ES results in a series of little hill climbs towards the optimal solution. The change to the individual, as a result of mutation, is only accepted in future generation, if the fitness of the individual is improved, as compared to the fitness of the individual without mutation.

## 5.3 Which Evolutionary Process is best for the work of this thesis?

In this thesis, a top-down approach has been used in the design of the model of Figure 3.3 of self-control behaviour. The suggestion is that the functions associated with the higher brain system, (i.e., rational thought) and the functions associated with the lower brain system, (i.e., instinctive behaviour) (Jacobs, 1999), are locked in some form of internal conflict for control of the organism and therefore its behaviour. The overall aim is to build a computational model that can help to guide research on the biophysical processes that underlie the mechanisms suggested by the functional analysis of this more abstract model. In order to do this, we need a model of the evolution of self-control through precommitment behaviour that makes some attempt to be both biologically and psychologically relevant. Therefore, it is paramount that the techniques used in the model emulate the biophysical mechanisms underlying this complex behaviour. Of the three main techniques of Evolutionary Algorithms (EP, ES, and GAs), only GAs are concerned with the evolution of the individual using a near true simulation of natural evolution. EP and ES are both concerned with the evolution of behaviour and the fitness functions for EP and ES reflects this emphasis on behaviour. It is the view of this thesis that EP is less biologically plausible than GAs for the following reasons: (i) in EP the representation is dependent upon the problem,

for example, for the travelling salesman problem the genotype is an ordered list. Thus representation of the genotype in EP and ES is generally non-binary and hence requires special genetic operators, as the binary operators for mutation and crossover cannot be easily deployed; (ii) EP does not use a recombination operator, for example, crossover. It is accepted that crossover is an operator used in genetic evolution. Although the importance of crossover is under discussion (Eshelman and Schaffer, 1993), in this thesis we are concerned with simulating genetic evolution and should utilize all genetic operators used in genetic evolution. It is the view of this thesis that ES is less biologically plausible than GAs for the following reasons: (i) when ES was initially implemented, it had only one individual in the population (Rechenberg, 1973), which deviates from natural evolution and (ii) in ES the offspring can be produced from more than two parents, which is biologically implausible. In addition, the exploration *versus* exploitation problem, which is a distinguishing feature of reinforcement learning as discussed in Chapter 4, is addressed by using crossover. The reason for this is the following: as a result of crossover the offspring may be significantly different to their parents, which transpires in a search algorithm, which explore new domains. For all of the above reasons it is believed that GA is the best EA for the work of this thesis.

## 5.4 Implementation of Genetic Algorithms

The inspiration for GAs comes from a desire to emulate the mystery of natural evolution. It is thought that by harnessing the mechanisms of natural evolution, solutions may be developed to complex real-world problems even though a full understanding of the how and why may elude the researcher.

### 5.4.1 Genetic Operators

#### 5.4.1.1 Crossover

The crossover operator in a GA mimics biological reproduction (Holland, 1992). Crossover produces two new offspring from two strings. The offspring do not replace their parents, instead they replace individuals with low fitness levels. Examples of crossover operators are: single-point crossover, two-point crossover and uniform crossover. In the *single-point crossover*, the parent strings line up and a point along the strings is selected at random (the *crossover point*). Two offspring are created; the first containing the first bits up to and including the crossover point of one parent followed by the remaining bits of the second parent and the second containing the bits following the crossover point from the first parent and the first bits up to and including the crossover point of the second parent. This is implemented by a crossover mask consisting of all ones up to and including the crossover point, followed by all zeros, as in Figure 5.2. Single-point crossover was the method used in the original application of Genetic Algorithms by Holland (1992). In *two-point crossover*, the crossover mask contains a string of zeros followed by the necessary number of ones, padded out by the necessary number of zeros to complete the string. For example, in Figure 5.2 the middle four bits are substitutes into the second parent. The *uniform crossover* takes bits uniformly from each parent. For example, in Figure 5.2 the offspring are created by taking the first two bits from one parent the next two bits from the other and so on. Crossover can be viewed as exchanging information between individuals of the population.

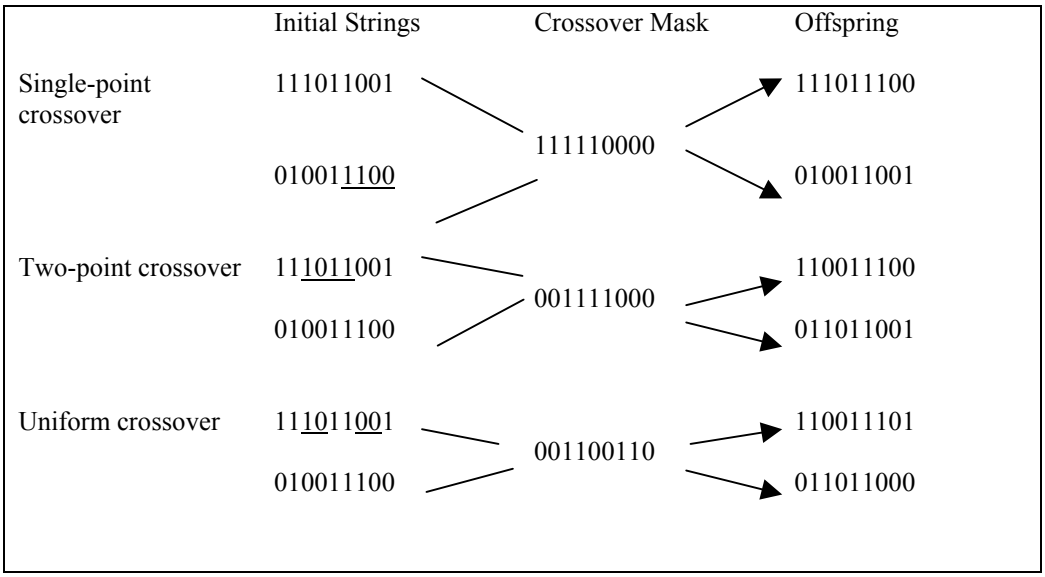



**Figure 5.2 Types of crossover operator for genetic algorithms**
Typical types of crossover operator for genetic algorithms. The crossover operator forms two new offspring from two parent strings by copying selected bits from each parent. It uses a crossover mask to determine which bit comes from which parent. The same crossover mask is used for both offspring, however the roles are reversed ensuring that the bits used in the first offspring are not used in the second.

#### 5.4.1.2 Mutation

The mutate genetic operator produces small random changes to the genotype of the offspring from a single parent. In natural evolution this happens infrequently. There are two forms of mutation in natural evolution: *point mutation* and *inversion*; point mutation is an error corresponding to a single misprint. Inversion is where a piece of chromosome detaches itself and then reattaches itself in the inverted position. The most common mutation used in a GAs, is point mutation where small random changes are applied to the bit string by choosing a single bit at random then swapping its value. By doing this, diversity and innovation is added into the population, which introduces randomness into the normal GA.

The importance of recombination, of which crossover is a form of, has come under scrutiny, with arguments both supporting and dismissing the role of crossover in the simulation of evolution. Evolutionary Algorithms using no recombination at all, such as Evolutionary Programming, have been criticized as being insufficiently powerful (Goldberg, 1989; Holland, 1992). Alternatively others have concluded that the role of mutation in genetic algorithms has been underestimated (Fogarty, 1989; Back, 1993) and that the crossover role has been totally overestimated (Eshelman and Schaffer, 1993).

***Critical Observation.*** The aim of simulating an evolutionary process in this thesis is not to determine a clear winner, but to explore what behaviours occur when. In GAs information on intermediate generations is easily retained for this purpose. This is yet another reason why GAs are appropriate for the work in this thesis.

#### 5.4.1.3 Selection

In a GA, various fitness functions can be employed to rank the individuals in a population and select them for inclusion in the next generation. In *fitness proportionate selection* otherwise known as the *roulette wheel selection*, the chance that an individual will be selected is proportional to its fitness. In *tournament selection* two individuals are chosen randomly from a population and then compete for selection for the next generation. In *rank selection* the individuals are sorted by their fitness and then a specified number of the fittest individuals are selected.

### 5.4.2 Representation

Representation in GAs is the problem of deciding how the genetic material of the individual is constructed. In GAs this is referred to as the genotype. The genotype of an individual has two purposes. It is not only the genetic blueprint of what that individual will become, but it also provides the genetic material for the next generation. In the construction of a genetic code, the critical design question is how to represent the problem? What characteristics of the problem need to be included in the chromosome, i.e., the collection of genes? How can the differences between the individuals in a population can be represented? What are the building blocks that make-up the genotype? In most GAs the genotypes are binary strings so that the genetic operators of crossover and mutation can easily be applied. Any base can be used, however the lower the base the longer the string will be. Choosing how the individuals will be represented will depend on the nature of the problem.

### 5.4.3 The Evolutionary Process

The GA process is summarized in Figure 5.3, as taken from Holland (1992):

1. Evaluate each individual in the population to determine fitness defined as the performance of the individual
2. Rank individuals from high to low in order of fitness
3. Apply selection.
4. Higher ranking ones mate to produce offspring by crossover which replace low ranking individuals in the population
5. Mutate a small fraction of the population, i.e., flip a zero to a one or vice versa inversion mutation
6. Repeat steps 1 to 5 until a desired level of fitness is achieved or the maximum number of generations is reached

**Figure 5.3 The basic genetic algorithm**
A typical Genetic Algorithm as described by Holland (1992).

#### 5.4.3.1 Population size

In a GA it is typical to keep the population size fixed. In nature the population size may number millions, for example, the population size of insects is in the order of trillions. Population size is what is sustainable by the environment. In a GA this is also the case. The larger the population the more computational resources required, e.g., memory, processor. An increase in computer power, through faster processors and parallel processing, means that the software populations of a GA can in fact support the populations of nature. The computational requirements for a genetic algorithm search are dependent upon the number of generations and the population size. This is because the fitness of each individual in the population has to be calculated for each generation. Equation 5.1 gives the computational requirements for a GA, in terms of the computational power *C* (based upon Holland (1992)):

$$C = G * P \qquad (5.1)$$

where *G* is the maximum number of generations and *P* is the population size. Eq. 5.1 demonstrates that for the same computational resources, a larger population requires fewer generations. Large populations support diversity. In nature, the bigger the gene pool, then the more diverse the population. In a diverse population, it is harder for elitism to occur, i.e., the population converges to an individual that is not necessarily the optimal solution (Riolo, 1992). This is called the *premature convergence*. In a smaller population the fittest individuals dominate and the premature convergence problem can occur. In smaller populations the mutation and crossover have a greater impact, as a small change to an individual can have a drastic impact on the

population. Research results suggest that the larger the population size, the more diverse the generation will be (Holland, 1992). In a small population, it might be possible to avoid premature convergence by mutation, which becomes more important in terms of adding diversity and innovation, whereas in larger populations both crossover and mutation are important.

#### 5.4.3.2 When to stop – Convergence

After the genetic algorithm has run for several generations, it maybe that the individuals in the population may consist of similar if not identical genotypes. This is called *convergence*. It occurs when the selection operator of the GA has targeted a particular search area to the exclusion of other regions. The GA may converge to a genotype, which may not be the best solution. By maintaining diversity in the population this problem may be avoided.

## 5.5 Evolutionary Algorithms and Artificial Neural Networks

Evolutionary algorithms such as GAs combined with Artificial Neural Networks (ANNs) embrace the Baldwin effect (Baldwin, 1896):

1. If a species is evolving in a changing environment, there will be evolutionary pressure to favour the individuals with the capability to learn during their lifetime
2. Individuals who have the capacity to learn many traits rely less on their hard-wiring and use learning to overcome missing or partial traits

Before discussing ANNs in the context of GAs it is pertinent that the key characteristics of an ANN are highlighted (a detailed discussion of ANN was given in Section 4.7. The study of an ANN is motivated by the desire to simulate the biological learning process. The speed and ability of a biological

neural system to capture and process information has led researchers to assume that highly parallel processes operate on knowledge distributed over many neurons. ANNs have been developed to capture these characteristics. There have been many attempts to model the biological systems, (e.g., Gabriel and Moore, 1990; Churchland and Sejnowski, 1992; Zornetzer et al. 1994). Research on ANNs has not solely focused on the need to model biological systems; a second area of research has focused on using ANNs to obtain highly performing machine learning algorithms (Tesauro, 1989; Pomerleau, 1993)[4].

Evolution can be introduced into ANNs at three levels: the connection weights, the architecture and the learning rule. Evolution of the connection weights can overcome the local minima problem of backpropagation (refer to Chapter 4 Section 4.6 for an explanation of the local minimum problem) by using an EA to find a set of connection weights that minimize a predefined error function, which for example could be defined as the total mean square of the difference between the actual and target outputs. The fitness of the individual is determined by the error, i.e., the higher the error the lower the fitness. There has been much research in the evolution of the connection weights of an ANN (Yao, 1999). Evolution of an ANN's architecture provides an alternative to the traditional trial and error process and is also an area of considerable research (Yao, 1999). The evolution of an ANN's learning rule is still in its infancy, but is important not only for optimization purposes, but also in exploring the complex relationship between evolution and learning.

[4] In this thesis we touch on both areas of research- ANN as a model of biological system and using ANNs to obtain highly performing machine learning algorithms.

### 5.5.1 Evolution of the Weights in an ANN

There are two distinct processes in evolving the connection weights. The first is concerned with the way of representing the ANN in the genotype. If the genotype is to be represented as a binary string and all weights are to be included, then a decision has to be made as to how the weights are to be ordered within the genotype. Including all weights, which are real values represented as a binary string, will increase the length of the genotype. There has to be a trade-off between precision and length of the chromosome. Representing the ANN as a bit string lends itself to the *permutation problem* where different chromosomes actually represent the same ANN, which renders the genetic operator of crossover ineffective. Alternatively the weights can be represented as real numbers (Liu et al., 2004). This renders the genetic operators of binary crossover and mutation ineffective and special search operators have to be created (Montana and Davis, 1989), (in this case EP and ES may be used, as less emphasis is placed on crossover, and mutation becomes the primary operator). The evolutionary system, in this case, is invariably a hybrid, as it would be most effective (Moriarty and Mikkulainen, 1996; Lin et al., 1998). For example, a GA is used first for the global search, and then an ANN trained with backpropagation for the local search optimization. A typical evolutionary process for weight optimization is given in Figure 5.4.

Repeat

1. Decode each individual (genotype) in the current generation into a set of connection weights and construct corresponding ANN with the weights
2. Evaluate each ANN by training the network and computing its total mean square error between actual and target. The fitness of the individual is determined by the error. The higher the error, the lower the fitness. The optimal mapping from error to the fitness is problem dependent. A regularization term can be included in the fitness function to penalize large weights.
3. Select parents for reproduction based on their fitness
4. Apply search operators, such as crossover and/or mutation, to parents to generate offspring, which form the next generation

Until fitness > predefined value, where (error < predefined value)

**Figure 5.4 An evolutionary algorithm for the optimization of an Artificial Neural Networks through the evolution of its connection weights (adapted from Yao, 1999)**

***Critical observation.*** In this thesis the connection weights will not be evolved. Evolution will focus on an *indirect* representation, i.e., the number of hidden nodes and learning rules, which is biologically more plausible as it is impossible to code up the entire nervous system's chromosomes (see Section 5.2.2 for further explanation).

### 5.5.2 Evolution of the ANN's Architecture

The architecture of an ANN is defined by the connectivity and transfer function of each node. According to Yao (1999), EAs are useful in optimizing ANN design. EAs provide an alternative to the traditional trial-and-error design of cutting through the large search space of possible nodes and connections. In addition, similar architectures yield different performances. Again there are two processes: representation and evolution. Representation of the architecture of an ANN in a genotype can either be direct or indirect. *Direct representation* separates the architecture from the weights. For example, an ANN with *N* nodes can be represented by an *N-by-N* matrix

where a 1 represents a connection and a zero no connection. The problems associated with this form of representation are: (i) scalability, i.e., a large ANN means a large matrix, and (ii) the permutation problem, where different chromosomes yielding the same ANN still exist. The weights and architecture are then evolved together (Pujol and Poli, 1998). *Indirect representation* is parameter-driven, where the number of hidden layers and the number of connections between the layers are included in the genotype (Harp et al., 1989).

### 5.5.3 Evolution of the ANN's Learning rules.

Evolution of the ANN learning rules can be viewed as learning to learn. Knowledge that is acquired during an individual's lifetime is not passed onto the offspring (Dawkins, 1989); the learning ability however, of an individual might be passed on. There are many variations on how this can be done. The *Pitt approach* encodes the whole learning rule as a population (Kitano, 1990), whereas the *Michigan approach* encodes each learning rule as a genotype (Yao and Shi, 1995). In this case the representation needs a set of terms and coefficients in the genotype.

***What is new and overview?*** Research in the evolution of learning rules for an ANN is in its infancy. In Chapter 7 in this thesis, the parameters that define the learning rule, i.e., the learning rate and discount rate, will be encoded into the genotype as in the Michigan approach. The results that will be presented in Chapter 7, of combining the techniques of GAs, ANNs and RL are a contribution as they further the understanding of how GAs and ANNs can be

combined not only for optimization purposes, but also to explore the complex relationship between evolution and learning.

### 5.5.4 Summary of Evolutionary Artificial Neural Networks

Research results support that an ANN's learning ability can be improved through evolution (Yao, 1999). For effectiveness both the architecture and the weights need to evolve. Separate evolution of architecture and weights can make fitness evaluation inaccurate and misleading. Evolving just the architecture can be misleading as the same architecture with different sets of initial weights can give different results. Hence the evolution of architectures without any weight information has difficulties in evaluating fitness accurately and as a result evolution could be inefficient. If just the weights are evolved, then problems may occur with crossover as knowledge distributed amongst the connections can be destroyed. Another problem that can occur when a GA is used as an optimizer for an ANN is the *permutation problem,* where different genotypes represent the same Artificial Neural Network. When this happens the crossover operator is ineffective as it destroys the knowledge distributed amongst the connections. The best approach depends upon what one is trying to achieve by evolving the ANN. If the aim is to find a particular type of ANN, then evolving the architecture using indirect representation (the genotype includes the number of hidden layers and the number of connections between the layers), rather than being too exact, would be the best solution (Yao, 1999). Bullinaria's (2003) solution of incorporating initial weight connections with the learning parameters seems to be a good compromise.

***What is new and overview?*** In this thesis the evolution of the 2-ANNs model is carried out using the indirect representation combined with the Michigan approach described above. Specifically the architecture is represented as the number of hidden nodes and the learning rule is represented as parameters, both encoded in the genotype of an individual. Evolving the architecture and learning together is a novel approach.

## 5.6 Combining the techniques of Evolutionary Algorithms, Artificial Neural Networks and Reinforcement Learning

Research combining the techniques of EAs, ANNs and RL, still seems to be in its infancy even though applications combining EAs with reinforcement learning (RL) commenced with the work of Holland's classifier systems (1986) in the mid-eighties. A survey of EAs for RL (EARL) carried out by Moriarty et al. (1999) suggests that combining aspects of these different approaches maximizes the advantages of all three methods. EAs such as GAs are search methods, which have the ability to handle incomplete information on the state of environment as well as being able to cope with dynamic environments. The essential component of RL is prediction or estimation of the value of future rewards, which involves learning from interaction with the environment. The advantages of ANNs are in their speed and ability to capture and process information, and their ability to generalize. Combining these methods can only produce better hybrid systems.

Implementing a hybrid system such as an EARL is not without its difficulties. In addition to the design decisions for the EAs, (how to represent the individuals and what operators to use), when an EA is combined with RL one

is faced with the decisions of how to represent the policies and how to evaluate fitness. A policy can be represented as a single genotype, or distributed over several genotypes, as in Holland's classifier system (1986). A policy can also be distributed over a population represented by an ANN as in the Symbiotic Adaptive Neuro-Evolution (SANE) system of Moriarty and Mikkulainen (1996). The fitness of a policy can be either the total payoff for an individual, or an average of the payoff of an individual over a certain number of trials. In RL there is the additional problem of how to assign blame to past actions, i.e., the *temporal credit assignment problem*. It may be the case, that special genetic operators for RL need to be defined, such as the Triggered operator in Holland's classifier system. Moriairty et al. (1999) lists the strengths of EARL as being able to cope with large, incomplete state spaces, and dynamic environments. However, they then suggest the weakness of an EARL system is that it requires a large number of experiences to learn, a problem also true for TD, and the states that are visited infrequently are overlooked, which is not a problem for TD as TD keeps a record of the states visited.

Although EA methods on their own have been criticized as not being suited to RL problems (Sutton and Barto, 1998), evolution and learning work naturally together. GAs are implemented in a variety of RL architectures; for example, Lin et al. (1998) implement a GA combined with RL in the actor-critic architecture of Barto et al. (1983), with the reinforcement signal used as the fitness level. Their results indicate faster training times. The SANE system of Moriarty and Mikkulainen (1996) and Richards et al. (1998), implements a

population of neurons. The fitness function of the neuron is defined by the level of fitness of the ANN to which it belongs, i.e., how well the network performs on a given task. In SANE the temporal credit assignment problem is dealt with by evolving the fittest neurons, which is interpreted as rewarding the best. Moriarty and Mikkulainen (1996) claim that this method overcomes the local minima problem and outperforms the traditional RL techniques such as the actor-critic and Q-learning.

## 5.7 Concluding Remarks

Evolutionary Algorithms have three main techniques GA, EP, and ES. Out of this, GAs are a more exact simulation of natural evolution. GAs are typically used as an optimization technique for ANNs. In this thesis, we combine the techniques of GAs, RL and ANNs in a novel approach. Specifically the role of evolution and learning in the development of self-control through precommitment behaviour will be investigated in later chapters. Combining the techniques of GAs, RL and ANNs is not without its difficulties, as described in this chapter. However, the combination of evolution and learning lends itself to better hybrid systems, which overcome the problems of: (i) the local minima as opposed to global minima found in using ANNs alone, (ii) the exploration *versus* exploitation dilemma of RL and dynamic environments, problems typical of the techniques used in isolation. The next chapter combines RL and ANNs in the context of MARL to develop and test the 2-ANNs model presented in Chapter 3.

# Chapter 6

# 6 Explaining Self-Control by Playing Games.

## 6.1 Chapter Outline

In this chapter the neural model of self-control in Figure 3.3 is implemented as two players competing in a game-theoretical situation. More specifically, the higher and lower centres of the brain are implemented as two simple feed forward multi-layer neural networks using reinforcement learning. The ANN representing the higher brain centre is implemented with the Temporal Difference weight update rule (Sutton, 1988) and is explained in detail in Section 6.4.2.1. In summary, the Temporal Difference rule is implemented in this thesis with a lookup table, which maintains a history of previous rewards and includes a discount rate used in determining the value of future rewards. For these reasons, in this thesis, the Temporal Difference rule is viewed as being far-sighted and thus associated with the higher brain processes. The ANN representing the lower brain centre is implemented with the Selective Bootstrap weight update rule (Widrow et al., 1973) and is explained in detail in Section 6.4.1.1. In summary, the Selective Bootstrap weight update rule has no memory of past rewards and no mechanism for estimating future rewards, hence, can be viewed as myopic.

In the development and testing of the neural model (2-ANNs) presented in Chapter 3, the two ANNs compete in two games, Rubinstein's Bargaining Game (RBG) (Rubinstein, 1982) and the Iterated Prisoner's Dilemma (IPD) game (Axelrod and Hamilton, 1981). The RBG and the IPD are *general-sum* games that model real-world situations. General-sum games were introduced

in Chapter 4. In summary, *general-sum* games are where the players' payoffs are neither totally positively nor totally negatively correlated (Sandholm and Crites, 1996). Learning can be considerably more difficult in such games, which require both a mixture of cooperation and competition (Kaebling et al., 1996). The ANNs in the 2-ANNs model in this thesis can be viewed as autonomous learners with interacting or competing goals. The ANNs in the 2-ANNs model learn separately, but simultaneously. This makes the game more difficult because (i) the simultaneous learning of the other player creates a dynamic environment, and (ii) the other learner also has no prior knowledge of the game. Research in multi-learners or multi-agent learning is still in its infancy. A review of multi-agent reinforcement learning in a shared environment, within the context of the current research in general-sum games is given in the next section.

In this chapter two sets of experiments are conducted to explain self-control through games. The first set of experiments uses Rubinstein's Bargaining game (Rubinstein, 1982). The Rubinstein's Bargaining game (RBG) exhibits key characteristics of the self-control problem. The players have the dilemma of either accepting an unreasonable offer now or holding out for an acceptable offer later. In the RBG the resource diminishes with time and hence, the players have to cooperate with each other by each taking into account the other player's impatience and making an acceptable offer without delay. It has been shown that the outcome of the RBG is dependent upon the other player's discounting of future rewards (Kreps, 1990). In the first experiment using the RBG, an artificial opponent, whose responses are generated randomly,

competes against, first the Selective Bootstrap network and then the Temporal Difference network. In the next experiment using the RBG, the two ANNs, representing the higher and the lower centres of the brain, compete against each other. The results are compared to the economic literature on self-control, specifically Rubinstein (1982) and Kreps (1990).

In the second set of experiments the Iterated Prisoner's Dilemma game is used (Axelrod and Hamilton, 1981). As we have seen in Chapter 2, research on self-control suggests there is a relationship between cooperation and self-control (Brown and Rachlin, 1999). Human cooperation has been modeled as a game of Prisoner's Dilemma (Axelrod and Hamilton, 1981). Brown and Rachlin (1999) played a variation of the Iterated Prisoner's Dilemma game (IPD) to represent the dilemma of the self-control problem, whereby choosing a higher immediate reward conflicted with behaviour that maximized the overall reward in the long term. For this reason, at this stage of this thesis the IPD is an appropriate game to use to verify the 2-ANNs model. The results of the 2-ANNs model are compared with the empirical results of Brown and Rachlin (1999), which showed a close analogy between self-control and social cooperation. To summarize, Brown and Rachlin (1999) concluded that the path to greater self-control is in our confidence that we will continue to cooperate with our selves in the future (refer to Section 2.3 for further details).

Precommitment is a mechanism for greater self-control by carrying out an action now with the aim of denying (or at least restricting) our future choices. In this chapter, it is proposed that this can be interpreted as biasing our

choices towards future rewards. The final set of experiments implements this bias towards future rewards in three ways: (i) as a *variable bias* with different values of the inputs of the network's bias existing node, (ii) as an extra input to one or both of the ANNs in the 2-ANNs model and (iii) as a *differential bias* applied to the payoff matrix. The results of this final set of experiments are compared to the empirical results of Baker (2001) summarized in Chapter 2, Section 2.3, Figure 2.6, which showed that increasing the probability of reciprocation promoted cooperation. The premise here is that precommitment, implemented as a bias towards future rewards, behaves in the same way, i.e., increasing precommitment promotes cooperation.

## 6.2 Multi-agent Reinforcement Learning and General-sum Games

In the last decade reinforcement learning (RL) as applied to games has been an active area of research (Kaebling et al., 1996; Sutton and Barto, 1998). Much of the research in RL and games has to date focused on single learners in strictly competitive games with clear winners. In recent years there has been much work on extending RL to the multi-agent domain (Hu and Wellman, 1998; Bowling and Veloso, 2001; Littman 2001). Littman (2001) defines multi-agent learning as the case where multiple adaptive agents, with interacting or competing goals, are learning simultaneously, in a shared environment. Since the other agents are also learning and adapting, this makes the environment dynamic. RL is well suited for multi-agent learning. In Chapter 4 we saw that RL does not need a complete specification of its environment, it can deal with a dynamic environment, and RL is adaptable.

In Chapter 4, Reinforcement Learning (RL) was discussed in the context of a single agent within the mathematical framework of a Markov Decision Process (MDP). RL within the MDP framework has an *Environment*, which contains a critic to evaluate the learner's (agent's) actions and everything external to the agent. The *State* is a summary of past behaviour that is needed to determine future behaviour, which is referred to as the *Markov property* (Sutton and Barto, 1998). There is a single *Agent*, which is the learner, e.g., the player or ANN. An *Action* is what the agent can do, e.g., board move, movement around the room, selecting a lever. There is a *Reinforcement Signal* to evaluate the current action. The MDP framework is a model for Single Agent Reinforcement Learning (SARL). SARL has been the focus of much active research in the context of zero-sum games with great success. Tesauro's TD-Gammon (Tesauro, 1994), a backgammon program using RL and ANN achieved expert level performance (Tesauro, 2002). TD-Gammon is a single learner playing itself in a zero-sum game, whose action at any point in time may be uncertain, but the state of the board and opponents are completely observable.

It is considerably more difficult to apply RL in general-sum games (Kaebling et al., 1996). General-sum games have multiple learners with interacting or competing goals in a shared environment. Earlier attempts in implementing mulit-agent reinforcement learning systems (MARL) used the SARL model with other agents treated as part of the environment; hence the environment became dynamic (Tan, 1993; Balch, 1997; De Jong, 1997). Littman (1994) was the first to introduce *Markov games* as a model for MARL. *N*-player

games fit into the mathematical framework of a *Markov game*. There is a set of actions, a transition function and a set of rewards for each player (van der Wal, 1981). This definition also holds true for *stochastic games* (Shapley, 1953) and sometimes the two terms are used interchangeably. *N*-player games prove an interesting challenge, as the effects of a player's action is dependent on the other players, who themselves are learning and adapting. This challenges the traditional notion of converging to a single optimum equilibrium as in a single learner zero-sum game. Littman (1994) tested his theory of Markov games as a model for MARL with the algorithm *minimax-Q*. Minimax-Q extends Q-learning for SARL by replacing the maximum Value Function with a function to calculate each player's best response. Littman (1994) found that minimax-Q worked on a restricted set of games, namely 2-player zero-sum games. The use of linear programming to calculate the minimax value made it computationally expensive, and although minimax-Q did guarantee convergence it did not necessarily converge to the best response. Sandholm and Crites (1996) examined the extent Q-learning could be applied in the Iterated Prisoner's Dilemma game, which is a general-sum game. Each learner, implemented as a Q-learner, competes against a fixed strategy of Tit-for-Tat. Their results showed that optimal strategies could by achieved; however convergence was not guaranteed. Hu and Wellman (1998) introduced, what is referred to by Bowling and Veloso (2000), as *Nash-Q*. Hu and Wellman (1998) showed that convergence to a single Nash equilibrium (Nash, 1950a) using Q-learning was possible for restrictive set of general-sum games implemented in a 2-player Markov game framework. Littman (2001) identifies two limitations of Nash-Q in the context

of general-sum *N*-player games: (i) it converges to a single equilibrium and (ii) it only converges after infinite trials. Claus and Boutlier (1998) developed *Joint Action Learners* (JAL) based on SARL using the TD method of Q-Learning. The agents in JAL learn actions based on the estimated actions of the other players. JAL converges for a restrictive set of general-sum games. Bowling and Veloso (2001) concurred that it was necessary that any MARL algorithm satisfied two properties: *rationality* and *convergence*. They concluded that all the MARL algorithms, proposed to date, failed to satisfy these two properties, either converging to non-optimal solutions or not converging at all. They proposed an alternative algorithm using Policy Hill Climbing (an extension of Q-learning to play mixed strategies), based on the Win or Learn Fast principle (*WoLF*). The basis of WoLF was to learn quickly while losing, and slowly while winning. It did this by varying the learning rate. When the player is losing, a larger learning rate is used, when the player is winning the learning rate is reduced. WoLF was applied to the simple grid-world general-sum game with some success. The results showed that the WoLF algorithm satisfied the properties of rationality and convergence.

More recently, this early research on MARL, e.g., Nash-Q, Minimax-Q, has been criticized as focusing on a unique equilibrium, which is too limiting (Bowling and Veloso, 2001; Shoham et al., 2003). In general-sum games there may be many Nash equilibria (Nash, 1950a). It is unrealistic to assume that all learners converge to a unique strategy; in fact it is incorrect to assume that all learners in a shared environment are playing with the same strategy. Littman (2001) presented an alternative to the Nash-Q, called *Friend-or-Foe Q-*

*learning* (FoF-Q) where other players are categorized as either a friend or a foe, which attempted to address the multiple strategy problem of MARL. FoF-Q was found to converge for a restrictive set of general-sum games, either cooperative or zero-sum games. Greenwald and Hall (2003) have developed *Correlated-Q* (CE-Q), which addresses the multiple equilibria problem of a general-sum game by categorizing the agents as either: (i) utilitarian, (ii) egalitarian, (iii) republican, or (iv) libertarian. CE-Q has been tested successfully on a restrictive set of deterministic general-sum games, the grid-world game and soccer. Shoham et al. (2003) have even questioned the approach and scope of earlier work on MARL and general-sum games. They argue that there still does not exist a formal model of learning for MARL, which addresses how the agents learn in the context of other learners. In addition, Shoham et al. (2003) argue that currently there does not exist a method to find what the best learning strategy is for each agent in order that the game successfully converges. They even criticise the use of the Markov mathematical framework as a model for general-sum games. Littman (2001) agrees that a complete treatment of general-sum games is still lacking.

All of the above methods have a centralized process that is multi-functional, it maintains Q-values, determines agents' actions, and approximates action-value functions. As an attempt to address the limitations of earlier research, recent research has moved away from this centralized processing placing control for an agent's behaviour with the agent. In this case the agent is responsible for determining his or her own best response. The only prerequisite is that the agent is aware of the other agents' actions. In addition,

other forms of learning are being investigated for their relevance to MARL. For example, *No-regret* learning is another form of learning other than RL, where the average payoff of each player exceeds that of any payoff achieved by any fixed strategy (Jafari et al., 2001). No-regret learning algorithms have been implemented, in Multi-Agents Systems (MAS) on a restrictive set of general-sum games with some success. Bowling (2005) combined no-regret with WoLF (GIGA-WoLF) to tackle the problems of convergence and no-regret. Gondek et al. (2001) has extended no-regret to *QnR-Learning* in a distributed multi-agent system with each agent playing according to their own policy generated independently. This work is still in its infancy. Early results are promising, comparable to CE-Q yet computationally less expensive than earlier MARL algorithms.

***Critical Observation.*** MARL is an active area of research. MARL as applied to general-sum games is still in its infancy with earlier research under scrutiny. New learning algorithms and alternative approaches are being explored. Recent work, that allows agents in a MAS to be autonomous (Gondek et al., 2001; Gao, 2004), removes the limitation of centralized learning and widens the scope for the agent's behaviour. This could be seen as a move back to a variation of SARL. There still lacks a clearly defined statement on when RL can be applied to general-sum games usefully and in what form. The exploration and exploitation trade-off, a distinguishing feature of RL, has not been dealt with adequately within the current MARL systems. As yet, an algorithm that can be applied to the complete set of general-sum

games does not exist. Although clearly much progress has been made, there is still a great deal of scope for future research.

## 6.3 What is new and overview?

The 2-ANNs model presented in this thesis has 2 autonomous players simultaneously learning in a shared environment playing a general-sum game. This makes our model framework a multi-agent system (MAS). From a review of the literature on MARL presented above, it is considerably more difficult to apply RL in such games (Kaebling et al., 1996). A feasibility study was conducted to establish the extent to which RL can be applied in a game with real world consequences (Banfield and Christodoulou, 2003). The results showed that reinforcement learning could be applied successfully to 2-player general-sum games that model real-world situations. More specifically convergence was reached where the opponent is artificial whose responses are generated randomly with uniform probability and also in the more complex scenario where the opponent is another learner with interacting or competing goals. The work in this thesis builds upon the initial results of the Banfield and Christodoulou (2003) study and breaks from the traditional framework for MARL by removing the limitation of centralized learning. It does this by implementing the two players, representing the higher and lower brain centres, as autonomous agents, which learn simultaneously in a shared environment.

## 6.4 Explaining Self-Control with The Rubinstein's Bargaining Game

The Rubinstein's Bargaining game (Rubinstein, 1982) is a general-sum game, incorporating aspects of the self-control problem. In addition, the Rubinstein's Bargaining Game (RBG) is considered to be to be particularly simple. For these reasons it seemed an appropriate game to use to develop and test the 2-ANNs model presented in Chapter 3. The RBG involves two players and a resource or pot, e.g., money. The two players seek to agree how to divide the pot. The pot decreases at each turn of the game by a fixed amount, hence it pays both players to reach an agreement sooner rather than later. Rubinstein (1982) added the concept of discounting to the bargaining game by diminishing the size of the pot with time (Nash, 1950b), in order to give the bargaining game its dynamic nature. At the beginning of a turn, one player makes an offer. An *offer* from one player to another is the fraction of the pot the player is willing to give to the other player. The other player can either accept or decline. If he rejects the offer, he then makes a counteroffer and the game continues for another turn. The game terminates when either nothing remains in the pot or one of the players accepts an offer. Each player seeks to gain as much of the pot as she or he can. The RBG has two further assumptions: (i) each of the two players has different discounting functions representative of the different degrees of impatience and (ii) the process of offers and counteroffers delays the players receiving their share of the pot. The first player will have the advantage in that he or she can determine the payoff for the second player. The second player may wish to avoid the delay and accept any offer depending upon his or her discount function. The

expected result is that the player least affected by the delay, i.e., the less impatient, will receive the larger output (Kreps, 1990).

### 6.4.1 Selective Bootstrap feed forward network (SB-FFWD) playing an Artificial Opponent in the RBG

#### 6.4.1.1 Introduction

In this experiment an ANN competes against an artificial opponent whose strategy is random. The motivation for this first experiment is to firstly to test the hypothesis that the lower brain functions can be modelled with the Selective Bootstrap weight update rule (Widrow et al., 1973) and secondly, to determine the optimal configuration for the ANN. In addition, there is also a requirement to verify the techniques to be used since, as discussed in Section 6.2, applying RL in games where there is no clear winner is considerably more difficult. The Selective Bootstrap rule, which is a trial and error technique for reinforcement learning, has no mechanism for discounting future rewards; it simply learns the value of each action. The discount rate in this case can be assumed to be zero. For this reason it can be considered to be myopic. The Selective Bootstrap weight update rule, as used in reinforcement learning, is a variation on the Widrow-Hoff rule (Widrow and Hoff, 1960) used in supervised learning. In the Selective Bootstrap rule the target output is unknown; instead if the actual output produces a success then the actual output plays the role of the desired output and the weights are updated as if the actual output produced was in fact the desired output. On the other hand if actual output leads to a failure then the desired effect is to negate the actual output and behave as though the actual output was never produced. The rule

works as follows, in the case of a success, the actual output is used to reward the network by updating the synaptic weights as shown in Eq. (6.1):

$$\Delta w_t = \eta \left[ a_t - s_t \right] x_t \qquad (6.1)$$

where time ($t$) is the of number of completed rounds, $s_t$ is the sum of the weighted inputs to the postsynaptic neuron, $w_t$ is the synapse weight at time $t$, $\eta$ is the learning rate and $x_t$ is the input to the postsynaptic neuron at time $t$, $a_t$ is the actual output of the neuron at time $t$ and is used as the target output. To penalise the network, if the action is deemed a failure, the weights are updated as shown by Eq. (6.2):

$$\Delta w_t = \eta \left[ 1 - a_t - s_t \right] x_t \qquad (6.2)$$

where $a_t$ is the actual output at time $t$, $s_t$ is the sum of the weighted inputs to the postsynaptic neuron, $w_t$ is the synapse weight at time $t$, $\eta$ is the learning rate and $x_t$ is the input to the postsynaptic neuron at time $t$. In summary, for the Selective Bootstrap rule, if the output from the neuron corresponds to a success, then the weights are updated as though the actual output is the desired output. However, if the output corresponds to a failure, the weights are updated as if the actual output was never produced.

#### 6.4.1.2 Methodology

The ANN plays an artificial opponent whose accept or decline response is generated randomly. If the artificial opponent rejects the network's offer it then generates a counteroffer randomly. The two opponents (the ANN and the artificial opponent) go through a number of turns for each game, each taking it

in turns to make an offer, which is either accepted or rejected, and a counteroffer is made.

The reinforcement learning algorithm is implemented as a multi-layer feed forward Artificial Neural Network with non-linear nodes, i.e., the output of the node is calculated using the Sigmoid threshold function of Eq. 4.8. In playing the Rubinstein's Bargaining game the concept of time is only relevant for the duration of the game, so there is no need to retain details of previous games. There is therefore no need to employ temporal neural networks, for example, Time Delay or recurrent neural networks. For optimization purposes the ANN implementation is flexible allowing for the learning parameters as well as the topology to be easily changed by the experimenter during training.

The design of the system in this experiment follows the single agent reinforcement learning of the Markov Decision Process (MDP) mathematical framework depicted in Figure 4.1. The system configuration for a RBG of complete information is shown in Figure 6.1. The environment contains: (i) a *process* that initializes the pot size and the artificial opponent's offer at the start of the game and at each round it reduces the size of the pot and generates the artificial opponent's counteroffer randomly, and (ii) a *critic* who rewards or penalizes the ANN at each turn of the game and at the end of the game. The artificial opponent is part of the environment. The *state* in the MDP is the input to the Artificial Neural Network (ANN). In the case of the RBG, if the game is a game of *complete* information, where both players know the size of the pot, the state is both the offer and the pot size. A game of *incomplete*

information is where the size of the pot is unknown and is modelled as having just one input to represent the opponent's offer. The *action* in the MDP is the output from the ANN. In the case of the RBG this is the accept/reject response and the counteroffer. The *agent* from the MDP is modelled as an ANN.

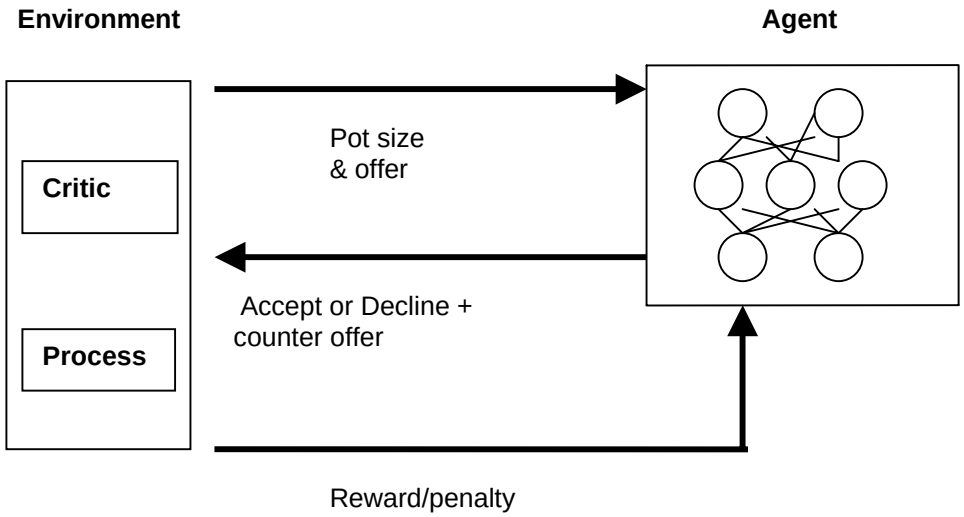



**Figure 6.1 System Configuration for an ANN playing an Artificial Opponent in a Rubinstein's Bargaining Game of *complete* information**
The environment has a *process* that initializes the pot size and the artificial opponent's offer at each round, and a *critic* who rewards or penalizes the network at each turn of the game and at the end of the game. The artificial opponent is part of the environment.

The network is rewarded (i.e., $r_{t+1}$ is set to one), if:

1. the artificial opponent's offer is lower than the ANN's previous offer and the ANN rejects it or
2. the artificial opponent's offer is higher than the ANN's previous offer and the ANN either:
    a. accepts the offer or
    b. makes a counteroffer higher than the artificial opponent's offer

For all other actions the ANN is penalised (i.e., $r_{t+1}$ is set to zero). The ANN receives a reinforcement signal at each turn of the game. At the end of the game, the ANN is rewarded if it won the greater share of the pot, otherwise it is penalised. The number of turns and the number of games played are varied. At each turn of the game, the artificial opponent's offer, the ANN's offer and the network's accept or decline response are all recorded. At the end of the game the average winnings, total winnings, the total number of mistakes and the average number of mistakes are also recorded.

Traditionally the performance criterion used to measure the success of learning in an ANN is an error function computed as the mean square of the difference between some desired output and the actual output of the ANN. In the bargaining game, and in RL in general, the desired output is unknown. In the bargaining game the players may or may not know the size of the pot. For example, in the case of a strike negotiation the strikers may have some desired figure, which may be based on their current salaries as opposed to a percentage of the company's profits, which are represented by the pot in this case. For this reason it was decided to measure how successful the ANN learnt by measuring the number of *mistakes* the ANN made. In addition, a mistake is defined with the aim of helping the ANN to maximize its share of the pot. Hence it is expected that reducing the number of mistakes made, results in an increase in the share of the pot. A *mistake* is defined to be the case where:

- The opponent exceeds his last offer, but the ANN declines

or

- The opponent's offer is less than the ANN's last offer and the ANN accepts

***Success Criteria.*** How successful the ANN learns is evaluated by the following criteria:

1. The number of mistakes the ANN made
2. The share of the pot the ANN won
3. The length of the game, i.e., the number of turns

#### 6.4.1.3 Test Procedure

The ANN has as input the opponent's offer. An offer is represented as a real value between zero and one. If the game is a game of complete information then the ANN has the size of the pot as an extra input. The pot is a fraction of the previous pot and is represented as a value between zero and one. There are two outputs, represented by real values between zero and one; one output accepts or declines the opponent's offer and the other output gives the ANN's counteroffer. An accept response is represented as any value greater than or equal to 0.5. A decline is represented as any value less than 0.5. There is no pre-processing carried out on the input or output values as these are already real numbers between zero and one.

The network bias is implemented as a node with an input value of 1 (typically the bias is a value between -1 and 1 or -0.5 and 0.5) with random weights. The bias is evaluated as a weight and added to the sum of the weighted inputs to

the node in the next neural network layer. The bias is in effect the weight, given that the input to the extra node is 1, as shown in Figure 6.2.

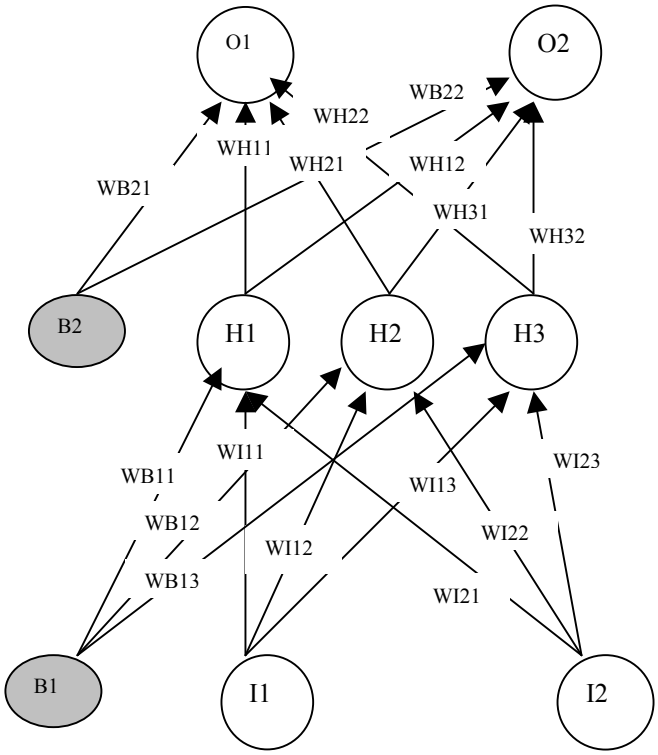



**Figure 6.2 Implementation of the bias mechanism in an Artificial Neural Network**
The bias is implemented as a node with an input value of 1, which is fixed. The bias nodes are shown as grey. The weights of the bias nodes are modified during learning in the same way as all the other network's weights.

For example, with reference to Figure 6.2, consider the *H1* neuron in the hidden layer. The neuron *H1* is connected to two neurons in the input layer *I1* and *I2*. The neural connection between the neuron *H1* in the hidden layer and the neuron *I1* in the input layer is *WI11*, likewise, the connection between the neuron *H1* in the hidden layer and the neuron *I2* in the input layer is *WI21*.

The net input (sum of weighted inputs) to the hidden layer neuron *H1* is calculated as in Eq. 6.3:

$$net_{H1} = (WI11 * I1) + (WI21*I2) + Bias \qquad (6.3)$$

The bias is effectively the weight *WB1i* (where *i* equals 1, 2 or 3) of the neuron *B1*, which has an input value of 1. Given this the $net_{H1}$ is calculated as in Eq 6.4:

$$net_{H1} = (WI11 * I1output) + (WI21*I2output) + (WB11*1.0) \qquad (6.4)$$

and similarly for the hidden layer neurons *H2* and *H3*. Now consider the *O1* neuron in the output layer. *O1* is connected to three neurons in the hidden layer *H1*, *H2* and *H3*. The neural connection between neuron *O1* in the output layer and neuron *H1* in the hidden layer is *WH11*, likewise, the connection between neuron *O1* in the output layer and neuron *H2* in the hidden layer is *WH21* and finally the connection between neuron *O*1 in the output layer and neuron *H3* in the hidden layer is *WH31*.

The net input (sum of weighted inputs) to the output layer neuron *O1* is calculated as in Eq. 6.5:

$$net_{O1} = (WH11 * H1) + (WH21*H2) + (WH31*H2) + (WB21*1.0) \qquad (6.5)$$

and similarly for the output layer neuron *O2*.

The task of the hidden nodes for this experiment is similar to a classification problem. In this experiment, the task is to form a decision boundary as to what is an acceptable offer.

When an ANN is initialized, all the weights are assigned a random value between - 1 and 1, which means that the bias weights *WB1i* and *WB2i* are also initially assigned a small random value between -1 and 1. Starting on randomly selected initial weights means that each time the ANN plays, different results can be expected, since learning starts at different points (of error *versus* weights hypersurface). To overcome this disparity the results are compared for multiple trials and the best are selected. The ANN's response initially is generated at random, with training it is expected that the ANN's response will be that of the best response, which according to game theory in the RBG should be close to half of the pot. The artificial opponent's responses are generated randomly. The artificial opponent goes first, which gives it the first player advantage. Since the main aim of this experiment is to determine the optimal configuration for the ANN this is not considered to be a problem. The game was played with the maximum number of turns per game held at 10.

The elements of the ANN that were varied during testing to determine the optimal configuration included the topology of the network, (i.e., the number of hidden layers and the number of neurons in each layer) and the learning variables (i.e., the learning rate for the Selective Bootstrap network). How successfully the ANN learns is measured by the success criteria as listed in Section 6.4.1.2.

#### 6.4.1.4 Results

The objective of this first set of tests was to find the optimum configuration for the Selective Bootstrap ANN (Boot) for both a game of incomplete information, i.e., where the players do not know the size of the pot and a game of complete information, i.e., where the players know the size of the pot. Figure 6.3 shows the effect of varying the learning rate for a game of *incomplete* information. More specifically, Figure 6.3 shows the best learning curve from at least three tests for each learning rate tested.

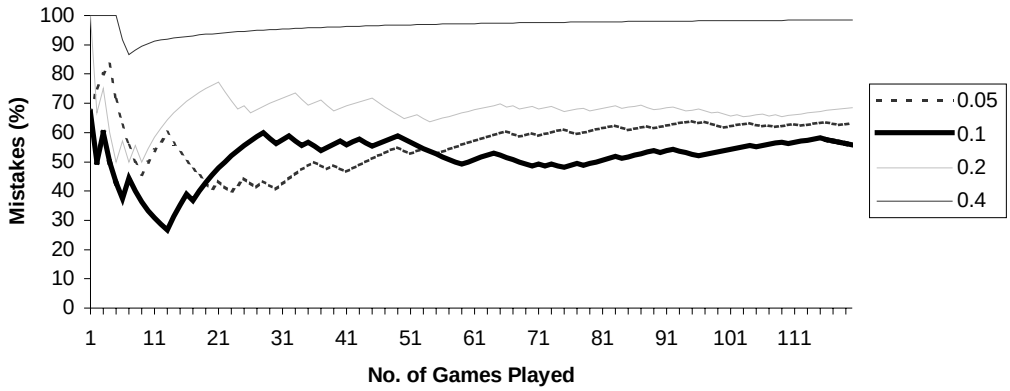



**Figure 6.3 The effect of varying the learning rate for a RBG of *incomplete* information**
The graph is the best learning curve from at least three tests for each learning rate tested for a game of incomplete information, i.e., neither player knows the size of the pot. The learning rate of 0.1 was selected as optimum as it produces the lowest probability that the ANN will make a mistake on the next game.

In each case the tests were carried out on an ANN of 1 input node (to represent the artificial opponent's offer), a hidden layer of 3 nodes and an output layer of two nodes to represent the accept/decline response and counteroffer. The graphs show the probability of making a mistake based on the number of mistakes made over the total number of turns in 121 games. A trial was actually run for 1000 games, but it was found after approximately 121 games the changes to the number of mistakes and weights to be

insignificant. All the learning rates tested produced learning curves that decayed rapidly and somewhat unsteadily, but then settled down to a plateau. The reason for the variations in the number of mistakes made in the earlier games may be explained in that learning starts at different points due to the randomly selected initial weights. The learning rate of 0.1 was selected as optimum as this produced the lowest probability that the ANN would make a mistake on the next game.

The test was repeated for a game of complete information, i.e. the players know the size of the pot. Figure 6.4 shows the effect of varying the learning rate for a game of *complete* information.

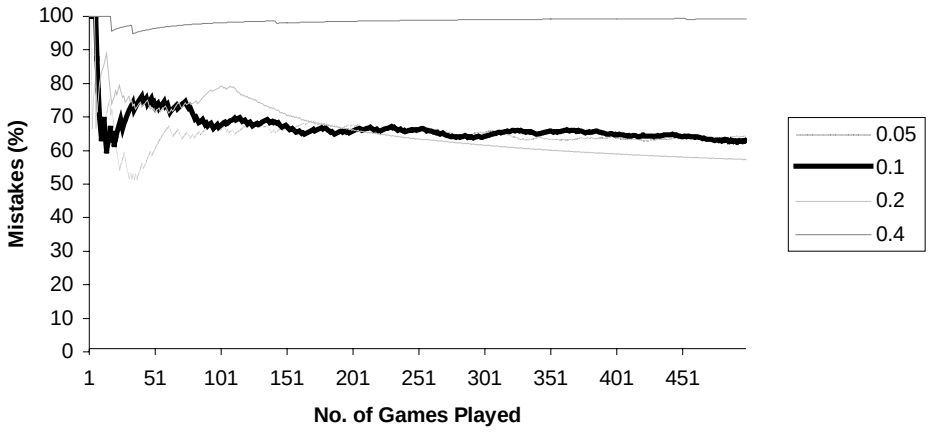



**Figure 6.4 The effect of varying the learning rate for a RBG of *complete* information**
The graph is the best learning curve from at least three tests for each learning rate tested for a game of complete information, i.e., each player knows the size of the pot. The learning rate of 0.1 was selected as optimum as it is less volatile than 0.2 and 0.05 and plateaus to a low probability that the ANN will make a mistake on the next game than 0.05 and 0.4.

Again, the graph shows the best learning curve from at least three tests for each learning rate tested. In each case the tests were carried out on an ANN of 2 input nodes (one to represent the artificial opponent's offer and another to

represent the pot), a hidden layer of 6 nodes and an output layer of two nodes to represent the accept/decline response and counteroffer. A trial was actually run for 1000 games, but it was found after approximately 500 games the changes to the number of mistakes and weights to be insignificant. As in the previous experiment, all the learning rates tested produced learning curves that decayed rapidly and somewhat unsteadily, but then settled down to a plateau. Again, the learning rate of 0.1 was selected as optimum, as this produced the curve with the least volatility and a low level of probability that the ANN would make a mistake on the next game.

Figure 6.5 shows the effect of varying the depth, i.e., number of hidden layers and the number of hidden nodes in each layer, for a game of *incomplete* information. Varying the depth and the number of hidden nodes produced some interesting results. With fewer hidden nodes and no hidden layer the behaviour from the network tended to be volatile. With more hidden nodes the time to learn increased shown by a somewhat unsteady decline settling to a plateau. From these results, for a game of *incomplete* information the configuration of 1-3-2 with a learning rate of 0.1 was considered optimal.

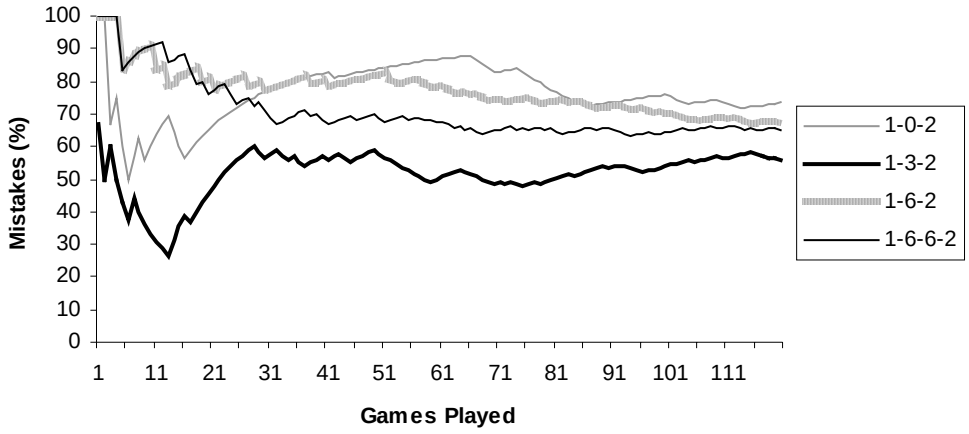



**Figure 6.5 Effect of varying depth and numbers of hidden nodes for a RBG of *incomplete* information**
Reducing the either the depth or the number of nodes increased the volatility from the ANN. Increasing the number of hidden nodes increased the learning time, shown as the network settling to a plateau, but at a higher probability that the network will make a mistake on the next game. The configuration of 1-3-2 was considered optimal.

The tests were repeated for a game of complete information where both players know the size of the pot they are bargaining for. Figure 6.6 shows the effect of varying the depth, i.e., number of hidden layers and the number of hidden nodes in each layer, for a game of *complete* information.

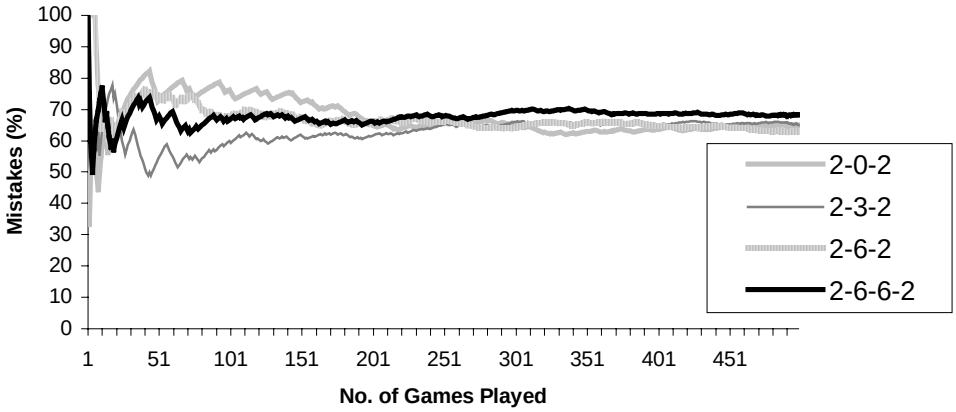



**Figure 6.6 The effect of varying the depth and number of hidden nodes for a RBG of *complete* information**
With no hidden layer the behaviour from the ANN was more erratic with a higher level of mistakes made on average. Increasing the number of hidden layers and nodes (2-6-6-2) reduced the volatility in the graph, which continued to decline, although taking longer to settle to a plateau. For this reason 2-6-6-2 was considered optimal.

With fewer hidden nodes and no hidden layer the behaviour from the ANN tended to be erratic and unpredictable. With more hidden nodes the time to learn increased as shown by an unsteady decline settling to a plateau. The configuration of 2-6-6-2 produced a graph of least volatility and continued to decline although taking longer to settle to a plateau thus, for a game of *complete* information the configuration of 2-6-6-2 with a learning rate of 0.1 was considered optimal.

#### 6.4.1.5 Conclusion

From this first set of experiments the preliminary results suggest that the Selective Bootstrap weight update rule does indeed learn although learning is erratic with the ANN more often than not accepting the first offer made. This is reflected in the fact that the ANN made a high number of mistakes (>50%) and the average number of turns per game was 1. In this set of experiments the optimum configuration and parameters were determined for the Selective Bootstrap network. A similar set of experiments follow to determine the optimal configuration for the second ANN in the model of Figure 3.3, the Temporal Difference Network. Once the optimum network configuration and parameters had been determined, experiments were carried out to see how both ANNs behaved competing against an artificial opponent and the results are compared.

### 6.4.2 Temporal Difference feed forward network playing an Artificial Opponent in the RBG

#### 6.4.2.1 Introduction

The motivation for this experiment is firstly to test the hypothesis that the higher brain functions can be modelled with the Temporal Difference weight update rule (Sutton, 1998) and secondly to optimise the Temporal Difference network's configuration. In addition, the results of the Temporal Difference network playing an artificial opponent in the RBG is compared with the Selective Bootstrap network playing an artificial opponent in the RBG. As discussed in Chapter 4, Temporal Difference (TD) learning is a reinforcement method used as a model of classical conditioning learning from psychology (Sutton and Barto, 1998). The Temporal Difference update rule maintains an approximation of the expected return of future rewards and includes a discount rate for determining the value of future rewards; therefore it can be considered far-sighted, i.e., concerned with long-term goals, for this reason it was considered suitable to model the higher brain functions at an abstract level. In addition, there is also again a requirement to verify the RL techniques to be used in general-sum games for the reasons listed in Section 6.2.

As in the previous experiment, with the ANN implemented with the Selective Bootstrap rule, the ANN with the TD weight update rule competes against an artificial opponent whose strategy is random. In this experiment TD is implemented as *TD(0)*, as only one state preceding the current one is changed by the TD error in contrast to *TD(λ)*, where all eligible states are changed by

the TD error (Sutton and Barto, 1998). This is because in the RBG game there are only a few states, as shown in Figure 6.7.

| State (Offer | Pot) | Action (Accept/Decline | Counteroffer) | Value Function V(St) |
|---|---|---|---|---|
| 0 | 0 | A | * | 0 |
| 0 | >0 | D | * | 0.5 |
| <0.5 | * | D | <0.5 | 0.5 |
| <0.5 | * | D | >=0.5 | 0.5 |
| <0.5 | * | A | * | 0.5 |
| >=0.5 | * | D | <0.5 | 0.5 |
| >=0.5 | * | D | >=0.5 | 0.5 |
| >=0.5 | * | A | * | 1 |

**Figure 6.7 The look-up table used in the Temporal Difference learning in the Rubinstein's Bargaining Game**
Initial values for the learned Value Function *V* for the state at time *t* ($S_t$). *V* is the latest estimate of the probability of success (winning) from that state *S* if the corresponding action is taken. * is a wildcard, i.e., any value between 0 and 1. The initial values are based on the following assumptions: if the offer is more than half the pot and the network has accepted, then the probability of winning is 1; similarly if offer is zero and the network has accepted, the probability of the network winning from this state is zero. The initial estimates of all other states are set to 0.5 that is the network has a 50% chance of winning.

The *State* is the input to the ANN. For RBG this is the opponent's offer and, if the game is a game of complete information, the current pot size. Temporal Difference learning (*TD(0)*) in this experiment is implemented with a look-up table, which is in effect the Value Function *V*. The Value Function gives an estimation of the probability of success given the current state of the environment and the ANN's action at time *t*. For this experiment the initial values for the Value Function for each possible state/action pair in the RBG are given in Figure 6.7. The initial values for the look-up table are derived from the following assumptions: if the offer is equal to the pot or more than half of the pot and the ANN has accepted, then the probability of winning is 1; similarly if the offer is zero and the ANN has accepted, then the probability of the network winning from this state is zero. The initial estimates of all other

states are set to 0.5 that is the network has a 50% chance of winning. Eq. 6.6 is used to update the estimate of the probability of success for the previous state, $V(S_p)$ from Figure 6.7. Eq 6.6 is adapted from Sutton and Barto (1998):

$$V(S_p)=V(S_p)+\alpha\left[V(S_c)-V(S_p)\right] \qquad (6.6)$$

where $V(S_p)$ is the value of the previous state (the offer and pot size) and the matching action (accept/decline, counteroffer) for the previous time ($p$). Similarly $V(S_c)$ is the Value Function of current state (the offer and pot size) and the matching action (accept/decline, counteroffer) for the current time ($c$), $\alpha$ is the step-size parameter or learning rate. The estimation of the probability of the network winning from each state/action pair is updated at the end of each turn of the game.

The Value Function for the current time $V(S_{t+1})$ and the previous time $V(S_t)$ are used to calculate the temporal difference error $\delta_t$ given by Eq. 6.7:

$$\delta_t = r_{t+1} + \gamma V(S_{t+1}) - V(S_t) \qquad (6.7)$$

where $r_{t+1}$ is the reinforcement signal at time $t$+1 (for a reward this will be 1), $\gamma$ is the discount rate of future rewards, $V(S_{t+1})$ is the Value Function for the state, that is, the probability of winning from the state at time $t$+1 and $V(S_t)$ the probability of winning from the state at time $t$ where state is taken from Figure 6.7. For the Temporal Difference update rule, the weights are updated based on this temporal difference error. More specifically the change in weights is given by Eq. 6.8:

$$\Delta w_t = \alpha \delta_t x_t \qquad (6.8)$$

where $\alpha$ is the step-size parameter and $x_t$ is the input at time $t$ to the neuron. Both Eq. 6.7 and Eq. 6.8 are adapted from Sutton and Barto (1998).

#### 6.4.2.2 Methodology

As in the previous experiment with Selective Bootstrap network, the network plays an artificial opponent whose accept or decline response is generated randomly and if the artificial opponent rejects the network's offer it then generates a counteroffer randomly. Hence, the system configuration is the same as in Figure 6.1 with the ANN in this case implemented with the TD weight update rule. The game follows the same pattern as in the previous experiment, i.e., the ANN and the artificial opponent go through a number of turns for each game, each taking it in turns to make an offer, which is either accepted or rejected, and a counteroffer is made.

As in the case of the Selective Bootstrap weight update rule, the TD weight update rule is implemented as a multi-layer feed forward Artificial Neural Network with non-linear nodes, i.e., the output of the node is calculated using the Sigmoid threshold function of Eq. 4.8. The network is rewarded (i.e., $r_{t+1}$ is set to one) and penalised (i.e., $r_{t+1}$ is set to zero) as described in the previous experiment (Section 6.4.1.2).

The criterion for measuring how successfully the ANN learns is the same as that of the previous experiment, i.e., the number of *mistakes* the ANN makes, the share of the pot it wins and finally the length of the game.

#### 6.4.2.3 Test Procedure

The system is configured in the same way as in the previous experiment, that is:

- For a game of *complete* information the ANN has as input the opponent's offer and pot size. An offer is represented as a real value between zero and one. The pot is a fraction of the previous pot and is represented as a value between zero and one.
- For a game of *incomplete* information the ANN has as input just the opponent's offer.
- The ANN has two outputs represented by real values between zero and one; one output either accepts or declines the opponent's offer. If the ANN rejects the offer then the other output is the ANN's counteroffer. If the ANN accepts the offer then the other output is ignored. An accept response is represented as any value greater than or equal to 0.5. A rejection is represented as any value less than 0.5.

The bias was implemented as in the previous experiment, i.e., as a node whose weight is trainable in the same way as the other nodes in the network. As before the weights were initialized to random values and the results are compared for multiple trials and the best selected. Again the same elements of the ANN were varied during testing to determine the optimum configuration, i.e., the number of hidden layers and neurons in each layer, and the learning variables, which in the case of the TD network are the step-size and discount

rate. How successfully the ANN learns is measured by the success criteria as listed in Section 6.4.1.2.

#### 6.4.2.4 Results

The objective of this set of tests was to find the optimum configuration of the Temporal Difference Network (TD) for both a game of incomplete information, i.e., where the players do not know the size of the pot and a game of complete information, i.e., where the players know the size of the pot. Figure 6.8 shows the effect of varying the step-size parameter for a game of *incomplete* information.

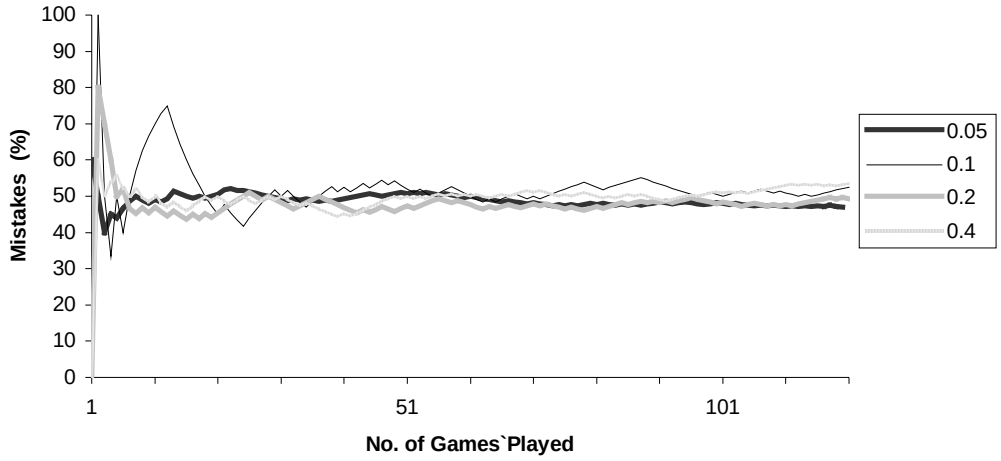



**Figure 6.8 Effect of varying the step-size parameter for a game of *incomplete* information**
Varying the step-size parameter had no dramatic effect. The graph shows the best learning curve from at least three test for each parameter tested. In each case the tests were carried out on an ANN configured as 1-3-2 and the discount rate was held at 0.5. The step-size parameter of 0.05 was considered as optimum for this configuration.

Figure 6.8 shows the best learning curve from at least three tests for each step-size parameter tested. In each case the tests were carried out on an ANN of 1 input node (to represent the artificial opponent's offer), a hidden layer of 3 nodes and an output layer of two nodes to represent the accept/decline

response and counteroffer. The discount rate was held at 0.5. The graphs show the probability of making a mistake based on the total number of mistakes made over the total number of turns in 121 games. As in the previous experiment for the Selective Bootstrap, although the trial consisted of 1000 games, it was found that the weight changes and the changes to the number of mistakes to be insignificant after approximately 120 games. Varying the step-size parameter had no dramatic effect. The learning curves decayed rapidly, but then settled down to a plateau. The step-size parameter of 0.05 produced a learning curve of a steady gentle decay and was considered optimum for this configuration.

As in the experiments for the Selective Bootstrap network, the test was repeated for a game of *complete* information, i.e., the players know the size of the pot. Figure 6.9 shows the effect of varying the step-size parameter for a game of *complete* information.

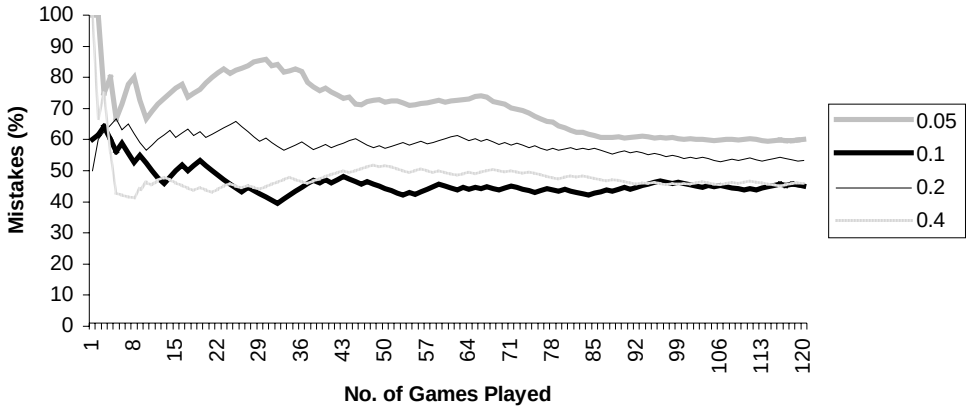



**Figure 6.9 The effect of varying the step-size parameter for a game of *complete* information**
The graph shows the best learning curve from at least three test for each parameter tested. In each case the tests were carried out on an ANN with a configuration of 2-6-2 and the discount rate was held at 0.5. The step-size parameter of 0.1 was considered as optimum for this configuration.

Again the graph selected is the best learning curve from at least three tests for each parameter tested. In each case the tests were carried out on an ANN of 2 input nodes (one to represent the artificial opponent's offer and another to represent the pot), a hidden layer of 6 nodes and an output layer of two nodes to represent the accept/decline response and counteroffer. The discount rate was held at 0.5. Again, the changes to the weights and the number of mistakes were insignificant after 120 games. The results show that all the learning curves decayed rapidly and somewhat unsteadily, but then settled down to a plateau. The step-size parameter of 0.1 was selected as optimum, as this produced the curve with the lowest percentage of mistakes.

The effect of varying the depth, i.e., the number of hidden layers, and the number of hidden nodes was tested. For a game of *incomplete* information the step-size parameter was held at 0.05 and for a game of complete information the step-size parameter was held at 0.1. The discount rate was held at 0.5 for both sets of experiments. Figure 6.10 shows the effect of varying the depth and number of hidden nodes in each layer on the learning curve for a RBG of *incomplete* information. The best learning curve was selected from at least three tests for each configuration tested. With fewer layers and nodes the graph tended to be more volatile (1-0-2). Increasing the number of hidden nodes reduced the volatility of the graph, but took longer, i.e., more games, to settle to a plateau (1-6-2 and 1-6-6-2). Thus the configuration of 1-3-2 was selected as optimum.

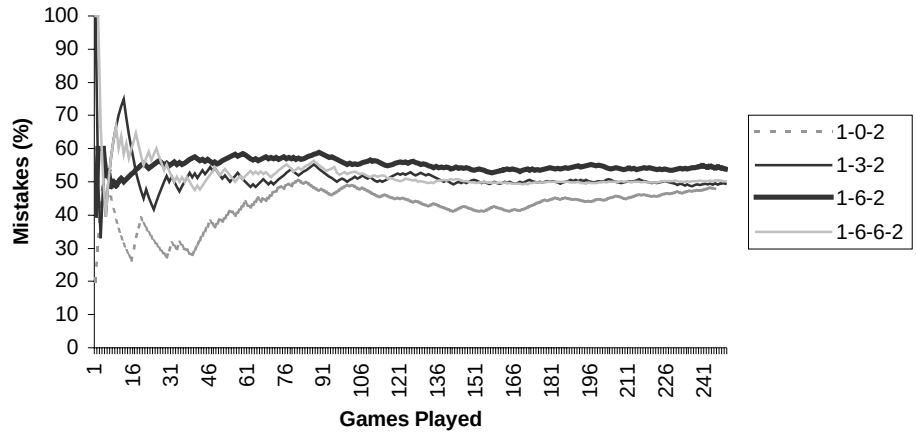



**Figure 6.10 The effect of varying the depth and number of hidden nodes for a RBG of *incomplete* information**
With no hidden layer learning did not appear to take place with the graph 1-0-2 stabilizing at a plateau at a higher level than at the start of play. Increasing the number of hidden nodes increased the learning time, but reduced the volatility in the learning curve. 1-3-2 was selected as optimum.

The test was repeated for a game of *complete* information. Figure 6.11 shows the best learning curve selected from at least three tests for each configuration tested for a RBG of *complete* information.

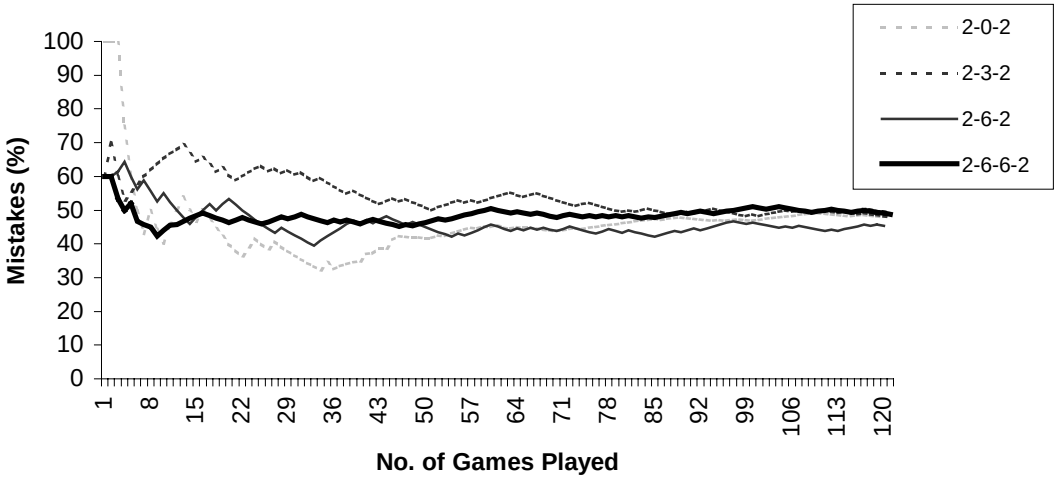



**Figure 6.11 The effect of varying the depth and number of hidden nodes for a RBG of *complete* information**
With no hidden layer the graph was more volatile. With more hidden nodes the graph was less volatile settling to a plateau. The configuration 2-6-6-2 was selected as optimum.

The results in Figure 6.11 were less dramatic than that of Figure 6.10. With no hidden layer the graph was more volatile. Increasing the number of hidden nodes reduced this. The configuration of 2-6-6-2 was selected as optimum as this produced a learning curve that was less volatile and that settled to a plateau equal or better than the alternate configurations.

The final test in determining the optimal configuration for the TD network was to determine the effect of varying the discount rate. Again the best curves from multiple experiments were selected. Figure 6.12 shows the results of varying the discount rate for a RBG of *incomplete* information, the step-size parameter was held at 0.05 and the ANN configured at 1-3-2.

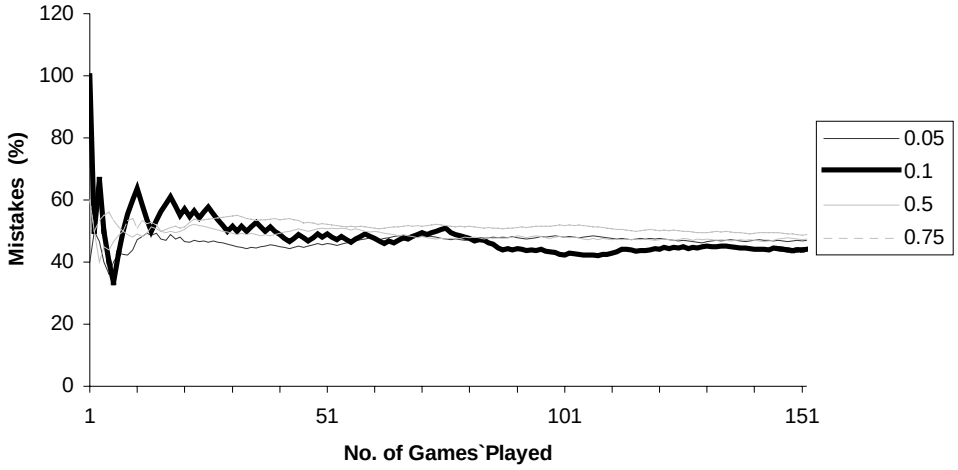



**Figure 6.12 The effect of varying the discount rate for a RBG of *incomplete* information**
The graphs show the best curves selected from multiple experiments. All the learning curves behaved the same, decaying rapidly then settling to a plateau lower than that at the start of play. The discount rate of 0.1 was selected as optimum as this settled to the lowest number of mistakes of all the discount rates tested.

Varying the discount rate for a RBG of *incomplete* information had no dramatic effect. All the learning curves decay rapidly though somewhat

erratically, settling to a plateau lower than at the start of play. The discount rate of 0.1 was selected as optimum as it settled to the lowest level of all the discount rates tested.

Figure 6.13 shows the results of varying the discount rate for a RBG of *complete* information the step-size parameter was held at 0.1 and the ANN configured at 2-6-6-2.

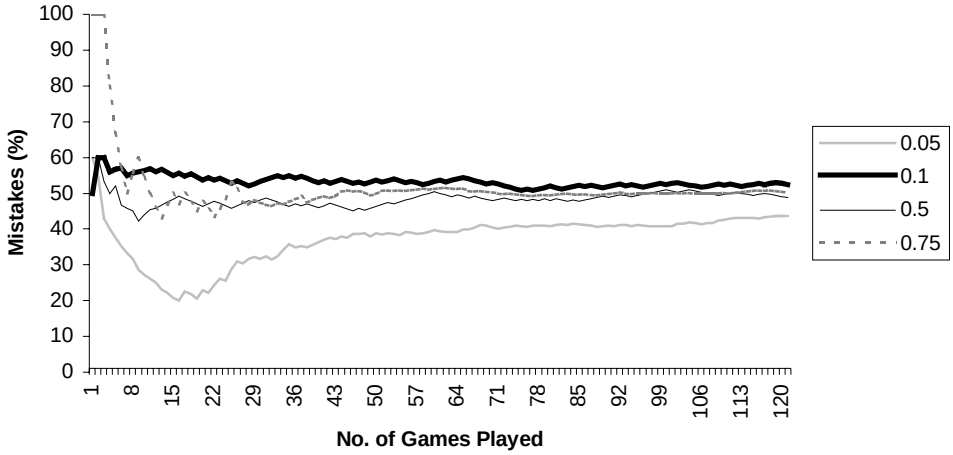



**Figure 6.13 The effect of varying the discount rate for a RBG of *complete* information**
Again the learning curves were the best curves selected from multiple experiments. All the learning curves behaved the same, decaying rapidly then settling to a plateau. Although a discount rate of 0.5 was consistently lower than the discount rate of 0.1 the graph showed more fluctuation and was discarded as being too middling for the purpose of these experiments, as effectively a discount rate of 0.5 is saying "we value future rewards 50% of the time". Hence a discount rate of 0.1 was selected as optimum.

Varying the discount rate for a game of *complete* information produced some interesting results. All the learning curves behaved the same, decaying rapidly then settling to a plateau, however, a low discount rate of 0.05 produced a learning curve that was more erratic and hence less predictable. The discount rate of 0.5 was consistently lower than the discount rate of 0.1 however, the graph showed more fluctuation and a discount rate of 0.5 was considered as

being too middling, as effectively a discount rate of 0.5 is saying "we value future rewards 50% of the time". Hence a discount rate of 0.1 was selected as optimum.

Having selected the optimum ANN configuration and parameters for both ANNs, experiments were carried out to determine how the ANNs behaved and to test the hypothesis that the lower brain functions can be modelled with the Selective Bootstrap weight update rule (Widrow et al., 1973) and that the higher brain functions can be modelled with the Temporal Difference weight update rule (Sutton, 1998). Figure 6.14 compares the typical results (number of mistakes and winnings expressed as a percentage of the total for the game) for both weight update rules playing an artificial opponent in a RBG of *incomplete* information.

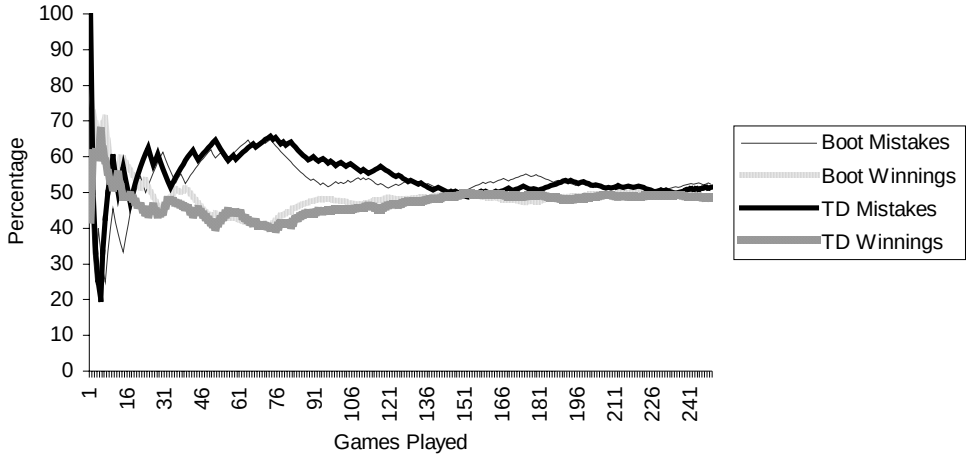



**Figure 6.14 Results for a Temporal Difference network and a Selective Bootstrap network playing an artificial opponent in a RBG of *incomplete* information**

A comparison of the number of mistakes and winnings per game. The TD Network is configured as 1-3-2 with a step-size parameter of 0.05 and a discount rate of 0.1 and the Selective Bootstrap Network as 1-3-2 with a learning rate of 0.1, which were selected as the optimum configuration from the above experiments. The more games played the less number of mistakes were made, indicating that both ANNs have learnt. However, the results are not significantly different.

The Temporal Difference Network is configured as 1-3-2 with a step-size parameter of 0.05 and a discount rate of 0.1 and the Selective Bootstrap Network as 1-3-2 with a learning rate of 0.1, which were selected as the optimum configuration from the above experiments. The ANNs played 5000 games, however, after a certain point (200 games) the amount of change to the mistakes made and the winnings per game is so small as to be insignificant. The results show that the more games the networks played, the less number of mistakes the networks made, indicating that learning has occurred in both networks. However, when the results for each of the networks are compared they are not significantly different.

Figure 6.15 compares typical results (number of mistakes and winnings expressed as a percentage of the total for the game) for both weight update rules playing an artificial opponent in a RBG of *complete* information.

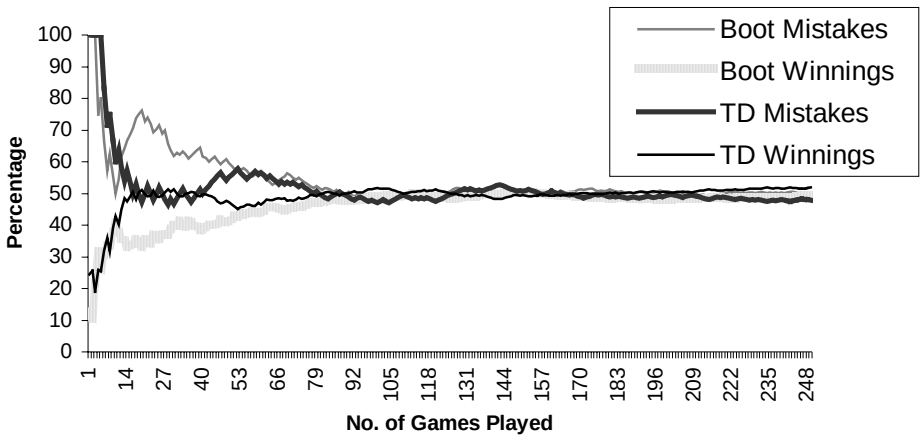



**Figure 6.15 Results for a Temporal Difference network and a Selective Bootstrap network playing an artificial opponent in a RBG of *complete* information**

A comparison of the number of mistakes and winnings per game. Both networks are configured as 2-6-6-2. The TD Network has a step-size parameter of 0.1 and a discount rate of 0.1 and the Selective Bootstrap Network has a learning rate of 0.1. The TD network fares slightly better in that, on average the TD network makes fewer mistakes and wins slightly more of the pot than the Selective Bootstrap network.

Both ANNs were configured as 2-6-6-2 as this was considered the optimal configuration from the experiments above. The Temporal Difference Network had a step-size parameter of 0.1 and a discount rate of 0.1 and the Selective Bootstrap Network had a learning rate of 0.1, which were selected as the optimum parameters from the above experiments. Again, the networks played 5000 games, however, once again, after a certain point (250 games) the amount of change to the mistakes made and the winnings per game is so small as to be insignificant. In this case, the TD network does slightly better than the Selective Bootstrap network, in that it made fewer mistakes (47%) and won a slightly larger slice of the pot (51%).

#### 6.4.2.5 Conclusion

A mistake was defined with the aim of helping the ANN to maximize its share of the pot. Hence it was expected that reducing the number of mistakes made would result in an increase in the share of the pot. The results shown in the above graphs support this. The less number of mistakes made by either ANN results in the ANN gaining a greater share of the pot.

The decrease in the number of mistakes the ANNs make, indicates that the networks are behaving as expected, that is, the more games the networks play, the less mistakes it makes, indicating that the network has "learned". On average, the Temporal Difference network gained a slightly larger slice of the pot as compared to the results of the Selective Bootstrap network as shown in Figure 6.15 and Figure 6.14. This may be can be explained by the tendency for Selective Bootstrap to accept a less than optimal offer reflected in the average turn per game for the Selective Bootstrap network as being one. This

supports the hypothesis that the player least affected by the delay as a result of the bargaining process, in this case the Temporal Difference network, will receive the larger slice of the pot (Kreps, 1990). This hypothesis is tested further in the next set of experiments.

### 6.4.3 2-ANNs Playing the Rubinstein's Bargaining Game

#### 6.4.3.1 Introduction

In this experiment the two ANNs compete as two autonomous agents learning simultaneously in a shared environment. The motivation for this experiment is to further test the hypothesis that the Temporal Difference network exhibits behaviours of the higher brain processes and that the Selective Bootstrap network exhibits behaviours of the lower brain processes. A version of RBG with one player implemented as an ANN with the Selective Bootstrap weight update rule and the other player implemented as an ANN with the Temporal Difference weight update rule was played. In this case the ANNs are two autonomous agents learning in a shared environment. Although each ANN is learning independently the existence of the other ANN is not ignored, as there is a requirement for cooperation between the players, since it is in both ANNs' interest to reach an agreement sooner rather than later, as the resource diminishes with time. As the ANN implemented with TD learning is the player least affected by the delay, i.e., the less impatient, it is expected that it will receive the larger slice of the pot.

#### 6.4.3.2 Methodology

In this version of the RBG, one player is implemented as an ANN with the Selective Bootstrap weight update rule and the other player is implemented as

an ANN with the Temporal Difference weight update rule. The system configuration for a RBG of complete information, i.e., where both the players know the size of the pot, is shown in Figure 6.16. A game of incomplete information, (i.e., where the players do not know the size of the pot) was modelled as having one input to represent the opponent's offer.

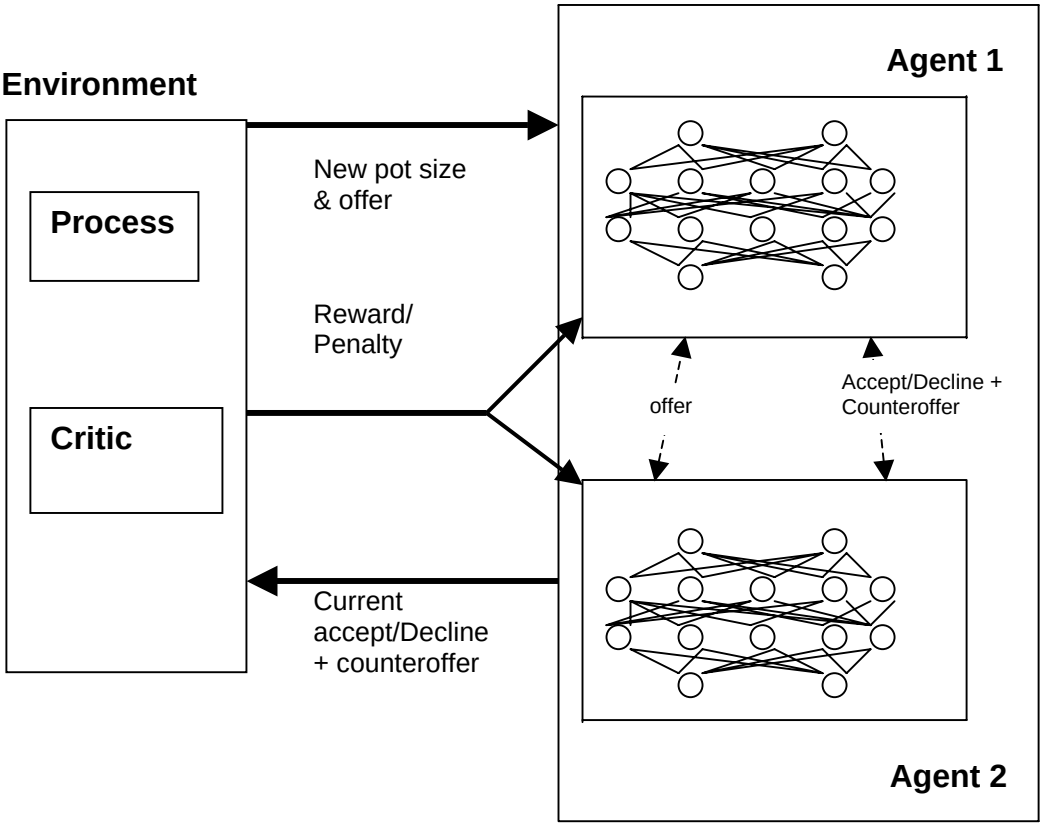



**Figure 6.16 System Configuration for 2-ANNs playing a RBG of *complete* information**
The environment has a *process* that initializes the pot size and the offer at the start of the game and a *critic* who rewards or penalizes the networks at each turn of the game and at the end of the game.

At the end of each turn, the ANN whose turn it is (for illustration purposes call this ANN *A* and let the opposing ANN, be *B*) is rewarded (i.e., $r_{t+1}$ is set to one), if:

1. the offer from *B* is lower than *A*'s previous offer and *A* declines the offer or
2. the offer from *B* is higher than *A*'s previous offer and *A* either:
    a. accepts the offer or
    b. makes a counteroffer higher than the incoming offer

For all other actions ANN *A* is penalised (i.e., $r_{t+1}$ is set to zero). At the end of the game, the ANN that received the greater share of the pot, is rewarded and the other ANN is penalised. For each ANN, the average winnings, total winnings, the total number of mistakes and the average number of mistakes are recorded and compared. Each ANN is evaluated by the number of mistakes it made and the share of the pot it wins.

#### 6.4.3.3 Testing Procedure

The ANNs' configuration and parameters are held as those selected as optimum in the experiments in Section 6.4.1.4 and Section 6.4.2.4, that is, for a game of *incomplete* information the ANNs are configured as 1-3-2 with a step-size parameter of 0.05 and a discount rate 0f 0.1 for the Temporal Difference network and a learning rate of 0.1 for the Selective Bootstrap network; for a game of *complete* information the ANNs were configured as 2-6-6-2 and all parameters were held at 0.1. Bias was implemented as in the previous experiments, i.e., as a node whose weight is trainable in the same way as the other nodes in the ANN. In the Rubinstein's Bargaining game the player that moves first has the advantage, as the first player can determine what the other player might receive (Rubinstein, 1982). To overcome this problem of first player advantage, the starting ANN was selected at random.

Tests were run for games of complete and incomplete information. Variations of the RBG were also tried where the players have different policies or strategies, for example, one player (either the TD network or the Selective Bootstrap network or both) operates with a greedy policy. A greedy policy is modelled with an initial offer of zero, i.e., the first player is not prepared to bargain and has kept all of the pot. For a non-greedy policy an initial random offer is made indicating that the first player is willing to bargain.

#### 6.4.3.4 Results

Figure 6.17 shows the results of the TD network competing with the Selective Bootstrap network in a non-greedy RBG of *incomplete* information, i.e., neither player knows the size of the pot and both players are willing to cooperate in order to reach an agreement sooner rather than later.

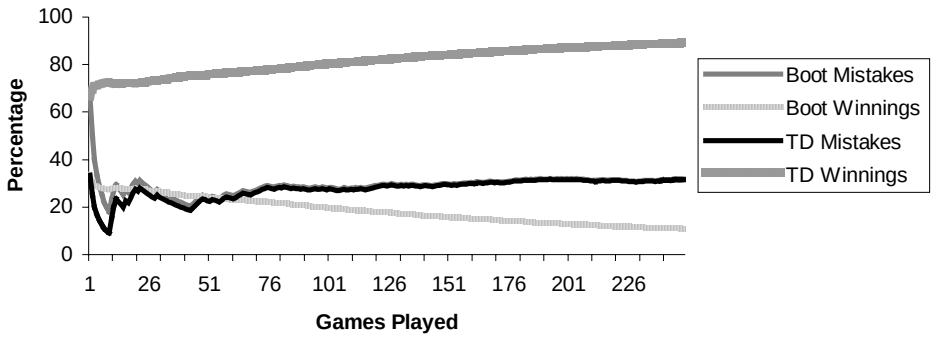



**Figure 6.17 TD network *versus* Selective Bootstrap network in a RBG of *incomplete* information**
A TD network competing against a Selective Bootstrap network playing RBG, played as a game of incomplete information with a non-greedy policy, i.e., both players are willing to cooperate in order to reach an agreement sooner rather than later. The TD network fares better than the Selective Bootstrap in that it consistently wins the greater share of the pot (89%) and makes less mistakes on average (25% as opposed to an average of 40% for the Selective Bootstrap network).

Figure 6.18 shows the results of the TD network competing with the Selective Bootstrap network in a non-greedy RBG of *complete* information, i.e., both player knows the size of the pot they are playing for and are willing to cooperate in order to reach an agreement sooner rather than later.

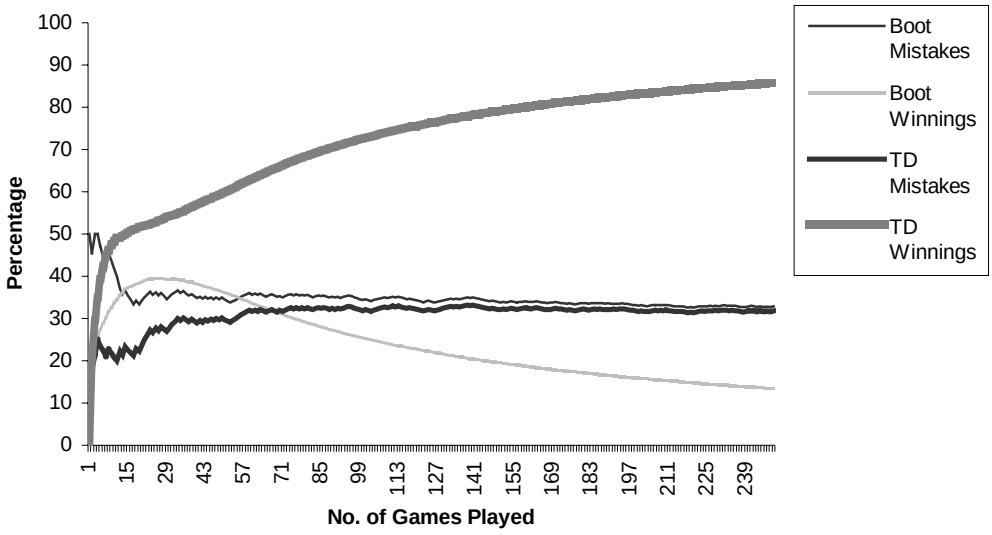



**Figure 6.18 TD network *versus* Selective Bootstrap network in the RBG of *complete* information**
A Temporal Difference Network competing against a Selective Bootstrap network playing RBG, played as a game of complete information with a non-greedy policy. Again the TD network fares better than the Selective Bootstrap in that it consistently wins the greater share of the pot (86%) and makes slightly less mistakes on average.

The graphs in Figure 6.18 show the ANN's behaviour over multiple experiments run over a number of games. After a certain number of games (250) the amount of change to the number of mistakes made and the amount of pot won is so small as to be insignificant. The results show that the ANNs' behaviour settled down to a plateau with the TD network gaining a significant majority of the pot (86%) as opposed to the Selective Bootstrap network, which gains significantly less of the pot (14%). Different types of RBGs were tried. For example, an experiment was conducted where both players had a

policy of greedy play competing in the RBG. The results are the same, i.e. the TD network does slightly better than the Selective Bootstrap network in terms of the number of mistakes and the TD network gains a significant larger slice of the pot than the Selective Bootstrap network.

#### 6.4.3.5 Conclusion

In summary, the following observations were made for both weight update rules, i.e., Selective Bootstrap and Temporal Difference. For the Rubinstein's Bargaining game even though the number of turns was set at ten, play never exceeded five turns per game. Either there was nothing left in the pot or the ANN accepted the opponent's offer and the game ended. Both ANNs would seem to have learnt, supported by the decrease in the number of mistakes made by both ANNs. The Selective Bootstrap weight update rule for the reinforcement learning algorithm performed poorer both in the average number of mistakes made and in the pot won compared to the Temporal Difference weight update rule. This was made clear in the final set of experiments, which saw the TD network compete with the Selective Bootstrap network in variations of play for the RBG.

In conclusion, the TD network did significantly better than the Selective Bootstrap network in that TD gained the largest share of the pot and made fewer mistakes. For the Temporal Difference network maintaining a history of previous rewards (even though it is only one state preceding the current) and including a discount rate would appear to have helped the TD network to learn better than the Selective Bootstrap network. The ANNs behaved as expected in that the Temporal Difference network, as the player least effected by delay,

received the larger slice of the pot (Kreps, 1990). The results support the theoretical premise, made at the beginning of the chapter, that the Selective Bootstrap network is myopic, and hence more impatient than the Temporal Difference network. For the Selective Bootstrap network the waiting time till the reward is received, in this case its share of the final the pot, is more costly and hence at times it accepts an unreasonable offer (i.e., less than 50%).

## 6.5 Explaining Self-control with the Iterated Prisoner's Dilemma game

In this set of experiments the two ANNs compete in the Iterated Prisoner's Dilemma (IPD) game. The IPD is appropriate since, as discussed in Chapter 2 Section 2.3, an experiment by Brown and Rachlin (1999) explored the relationship between self-control and cooperation, using human subjects playing a version of the IPD game. The IPD has been used to model the evolution of human cooperation (Axelrod and Hamilton, 1981). The IPD consists of two players who compete with each other repeatedly. Each player can either cooperate or defect. Defection is the higher payoff for the individual player, however if both players defect then the resulting payoff for both is worse. The goal is to maximise the total payoff.

A preliminary version of this set of experiments appeared in Banfield and Christodoulou (2005), and was presented at 9th Neural Computation and Psychology Workshop.

### 6.5.1 The Temporal Difference Network *versus* the Selective Bootstrap Network playing an IPD game with local reward

#### 6.5.1.1 Introduction

The motivation for this set of experiments is: firstly to verify the model proposed in Chapter 3 shown in Figure 3.3 and developed in Section 6.4, of the higher brain functions competing with the lower brain functions for control of the organism, and secondly to investigate the idea proposed by Brown and Rachlin (1999) that there is a direct relationship between cooperating with one's self and self-control behaviour. These ideas were discussed in detail in Chapter 2 Section 2.3. To summarize, Brown and Rachlin (1999) concluded that the path to greater self-control is in our confidence that we will continue to cooperate with our selves in the future. In the implementation of the IPD game in this thesis, to cooperate with our selves is represented by the reward for mutual cooperation, the top left hand box in the IPD payoff matrix (*CC*) in Figure 2.3. Hence, in this experiment and in the following experiments, the probability of continuing to cooperate with our selves in the future, is measured by the number of times the players select the reward for mutual cooperation (*CC*). In this experiment and in the following experiment, the effect of different reward structures on the patterns of play, for example the number of times the players play a game of mutual cooperation, are investigated. In this experiment, each ANN in the 2-ANNs model receives a different reward and hence each ANN is playing to maximize its own payoff, for this reason we could say that the ANNs are playing an IPD game of *selfish play*. In the following experiment the ANNs

receive the same global reward with the aim of maximizing the reward for the organism as a whole, where the 2-ANNs model represents the organism.

It follows from the results of the RBG experiments that the Selective Bootstrap network best represents the lower brain system associated with myopic behaviour in that it more often than not, accepted a less than optimal offer reflected in the size of the pot won. The Temporal Difference network exhibited behaviour associated with the higher brain functions such as planning and control in that it did not accept the first offer made and appeared to hold out for a more acceptable offer, reflected in that the TD network won the greater share of the pot.

The Selective Bootstrap rule is implemented as described for the RBG, the same equations for updating the synaptic weights apply (see Section 6.4.1.1). The TD rule for IPD is implemented as described for RBG, but with a different look-up table, which is in effect the value function *V*. In this experiment, for the IPD game, the *State* is the opponent's last action, i.e., to cooperate or to defect and the *action* is the player's response based on the opponent's last action. The Value Function is the probability of receiving the highest payoff, given the current state of the environment and the ANN's action. Implemented in this way, the TD network maintains a history of the opponent's previous action even though it is only one state preceding the current one. Figure 6.19 shows the initial values for the Value Function for each state/action combination.

| State D/C | Action D/C | Value Function $V(S_t)$ |
|---|---|---|
| D | D | 0 |
| C | C | 0.5 |
| D | C | 0 |
| C | D | 1 |

**Figure 6.19 The look-up table for Temporal Difference learning in the IPD Game**
A table of initial values for the learned Value Function *V*, where *D* stands for Defect and *C* represents Cooperate. The initial values are based on the probability of gaining the highest payoff derived from the payoff matrix by Maynard Smith (1982).

The equations for updating the look-up table, calculating the TD error and the updates to the synaptic weights remain unchanged from those used in the RBG, i.e., Eq. 6.6, Eq. 6.7 and Eq. 6.8 respectively.

#### 6.5.1.2 Methodology

The ANN is configured with two inputs nodes to represent the opponent's previous action (a node to represent defection and a node to represent cooperation), and two output nodes representing a response (a node to represent defection and a node to represent cooperation). The nodes act like a binary switch, i.e., a value of 1 indicates that node is active. For example, if the opponent's previous action was to defect, then the defection node would be set with an input value of 1 and the node representing cooperation with a value of zero. The output of the nodes are calculated using the Sigmoid threshold function of Eq. 4.8, as in the experiments for the RBG. The output is normalized to either a value of 1 or zero. A value of 1 in the output node indicates that node is active. For example, if the ANN's response is to cooperate then the value of the defection node is zero and the node representing to cooperate is 1. A game consists of one or more rounds. The goal is to maximise the total payoff. Figure 6.20 shows the system configuration for this experiment. The system configuration is similar to the

one in Figure 6.16 for the experiment of 2-ANNs playing the RBG. In the case of the IPD game, the environment contains a *process* that initializes the input/state to the opponent's previous action (to defect or to cooperate) at the start of each round. The output/action is the ANN's action (to defect or to cooperate). The environment also contains a *critic* that assigns a reward or penalty, which is based on the payoff for each round and at the end of a game the ANN that has the highest accumulated payoff is rewarded and the other ANN is penalised.

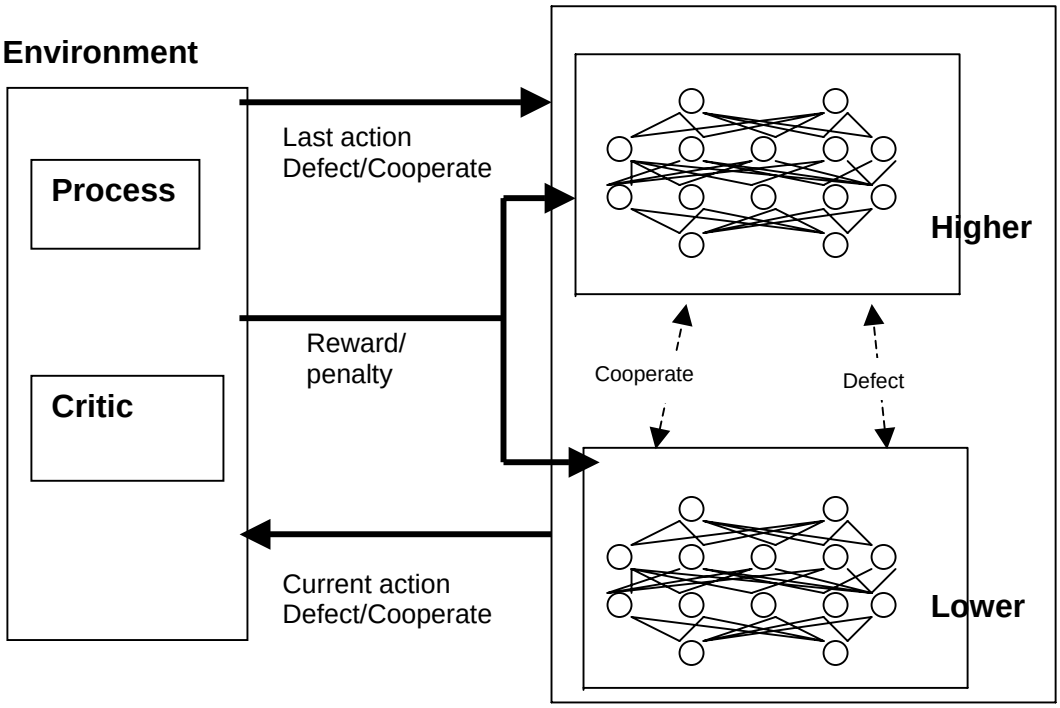



**Figure 6.20 System configuration for 2-ANNs playing IPD game**
The environment contains a *process* that initializes the input/state to the opponent's previous action (to defect or to cooperate) at the start of each round. The output/action is the network's action (to defect or to cooperate). The environment also contains a *critic* that assigns a reward or a penalty based on the payoff for each round and at the end of a game

The payoff matrix is adapted from the definition of the Prisoner's Dilemma game given by Maynard Smith (1982), summarised in Figure 6.21.

| | Cooperate | Defect |
|---|---|---|
| **Cooperate** | R | S |
| **Defect** | T | P |

**Rules:**
1. T>R>P>S
2. 2R>T+S

**Figure 6.21 Maynard Smith's Payoff Matrix for the Iterated Prisoner's Dilemma game**
Maynard Smith's Payoff Matrix for the Iterated Prisoner's Dilemma (Maynard Smith, 1982) where *T*: Temptation, *R*: reward, *P*: punishment, *S*: sucker's payoff. Rule 1 (T>R>P>S) defines the game. Rule 2 (2R>T+S) ensures that the payoff is greater to two players who cooperate, than a pair who alternately cooperate and defect. Rewards are shown for the row player.

In this thesis Temptation has a numerical value of 2, Reward a value of 1, Punishment a value of zero and Sucker's payoff a value of (–1). The payoff matrix to be used in this experiment is shown in Figure 6.22. The payoff matrix is explained in terms of higher and lower brain regions rather than row and column players. The reward in this first experiment is local, i.e., each ANN receives an individual reward, which is the payoff shown in the matrix in Figure 6.22. The actions are rewarded at each round of the game as in shown in Figure 6.22. The payoffs to the player representing the higher brain functions are listed first: (*C*,*C*) the reward for mutual cooperation (*R*) a value of (1,1), (*D,D*) the punishment for mutual defection (*P*) a value of (0,0), (*C,D*) the penalty for Sucker's payoff (*S*) a value of (–1) and (*D,C*) the reward for Temptation to defect (*T*) has a numerical value of (2,-1). The Temptation to defect is the highest immediate reward, representative of the smaller-sooner reward (*SS*) in the self-control problem illustrated in Figure 2.1 in Chapter 2. Mutual cooperation yields the highest reward in the long-term, representative of the larger-later reward (*LL*) in the self-control problem illustrated in Figure 2.1 in Chapter 2.

| | | Lower Player | |
|---|---|---|---|
| | | **C** | **D** |
| **Higher Player** | **C** | R=1, R=1<br>Reward for mutual cooperation | S=-1, T=2<br>Sucker's payoff and temptation to defect |
| | **D** | T=2, S= -1<br>Temptation to defect and sucker's payoff | P=0, P=0<br>Punishment for mutual defection |

**Rules:**
1. T>R>P>S
2. 2R>T+S

**Figure 6.22 The payoff matrix for the Prisoner's Dilemma game used in the simulation of the IPD game in this thesis.**
Defined by: Temptation to defect (*T*) must be better than the Reward for mutual cooperation (*R*), which must be better than the Punishment for mutual defection (*P*), which must be better than the Sucker's payoff (*S*); the average of the Temptation to defect and the Sucker's payoffs must not exceed the Reward for mutual cooperation. The Temptation to defect is the highest immediate reward, representative of the smaller-sooner (*SS*) reward in the self-control problem shown in Figure 2.1. Mutual cooperation (*C,C*) yields the highest reward in the long-term, representative of the larger-later (*LL*) reward in the self-control problem shown in Figure 2.1. The payoffs to the player representing the higher brain functions are listed first.

To illustrate this consider a real world example of the self-control problem. With reference to Figure 2.1, assume that for a student that at the beginning of the academic year the *LL* represents getting good grades. At some point later in time, the student receives an invitation to go to the pub and it is at this time that his *SS* becomes known, i.e. going to the pub. When invited to the pub, the student is faced with the self-control problem of staying at home and studying (the *LL* reward) or going to the pub and socializing (the *SS* reward). In Figure 6.22, the (*C,C*) is the *LL* reward of staying at home and studying leading to good grades and the (*D,D*) is the *SS* reward of going to the pub and socializing. If it is assumed that *C* is staying at home and *D* is going to the pub, then (*C,D*) could represent the situation of when asked to the pub you decide to stay at home, but do not study as effectively because you wish you had gone to the pub, and (*D,C*) could represent the situation of going to the

pub, but having a miserable time because you feel guilty about not studying.[5] At the end of the game the ANN with the highest payoff is rewarded, (i.e., $r_{t+1}$ is set to one) and the other ANN is penalized (i.e., $r_{t+1}$ is set to zero).

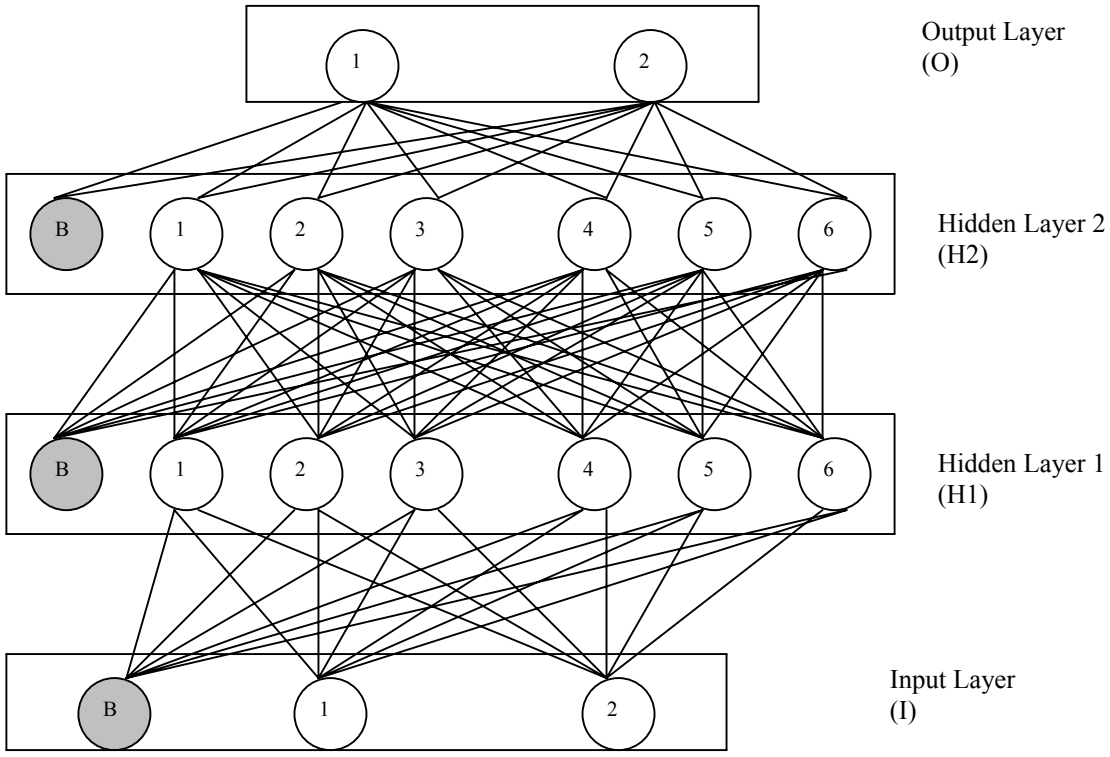
#### 6.5.1.3 Test Procedure

The ANNs are configured with two input nodes representing the opposing network's previous action (to defect or to cooperate) and two output nodes representing the ANN's response (to defect or to cooperate.) With this arrangement, the ANNs' configuration matches that of a RBG game of complete information, hence the networks topology and parameters are held as those selected as optimum for a RBG of complete information, i.e., the networks are configured as 2-6-6-2 and all parameters are held at 0.1. Bias was implemented as in the previous experiments, i.e., as a node whose weight is trainable in the same way as the other nodes in the ANN. Starting on a randomly selected initial weights means that each time the ANN plays different results can be expected, since learning starts at different points of error *versus* weights hypersurface. To overcome this disparity the results are held for three trials and the average shown on the graphs. The task of learning for this experiment is to find the best response based on the opponent's previous response; this is encoded in the hidden nodes. The best response in this case is to maximize the individual ANN's payoff. For evaluating the game, for each ANN the following are recorded, the pattern of play, i.e., the sequence of the ANN's actions to defect or to cooperate, the payoff for the round and the accumulated payoff for the game.

[5] Note the word cooperation has a non-conventional meaning in this thesis. Cooperation is defined as working together for a common end in which case (*D,D*) could be viewed as cooperation. In this thesis this is not so, cooperation means cooperating in order to gain the larger later reward (*LL*). Refer to Section 2.3 for further details.

### 6.5.1.4 Results

The topology implemented for both ANNs is shown in Figure 6.23. The weights are numbered from left to right as shown in the legend of weights in Figure 6.23.




| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | BI | H11 | **16** | I2 | H14 | **31** | H12 | H21 | **46** | H14 | H24 | **61** | BH2 | O0 |
| **2** | BI | H12 | **17** | I2 | H15 | **32** | H12 | H22 | **47** | H14 | H25 | **62** | BH2 | O1 |
| **3** | BI | H13 | **18** | I2 | H16 | **33** | H12 | H23 | **48** | H14 | H26 | **63** | H21 | O0 |
| **4** | BI | H14 | **19** | BH1 | H21 | **34** | H12 | H24 | **49** | H15 | H21 | **64** | H21 | O1 |
| **5** | BI | H15 | **20** | BH1 | H22 | **35** | H12 | H25 | **50** | H15 | H22 | **65** | H22 | O0 |
| **6** | BI | H16 | **21** | BH1 | H23 | **36** | H12 | H26 | **51** | H15 | H23 | **66** | H22 | O1 |
| **7** | I1 | H11 | **22** | BH1 | H24 | **37** | H13 | H21 | **52** | H15 | H24 | **67** | H23 | O0 |
| **8** | I1 | H12 | **23** | BH1 | H25 | **38** | H13 | H22 | **53** | H15 | H25 | **68** | H23 | O1 |
| **9** | I1 | H13 | **24** | BH1 | H26 | **39** | H13 | H23 | **54** | H15 | H26 | **69** | H24 | O0 |
| **10** | I1 | H14 | **25** | H11 | H21 | **40** | H13 | H24 | **55** | H16 | H21 | **70** | H24 | O1 |
| **11** | I1 | H15 | **26** | H11 | H22 | **41** | H13 | H25 | **56** | H16 | H22 | **71** | H25 | O0 |
| **12** | I1 | H16 | **27** | H11 | H23 | **42** | H13 | H26 | **57** | H16 | H23 | **72** | H25 | O1 |
| **13** | I2 | H11 | **28** | H11 | H24 | **43** | H14 | H21 | **58** | H16 | H24 | **73** | H26 | O0 |
| **14** | I2 | H12 | **29** | H11 | H25 | **44** | H14 | H22 | **59** | H16 | H25 | **74** | H26 | O1 |
| **15** | I2 | H13 | **30** | H11 | H26 | **45** | H14 | H23 | **60** | H16 | H26 | | | |

**Figure 6.23 The ANN's topology with weight legend**
The weights of the synapses are numbered left to right, the number is followed by the node at the start of the synapse followed by the node at the end of the synapse, e.g., weight 16 is associated with the synapse that starts at the second node (*2*) in the input layer (*I*) and ends at the fourth node (*4*) in hidden layer 1 (*H1*).

The number of rounds per game were initially held at 1000. A trial consists of three games of 1000 rounds. To avoid any first player advantage or disadvantage, the starting ANN was selected at random. The ANNs are rewarded or penalised at the end of each round and at the end of the game, as detailed above. Although reinforcement learning involves learning throughout one's lifetime (where learning in the case of ANNs is reflected in the change in the ANN's weights) it was found that in this experiment most learning occurred early in the game (around a 150 rounds). After this point, the amount of change to the weights, from both ANNs, was so small as to be insignificant. Figure 6.24 and Figure 6.25 show the weight changes for the Selective Bootstrap network and the Temporal Difference network respectively. The graphs show the typical variance across all trials. Given learning plateaus at, or around 150 rounds with the weight changes to be so small to be insignificant, a note was made for future experiments to limit the number of rounds per game to 250.

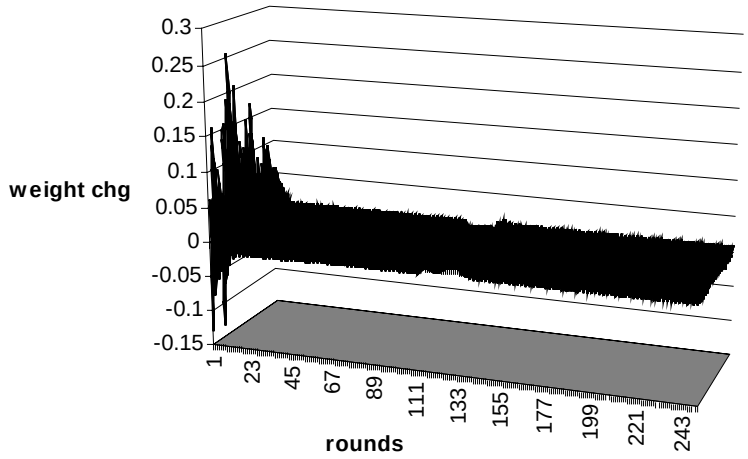



**Figure 6.24 Learning in the Selective Bootstrap Network for the IPD Game**
Typical weight changes for the Selective Bootstrap network competing in the IPD game. Most learning occurs before 150 rounds, with the weights changes after 150 rounds being so small as to be insignificant.

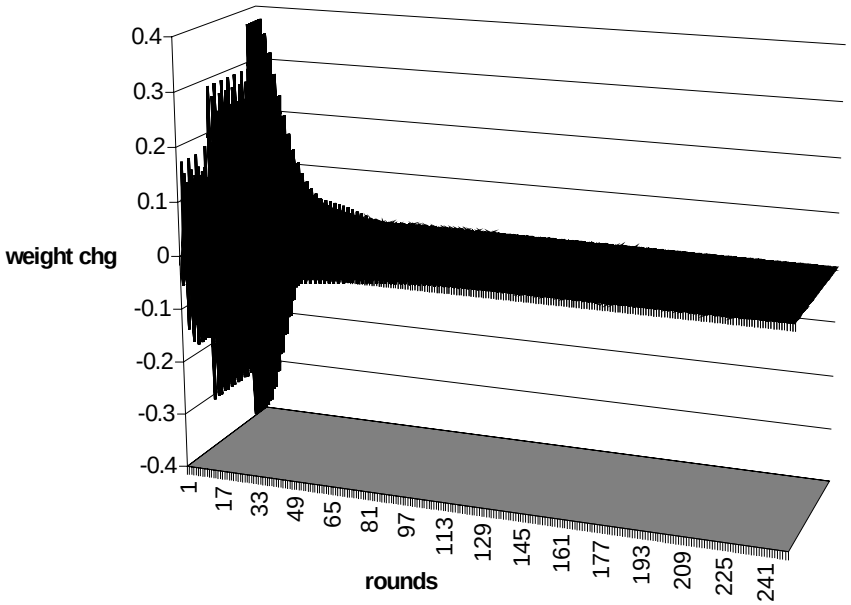



**Figure 6.25 Learning in the Temporal Difference Network for the IPD game**
Typical weight changes for the Temporal Difference network competing in the IPD game. Most learning occurs before 100 rounds, with weight changes after this point being so small as to be insignificant.

A trial is repeated 3 times and the total payoff and the average payoff for the three games for each ANN is recorded, as well as the net payoff, i.e., the sum of the total payoff for both ANNs. Figure 6.26 shows the average payoff for each ANN, the net payoff and the range, i.e., the minimum and maximum payoff for each ANN is also shown. Figure 6.26 shows the first 150 rounds even though each game was played for 1000 rounds. It was found that the disparity in the ANNs' payoff and the range continued to increase.

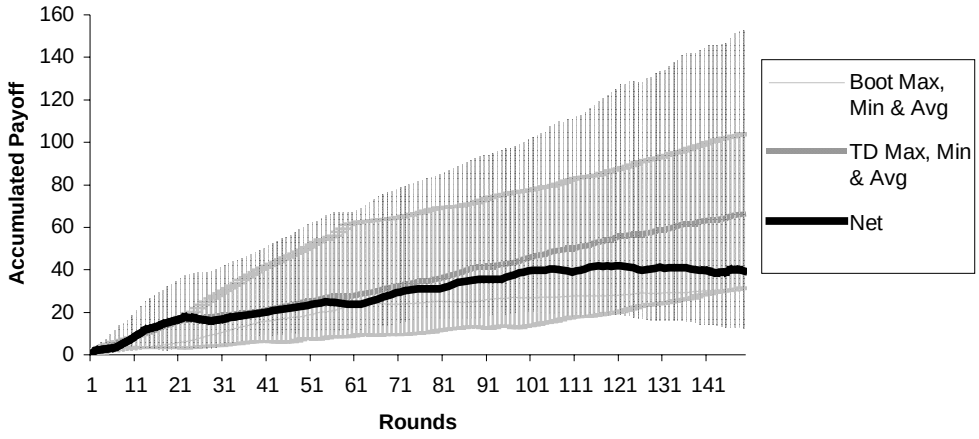



**Figure 6.26 The TD network *versus* the Selective Bootstrap network in the IPD game with selfish play**
The graph shows the average accumulated payoff, the range, i.e., the minimum and maximum values, shown as the shaded area for both the Temporal Difference network and the Selective Bootstrap network competing in an IPD game with selfish play, i.e., maximize their individual rewards. The game was played to 1000 rounds, but only the first 150 rounds are shown as the disparity in the networks' payoff continues in same way, i.e., the difference in the minimum and maximum payoffs continues to increase for both networks.

This disparity is also reflected in the patterns of play in the three trials. A breakdown of which is shown in Figure 6.27.

| Play | % trial 1 | % trial 2 | % trial 3 |
|---|---|---|---|
| **CC** | 12 | 3 | 2 |
| **CD** | 44 | 57 | 39 |
| **DC** | 11 | 26 | 37 |
| **DD** | 33 | 14 | 22 |

**Figure 6.27 Pattern of play for an IPD game where the players receive individual rewards**
Breakdown in percentage by trial of a certain type of play for the IPD game, for example, *DC* with a value of 11 says that 11% of the time the ANNs' played a game of one ANN defecting and the other ANN cooperating.

#### 6.5.1.5 Conclusion

The results showed that the TD Feed Forward network consistently achieved a higher accumulated payoff. However, the ANNs did not follow any specific pattern of behaviour, i.e., both ANNs may choose to defect at every round or one ANN may play a round of cooperation followed by a round of defection. Both ANNs' had a tendency to defect, as shown in the breakdown of the pattern of play in Figure 6.27.

### 6.5.2 The Temporal Difference Network *versus* the Selective Bootstrap Network in an IPD game with global reward

#### 6.5.2.1 Introduction

In the previous experiment where each ANN receives a different reward, we could say that the ANNs played selfishly, as each ANN played to maximize its own payoff. Although, the ANNs did not follow any specific pattern of behaviour, the results suggest both ANNs had a tendency to defect as opposed to cooperate. The motivation for this next set of experiments is: firstly to reduce the variability of the ANNs' behaviour with the aim of improving the net payoff of the organism by rewarding (or penalizing) the ANNs in the same way and secondly to investigate further the idea proposed by Brown and Rachlin (1999) that there is a direct relationship between cooperating with one's self and self-control behaviour. These ideas were presented in Chapter 2 Section 2.3 and discussed in the previous experiment Section 6.5.1.1. With reference to Figure 2.1, Brown and Rachlin (1999) concluded that choosing the reward for mutual cooperation (*CC*), i.e., the top left hand box in the IPD payoff matrix, leads us to the *LL* in the self-control problem in Figure 2.1. In

this experiment the ANNs receive the same global reward with the aim of maximizing the reward for the organism as a whole, where the 2-ANNs model represents the organism. This is measured by the number of times the players play a game of mutual cooperation.

Again the Selective Bootstrap network represents the lower brain system and the Temporal Difference network represents the higher brain functions such as planning and control. Both the Selective Bootstrap rule and the Temporal Difference rule are implemented as described as in the previous experiment. The Value Function for the TD network is shown in Figure 6.28. The Value Function is the probability of winning, given the current state of the environment and the ANN's action. In this experiment, the highest long-term reward is achieved if both ANNs learn to cooperate. This is reflected in the look-up table, which is in effect the Value Function. Figure 6.28 shows the initial values for the Value Function for each state/action combination.

| State D/C | Action D/C | Value Function $V(S_t)$ |
|---|---|---|
| D | D | 0 |
| C | C | 1 |
| D | C | 0.5 |
| C | D | 0.5 |

**Figure 6.28 The look-up table for Temporal Difference learning in the IPD Game with global rewards**
A table of initial values for the learned Value Function *V*, where *D* stands for Defect and *C* for Cooperate. The initial values are based on the fact that to gain the higher long-term reward both ANNs must cooperate.

#### 6.5.2.2 Methodology

The ANN is configured in the same way as in the previous experiment with two inputs nodes to represent the opponent's previous action (a node to represent defection and a node to represent cooperation), and two output

nodes representing a response (a node to represent defection and a node to represent cooperation). The input and output are normalized as in the previous experiment. Figure 6.29 shows the system configuration for this experiment. The system configuration is similar to the Figure 6.22 for the above experiment. In the case of this experiment, the *critic* assigns a global reward or penalty based on the actions of both ANNs, as opposed to assigning an individual reward to each ANN, as in the previous experiment.

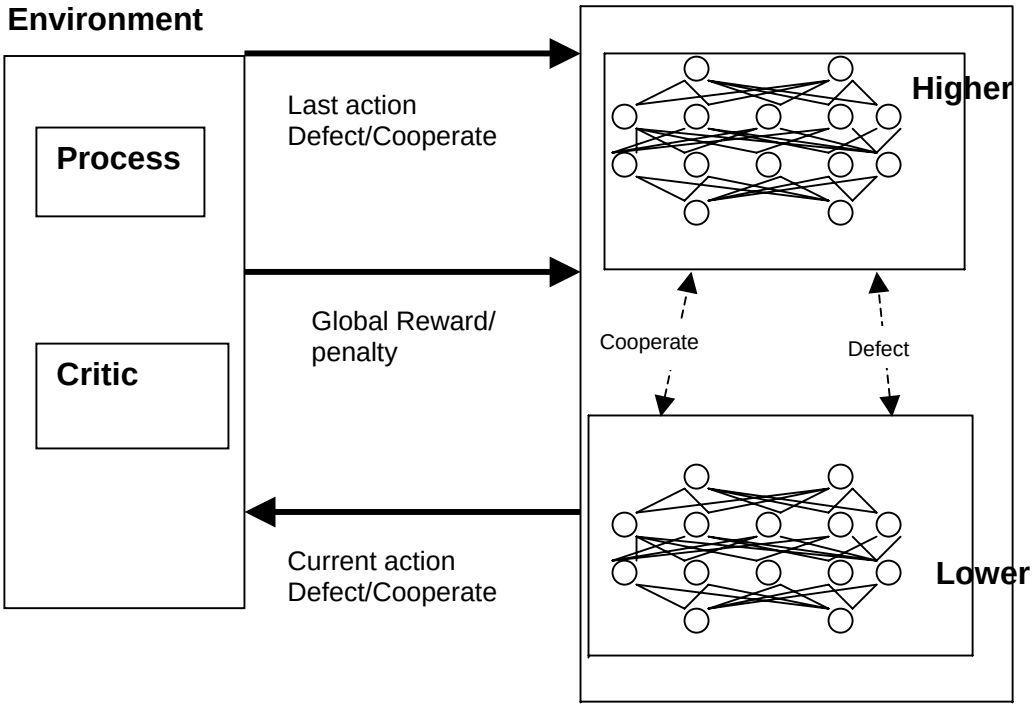



**Figure 6.29 System Configuration for two ANNs playing an IPD with a global reward**
The environment contains a *process* that initializes the input/state to the opponent's previous action (defect or cooperate) at the start of each round. The Output/Action is the network's action (defect or cooperate). The environment also contains a *critic* that assigns a global reward or penalty based on the payoff for each round.

The goal in the previous experiment was to maximize the individual ANN's payoff. As shown in the results in the above experiment, this did not maximize the net payoff of the organism as a whole. The goal in this experiment is to maximise the net payoff. The payoff matrix to be used in this

experiment is shown in Figure 6.30. The reward in this experiment is global, i.e., both ANNs receives the same reward, which is the payoff shown in the matrix in Figure 6.30[6]. The payoff matrix is explained in terms of higher and lower brain regions rather than row and column players. The reward is a global reward to both ANNs, i.e., both ANNs get the same. The reward is the sum of the rewards from the payoff matrix for the IPD game with local rewards shown in Figure 6.22.

| | **Lower** | **Lower** |
|---|---|---|
| **Higher** | 2 (C,C) | 1(C,D) |
| **Higher** | 1(D,C) | 0 (D,D) |

**Figure 6.30 The payoff matrix for the IPD game with global rewards**
The payoff matrix is explained in terms of higher and lower brain regions rather than row and column players. The reward is a global reward to both ANNs, i.e., both ANNs get the same. The reward is the net of the individual rewards from the payoff matrix for the IPD game with local reward Figure 6.22.

Although the payoff matrix as shown in Figure 6.30 violates the first rule of the IPD game, i.e., T>R>P>S, it is similar to the payoff matrix used by Brown and Rachlin (1999) in the self-control game discussed in Section 2.3, whose results we aim to emulate. The ANNs actions are rewarded at each round of the game with the global reward shown in Figure 6.30. The reward for mutual cooperation (*C*,*C*) is the highest at 2, as this is the desired behaviour, the punishment for mutual defection (*D,D*) is the lowest at 0, the penalty for Sucker's payoff (*C,D*) and the reward for Temptation to defect (*D,C*) both have a numerical value of 1. The reward for mutual defection is the lowest, representative of the cost of taking the smaller-sooner reward (*SS*) in the self-

[6] Note that the payoff matrix in the self-control game by Brown and Rachlin (1999) also used global rewards as well and also violates the first rule of the IPD game, i.e., T>R>P>S, refer to Figure 2.5

control problem illustrated in Figure 2.1 in Chapter 2. Mutual cooperation (*C*,*C*) yields the highest reward in the long-term, representative of the larger-later reward (*LL*) in the self-control problem illustrated in Figure 2.1 in Chapter 2 and hence the highest payoff. To illustrate consider our example of the student and the pub, the reward for Mutual defection (*D,D*) reflects the punishment for taking the immediate reward (*SS*) of going to the pub and having a good time, but at a cost of not studying and hence leading to poor grades in the future. Whereas the reward for Mutual cooperation (*C,C*) rewards us for staying at home and studying, leading us to the larger later reward (*LL*) of good grades

#### 6.5.2.3 Test Procedure

The ANNs are configured in the same way as in the previous experiment, i.e., the networks are configured as 2-6-6-2 and all parameters are held at 0.1. The bias was implemented as in the previous experiments, i.e., as a node whose weight is trainable in the same way as the other nodes in the ANN. The task of learning for this experiment is to find the best response based on the opponent's previous response; this is encoded in the hidden nodes. The best response in this case is to maximize the net payoff for the organism as a whole. Again, the pattern of play, i.e., the sequence of the ANN's actions (i.e., to defect or to cooperate), the payoff for the round and the accumulated payoff for the game were recorded. The number of rounds per game was again held at 1000. Again, a trial consists of three games of 1000 rounds. To avoid any first player advantage or disadvantage, the starting ANN is selected at random. The ANNs are rewarded or penalised at the end of each round. Since

the payoff will be the same for both ANNs, the networks are not rewarded or penalized at the end of the game.

#### 6.5.2.4 Results

A trial is repeated 3 times and the accumulated payoff and the average payoff for the three games are recorded, (the payoff in this experiment is the same for both ANNs, as both networks receive the same reward). Figure 6.31 shows the net payoff and the range, i.e., the minimum and maximum payoff for each round is also shown.

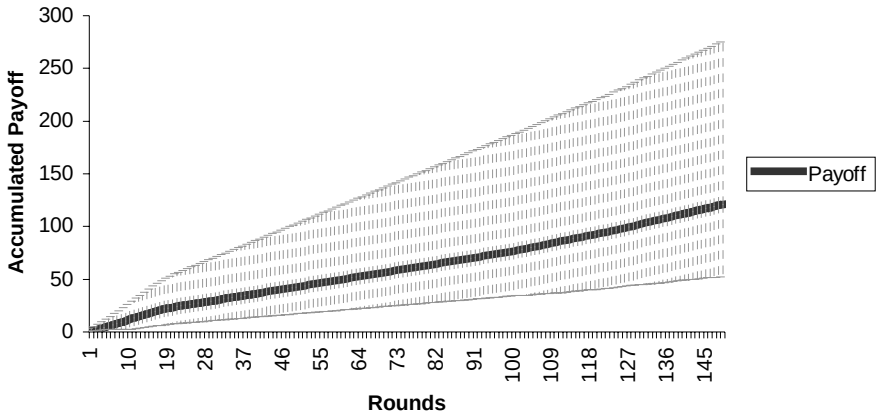



**Figure 6.31 Results for the TD Network and Selective Bootstrap Network playing the IPD game where both networks receive the same reward**
The accumulated payoff for the IPD game with a global reward, showing the range, i.e., the minimum and maximum values as the shaded area.

Figure 6.31 shows the average payoff and range (minimum and maximum values) for the first 150 rounds, as even though each game was played for 1000 rounds play continued in the same way, that is the payoff continued to increase and play tended to be cooperation. This is reflected in the patterns of play in the three trials as shown in Figure 6.32.

| Play | % trial 1 | % trial 2 | % trial 3 |
|---|---|---|---|
| **CC** | 28 | 50 | 55 |
| **CD** | 45 | 1 | 32 |
| **DC** | - | 39 | 12 |
| **DD** | 27 | 9 | 1 |

**Figure 6.32 Pattern of Play for an IPD Game with global reward**
Breakdown in percentage by trial of a certain type of play where players receive the same reward. To illustrate, for *CC* a value of 28 says that 28% of the time the networks played a game where both networks cooperate.

#### 6.5.2.5 Conclusion

In the case of the IPD game played with a global reward, i.e., both ANNs receive the same reward, play was symmetric with cooperation the dominant behaviour from both ANNs. The results show that the accumulated payoff is higher for both ANNs than in the previous experiment, suggesting that the ANNs performed better in terms of desired behaviour, i.e., a global reward promotes mutual cooperation. In summary, with this arrangement (both ANNs receiving the same reward) there is less variability in the ANNs' behaviour and the accumulated payoff increases, as opposed to the net payoff in the IPD with a local reward (Figure 6.26). This can be explained as follows: in the IPD game with a global reward there is a tendency for both ANNs to cooperate, hence the ANNs receive the higher reward for Mutual cooperation. In games of asymmetric play, i.e., one ANN defects at random and the other cooperates, the reward is less and hence the net payoff is less for the organism.

### 6.6 Modelling a bias towards future rewards

The most important purpose of this chapter is to model the behaviour self-control through precommitment. As discussed in Chapter 2, precommitment behaviour can be defined as carrying out an action with the aim of denying (or at least restricting) future temptations, e.g., the *SS* in Figure 2.1, or to go to

the pub in the student example. Precommitment is carried out in order that we can obtain the larger later reward, the *LL* in Figure 2.1, or the good grades in the student example. Precommitment in the case of the student who has to study and wants to avoid temptation such as going to the pub, could be simply switching their mobile phone off or rigging up some contraption to prevent him/her from answering the door. The results from the previous experiment would seem to suggest that a tendency to cooperate leads to the larger later reward, the *LL*, which is in line with the results of Brown and Rachlin (1999). Cooperation was enhanced to some extent by implementing a global reward. It would seem however from the results of the previous experiment that in order to maximize the long term payoff, we need to bias rewards to promote cooperation. Secondly, in the payoff matrix in Figure 6.30, we do not differentiate between the (*C,D*) situation, which represents the middling behaviour of staying at home, but not studying as effectively, because you wish you had gone to the pub and the (*D,C*) situation, i.e., the negative behaviour of going to the pub, but having a miserable time because you feel guilty for not studying.

In this next set of experiments we examine three ways of addressing these issues. The first experiment investigates the effect of implementing the bias towards future rewards by simply varying the input value of the ANN's bias node between 0 and 1 (instead of fixed as 1). The effect on the behaviour of the 2-ANNs model is recorded. This method is referred to as the *variable bias* method. The second experiment implements a bias towards future rewards as an extra input to one or both of the ANNs in the 2-ANNs model. In this case

the ANN's threshold is implemented in the usual way, i.e., as a node with an input value of 1 whose weight is trainable in the same way as the other nodes in the network and the bias towards future rewards is implemented as an additional node to the input layer. Different values are tried for this extra input and the effect is recorded. This method is referred to as the *extra input bias* method. The final experiment implements a bias towards future rewards as a differential bias applied to the payoff matrix of Figure 6.30 to calculate a differential payoff. Again the ANN's threshold is implemented in the usual way, i.e., as a node with an input value of 1 whose weight is trainable in the same way as the other nodes in the network. This method is referred to as the *differential bias* method. In all three techniques for implementing the bias towards future rewards the parameter is assigned a value between 0 and 1, which is fixed for the duration of the trial. The results for all three techniques are recorded and compared.

### 6.6.1 Modelling bias towards future rewards as a *variable bias*

#### 6.6.1.1 Introduction

In this experiment a bias towards future rewards is implemented by varying the value of the bias node of the ANN between the values 0 and 1, i.e., this variable bias is in effect the ANN's bias. These different values represents different values of bias towards future rewards. For example switching off his or her mobile phone biased the student's later choice of staying home and studying. A more extreme form would be to rig up some contraption, e.g., nailing planks of wood to the door, to physically prevent the student from going to the pub. A version of IPD was played with the TD network

competing against the Selective Bootstrap network. The *variable bias* value is fixed for the duration of the game. The aim of this experiment is to reduce behaviour variability of the ANNs by increasing the value of this *variable bias* and also to enhance cooperation. This is the desired behaviour if this *variable bias* technique did indeed represent precommitment.

#### 6.6.1.2 Methodology

Each ANN is configured as shown in Figure 6.33.

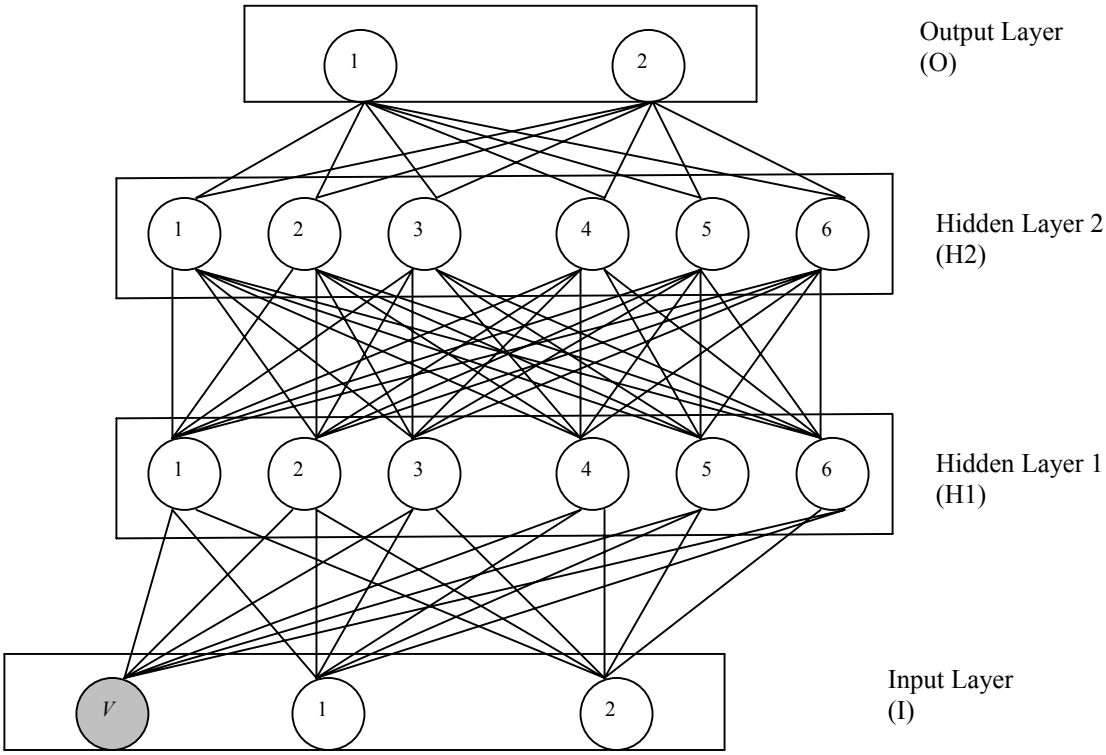



**Figure 6.33 The network topology for an ANN implemented with a variable bias**
The network topology used for the TD network and the Selective Bootstrap network where the bias towards future rewards is implemented by varying the input value of the network's bias. This technique is referred to as the *variable bias V*.

There are two inputs nodes to represent the opponent's previous action (a node to represent defection and a node to represent cooperation), a *variable bias* whose input takes a value between zero and 1 and whose weight is

trainable in the same way as the other nodes in the network and two output nodes representing a response (a node to represent defection and a node to represent cooperation). The input and output are normalized as in the previous experiments. The system configuration for this experiment is the same as in the experiment where the ANNs in the 2-ANNs play to maximize their own payoffs, which assigns a local reward or penalty to each ANN, see Figure 6.20. The payoff matrix is the same as the experiment where each ANN receives a local reward as shown in Figure 6.22.

#### 6.6.1.3 Test Procedure

The ANNs are configured as 2-6-6-2 and all learning parameters are held at 0.1 as in the experiment in Section 6.5.2. A series of experiments were run to test the effect of increasing the *variable bias*. Again, the pattern of play, i.e., the sequence of the ANN's action, i.e., to defect or to cooperate, the payoff for the round and the accumulated payoff for the game were recorded. The number of rounds per game was held at 1000, giving the ANNs a chance to learn. A trial consists of three games. To avoid any first player advantage or disadvantage, the starting ANN is selected at random. The ANNs are rewarded or penalised just at the end of each round.

#### 6.6.1.4 Results

A series of experiments were run to test the effect of changing the variable bias on the ANNs' emergent behaviour when both ANNs signal a bias towards future rewards. Figure 6.34 summarizes the results for a range of values for the *variable bias V*.

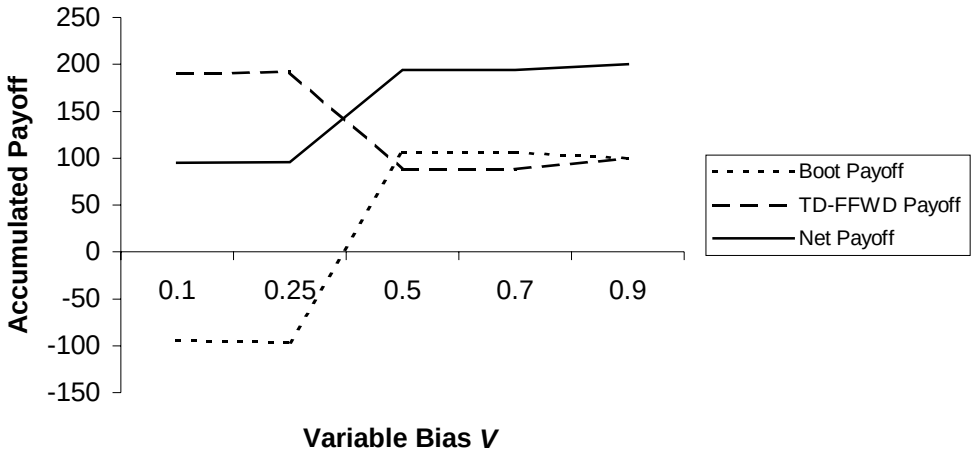



**Figure 6.34 TD network *versus* Selective Bootstrap network with a *variable bias* of the same value competing in the IPD game**
The accumulated payoff for a Temporal Difference Feed Forward Network competing against a Selective Bootstrap Network in the Iterated Prisoner's Dilemma's game. Both networks have a *variable bias V* with the same value for the bias towards future rewards. The net payoff (the summation of the accumulated payoffs from both networks) increases, as cooperation becomes the dominant play from both networks, since both networks are rewarded for mutual cooperation.

The results show that increasing the value of the variable bias node increases the tendency, from both ANNs, is to cooperate. This is shown in the increase in the net payoff, as mutual cooperation (1,1) produces the highest payoff in the long term. For low values of the variable bias node, the play tends to be asymmetric, reflected in the disparity in the payoffs between the two ANNs. In an asymmetric play one ANN will defect and receive the higher immediate reward of Temptation to defect of 2; the other ANN will cooperate and receive the lower reward of Sucker's payoff of (-1). This is reflected in the vastly different accumulated payoffs for the ANNs. Increasing the value of the value of the variable bias node promotes cooperation from both ANNs and the difference is reduced.

### 6.6.1.5 Conclusion

In this section the bias towards future rewards is modelled as a variable bias whose value ranged from zero, indicating that the network is not biasing towards future rewards, to one where the network is fully committed to long term rewards. Theoretically this could not act as precommitment to the network, because during training there is a possibility that the final values of the weights attained are such as to cancel out the different input value for the bias towards future rewards, as shown in Figure 6.35.

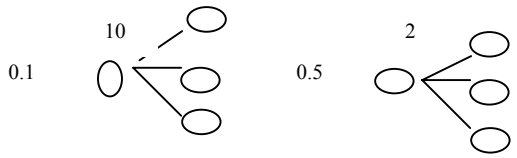



**Figure 6.35 A problem with implementing a bias towards future rewards as a *variable bias***
The effective input for of the above configurations are the same (i.e., 0.1 * 10 = 0.5 * 2), so they have the same effect despite the different initial input values.

So what is the effect of a *variable bias* on the ANN? It would seem from the results in this experiment that with a *variable bias* cooperation is enhanced, which is the desired behaviour if this *variable bias* did indeed represent precommitment. In addition, with the variable bias, the ANN still has the capability to learn the best solution. An alternative method, described in the next experiment, does not have the problem highlighted in Figure 6.35.

### 6.6.2 Modelling a bias towards future rewards as an *extra input* to the ANN

#### 6.6.2.1 Introduction

The motivation for this experiment is to determine if a bias towards future rewards can be modeled as an extra input to the ANN. Modeling a bias towards future rewards as an extra node in the input layer is in effect implementing an extra input to the ANN whose value is fixed. In this experiment, a bias towards future rewards is implemented as an extra node in the input layer with the following assumptions: (i) once announced it must be irrevocable, i.e., once the value is set it cannot be changed for the duration of the trial and (ii) the input value of the extra node has different values to model different levels of bias towards future rewards. For example, moving the alarm clock away from the bed biased our later choice of getting up when the alarm clock rings. A more extreme form of a bias towards future rewards however, would be to rig up some contraption to physically prevent ourselves from going back to sleep. These different levels of a bias towards future rewards are modeled as input values from zero (do not care about future rewards, e.g., the alarm clock stays by the bed), to one (fully committed to long term gain, e.g., the contraption).

#### 6.6.2.2 Methodology

The ANN is configured in the same way as in previous experiments, as shown in Figure 6.23 with the network's bias implemented as a node whose weight is trainable in the same way as the other nodes in the ANN. In this cases bias

towards future rewards is implemented as an extra input node whose value ranged from zero to one as shown in Figure 6.36.

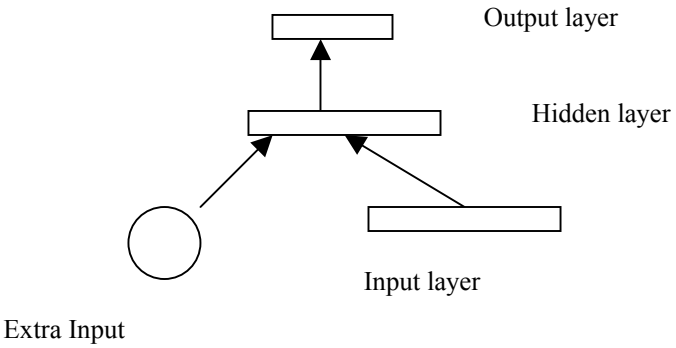



**Figure 6.36 The topology of an ANN with a bias towards future rewards implemented as an *extra input***
A bias towards future rewards is implemented as an additional input node whose value ranged from zero to one.

As in the previous experiment there are two inputs nodes to represent the opponent's previous action (a node to represent defection and a node to represent cooperation), and two output nodes representing a response (a node to represent defection and a node to represent cooperation). The input and output are normalized as in the previous experiment. The system configuration is the same as that shown in Figure 6.28. Again, a global reward or penalty, based on the actions of both ANNs, is assigned to both ANNs at the end of each round. The payoff matrix to be used in this experiment is the same as the global reward as shown in Figure 6.30.

#### 6.6.2.3 Test Procedure

The ANNs are configured as 3-6-6-2 and all learning parameters are held at 0.1. A version of IPD was played with the TD network competing against the Selective Bootstrap network. The extra input value is fixed for the duration of the game. The aim is to reduce behaviour variability from the ANN by

increasing the value of the extra input. A series of experiments were run to test the effect of increasing the bias towards future rewards on the ANNs emergent behaviour when (i) both ANNs signal a bias towards future rewards and (ii) when just one ANN signals a bias towards future rewards. Again, the pattern of play, i.e., the sequence of the ANN's actions to defect or to cooperate, the payoff for the round and the accumulated payoff for the game were recorded. The number of rounds per game was held at 1000, giving the networks a chance to learn. A trial consists of three games. To avoid any first player advantage or disadvantage, the starting ANN is selected at random. The ANNs are rewarded or penalised just at the end of each round.

#### 6.6.2.4 Results

Figure 6.37 summarizes the results when both ANNs signal a bias towards future rewards.

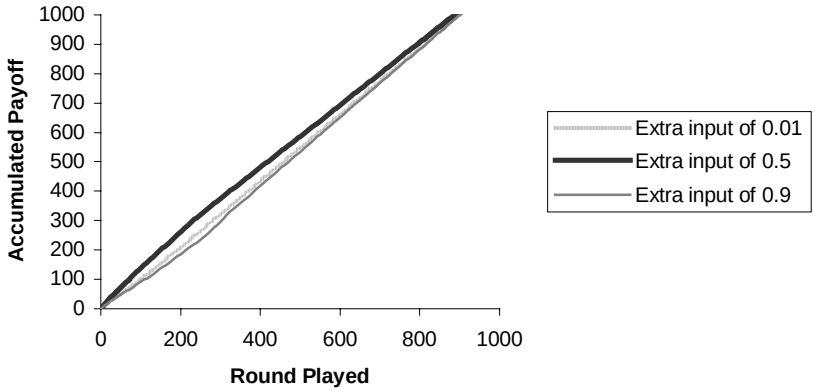



**Figure 6.37 TD network *versus* Selective Bootstrap network in the IPD game with a bias towards future rewards implemented as an extra input on both ANNs**
The net payoff for a Temporal Difference network competing against a Selective Bootstrap network in the Iterated Prisoner's Dilemma's game. Both networks signal bias for future rewards with the same value. The net payoff increases, as cooperation becomes the dominant play from both networks, hence both networks receive the higher reward for mutual cooperation.

The results show that when a bias towards future rewards is implemented on both ANNs, the tendency from both ANNs is to cooperate. This is shown as an increase in the payoff, as mutual cooperation produces the highest payoff in the long term, which is reflected in the patterns of play in Figure 6.38.

| | 0.01 (%) | 0.5 (%) | 0.9 (%) |
|---|---|---|---|
| **CC** | 60 | 36 | 69 |
| **CD** | 6 | 31 | 20 |
| **DC** | 19 | 32 | 5 |
| **DD** | 15 | 1 | 6 |

**Figure 6.38 Pattern of play with a bias towards future rewards implemented as an extra input to both ANNs**
Breakdown in percentage by trial of a certain type of play for the IPD game with global reward, for example, *DC* with a value of 38 says that 38% of the time the networks played a game where the TD network cooperates and the Selective Bootstrap network defects. A pattern of play of *CC* or *DD* is considered to be symmetric. A pattern of play of *CD* or *DC* is considered to be asymmetric.

Figure 6.39 summarizes the results when just the TD network signals a bias towards future rewards.

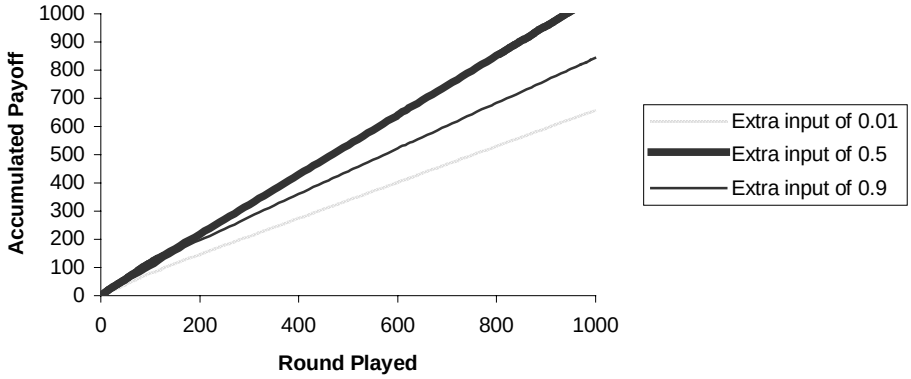



**Figure 6.39 TD Network *versus* Selective Bootstrap Network in the IPD game with a bias towards future rewards implemented as an extra input to only the TD Network**
The payoff increases for all levels, but the value of the bias has an effect on the amount of payoff achieved.

The results show that the level of bias affects the disparity in the payoffs. For low levels of bias, the play tends to be asymmetric, as shown in the breakdown of the typical patterns of play in Figure 6.40.

| | 0.01 (%) | 0.5 (%) | 0.9 (%) |
|---|---|---|---|
| **CC** | 24 | 55 | 35 |
| **CD** | 23 | 18 | 22 |
| **DC** | 38 | 14 | 26 |
| **DD** | 15 | 16 | 19 |

**Figure 6.40 Pattern of play with a bias towards future rewards implemented as an extra input on only the TD Network**
Breakdown in percentage by trial of a certain type of play for the IPD game with global reward when a bias towards future rewards is implemented on just the TD network. To illustrate, an entry *CD* with a value of 23 says that 23% of the time the networks played a game where the TD network cooperates and the Selective Bootstrap network defects. A pattern of play of *CC* or *DD* is considered to be symmetric. A pattern of play of *CD* or *DC* is considered to be asymmetric.

This is reflected in the lower payoff, as in asymmetric play where one ANN will defect and the other ANN will cooperate, the organism will receive the lower reward of Temptation to defect or Sucker's payoff. Increasing the level of bias on the TD Feed Forward network does not necessarily promote cooperation from both networks.

Figure 6.41 summarizes the results when just the Selective Bootstrap network signals a bias for future rewards.

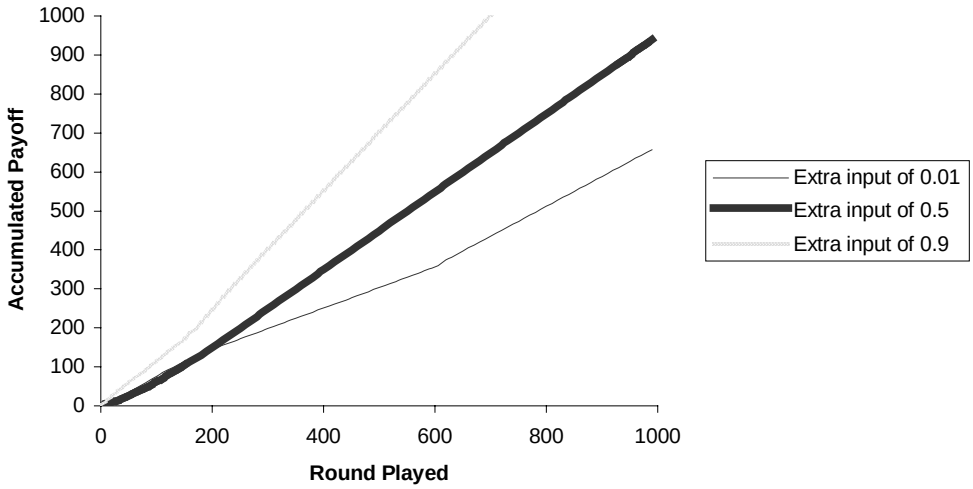



**Figure 6.41 The TD Network *versus* the Selective Bootstrap Network in the IPD game with a bias towards future rewards implemented as an extra input on only the Selective Bootstrap Network**
The net payoff increases for all levels. The value of the bias has a greater effect on the amount of payoff than when signaled on just the TD network or on both networks.

The results show that increasing the level of bias towards future rewards on the Selective Bootstrap network promotes cooperation from both networks, as shown in Figure 6.42 and the payoff increases, as mutual cooperation yields the highest reward. There is a direct relationship between the level of bias for future rewards and cooperation.

| | 0.01 (%) | 0.5 (%) | 0.9 (%) |
|---|---|---|---|
| **CC** | 31 | 32 | 89 |
| **CD** | 11 | 31 | 6 |
| **DC** | 34 | 30 | 2 |
| **DD** | 24 | 7 | 3 |

**Figure 6.42 Pattern of play with a bias towards future rewards implemented as an extra input on only the Selective Bootstrap Network**
Breakdown in percentage by trial of a certain type of play for the IPD game with global reward, with bias towards future rewards implemented on just the Selective Bootstrap network. For example, *CC* with a value of 31 says that 31% of the time the networks played a game where both networks cooperate. A pattern of play of *CC* or *DD* is considered to be symmetric. A pattern of play of *CD* or *DC* is considered to be asymmetric.

#### 6.6.2.5 Conclusion

In this section the bias towards future rewards is modelled as an extra input node whose input value ranged from zero, indicating that the ANN is not biased towards future rewards, to one, where the ANN is fully committed to long term rewards. This technique of modelling the bias towards future rewards as an extra input does seem to promote cooperation, although the percentage of cooperation depends upon which network signals a bias towards future rewards. In addition, this technique does not have the problem illustrated in Figure 6.35. The problem in this technique is by using the global reward payoff matrix of Figure 6.30 the situations of (*C,D*) and (*D,C*) are rewarded in the same way, which is not necessarily true. For example, if we consider the student and the temptation of going to the pub, if it is assumed that *C* is staying at home and *D* is going to the pub, then (*C,D*) could represent the middling situation of when asked to go to the pub you decide to stay at home, but do not feel so much of a conflict, as you were determined to stay at home any way. Therefore you work reasonably well. (*D,C*) could represent the more negative situation of going to the pub, but you feel doubly miserable because not only do you feel guilty about not studying, but you also are going against your long term goal. An alternative technique for modeling a bias towards future rewards, described in the next section, aims to address this.

### 6.6.3 Modelling a bias towards future rewards as a *differential bias* applied to the payoff matrix

#### 6.6.3.1 Introduction

The motivation for this experiment is to examine the effect of modelling a bias towards future long-term rewards as a variable bias applied to the payoff matrix referred to as the *differential bias*. This *differential bias*, with a value between 0 and 1, is assigned to the payoff matrix for both ANNs to calculate the differential payoff and is fixed for the duration of the trial. Let $\psi$ represent the *differential bias* to distinguish it from the Neural Network bias described in Figure 6.23. The *differential bias* $\psi$ is added only to the diagonal terms in the matrix as shown in Figure 6.43:

| | **Lower** | **Lower** |
|---|---|---|
| **Higher** | 2 (C,C) | 1(C,D) +ψ |
| **Higher** | 1(D,C)-ψ | 0 (D,D) |

**Figure 6.43 A bias towards future rewards implemented as a *differential bias* $\psi$ applied to the payoff matrix to calculate the differential payoff**
A differential bias $\psi$ is applied to the payoff matrix. The bias is assigned with a value between 0 and 1 for both networks, which is fixed for the duration of the trial. This bias is used in the payoff matrix, to calculate the differential payoff and is added only to the diagonal terms in the matrix. This represents a bias towards future reward in the following way. In the example of the student and the pub, when the student is asked to go to the pub, but stays at home (*C,D*), he or she does not feel so much of a conflict, as they were determined anyway to stay at home. Therefore, they work reasonably well and get a good payoff. Similarly, if the student is asked to stay at home, but goes to the pub (*D,C*), they feel doubly miserable as they also go against their own preference to the long-term reward.

This represents a bias towards future rewards in the following way. In the example of the student and the pub, when the student is asked to go to the pub, but stays at home (*C,D*), he or she does not feel so much of a conflict, as

they were determined anyway to stay at home. Therefore, they work reasonably well and get a good payoff. Similarly, if they are asked to stay at home, but go out (*D,C*), they feel doubly miserable as they also go against their own bias for long-term future gain.

#### 6.6.3.2 Methodology

The ANN is configured in the same way as in the experiment described in Section 6.5.2 and shown in Figure 6.23 with two inputs nodes to represent the opponent's previous action (a node to represent defection and a node to represent cooperation), and two output nodes representing a response (a node to represent defection and a node to represent cooperation). The ANN's bias was implemented as in the previous experiment, i.e., as a node whose weight is trainable in the same way as the other nodes in the network. The input and output are normalized as in the previous experiments. The system configuration for this experiment is the same as the previous experiments, which assigns a global reward or penalty based on the actions of both ANNs, i.e., Figure 6.29. The payoff matrix to be used in this experiment is shown in Figure 6.44. The differential bias $\psi$ is assigned a value between 0 and 1 at the beginning of a trial for both networks, and is fixed for the duration of that trial.

#### 6.6.3.3 Test Procedure

The ANNs are configured as 2-6-6-2 and all learning parameters are held at 0.1 as in the experiment in Section 6.5.2. A series of experiments were run to test the effect of increasing the differential bias, when applied to the payoff matrix, on the emergent behaviour of the ANNs. Again, the pattern of play,

i.e., the sequence of the ANN's actions to defect or to cooperate, the payoff for the round and the accumulated payoff for the game were recorded. The number of rounds per game was held at 1000, giving the ANNs a chance to learn. A trial consists of three games. To avoid any first player advantage or disadvantage, the starting ANN is selected at random. The ANNs are rewarded or penalised just at the end of each round.

#### 6.6.3.4 Results

Figure 6.45 compares the effect on the accumulated payoff of increasing the value of the *differential bias* $\psi$.

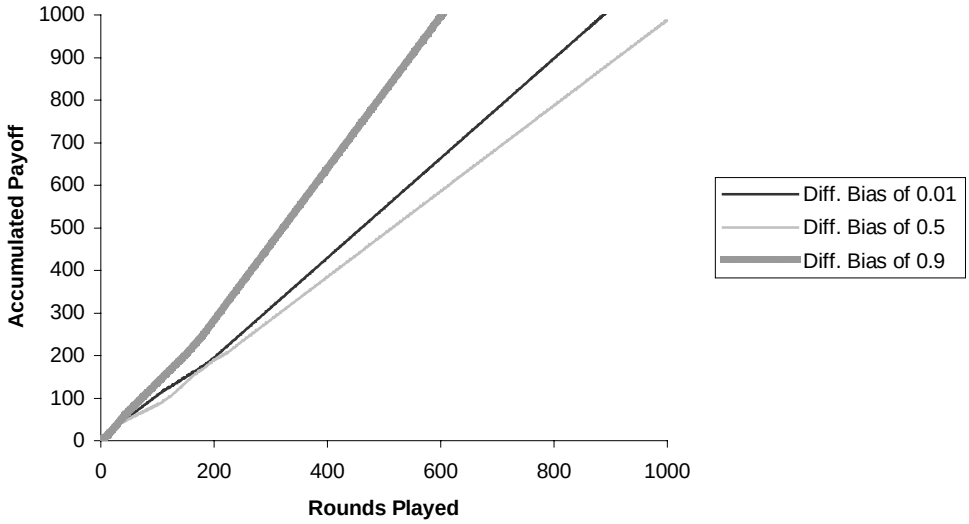



**Figure 6.44 The effect of increasing the *differential bias* when added to the diagonal rewards of the payoff matrix in the IPD game**
When a differential bias $\psi$ is applied to the diagonal rewards for asymmetric play, i.e., (*C,D*) or (*D,C*), the results suggest that increasing $\psi$ promotes cooperation behaviour leading to the reward for mutual Cooperation.

The results show that increasing the level of bias $\psi$, implemented as described above, would seem to promote cooperation, as shown in Figure 6.45, and hence the accumulated payoff increases, as the ANNs receive the

higher reward for mutual cooperation. In addition, it would seem from the results in Figure 6.46, that implementing the bias for future rewards as a differential bias $\psi$ addresses the conflict represented by the situations of (*C,D*), i.e., when asked to the pub you stay at home, but do not study as effectively because you wish you had gone to the pub, and (*D,C*) going to the pub, but having a miserable time because you feel guilty about not studying. It does this by promoting the middling behaviour of when you are asked to go to the pub, you stay at home (*C,D*), but you do not feel so much of a conflict, as you were determined anyway to stay at home.

| | **0.01 (%)** | **0.5 (%)** | **0.9 (%)** |
|---|---|---|---|
| **CC** | 55 | 33 | 75 |
| **CD** | 6 | 0 | 24 |
| **DC** | 34 | 52 | 1 |
| **DD** | 79 | 15 | 0 |

**Figure 6.45 Pattern of play when the bias for future reward is implemented as a *differential bias* applied to the payoff matrix**
Breakdown in percentage by trial of a certain type of play for the IPD game with bias for future rewards implemented as a *differential bias* applied to the payoff matrix. For example, *CC* with a value of 61 says that 61% of the time the networks played a game where both networks cooperate. A pattern of play of *CC* or *DD* is considered to be symmetric. A pattern of play of *CD* or *DC* is considered to be asymmetric.

#### 6.6.3.5 Conclusion

Implementing a bias towards future rewards as a bias applied to the payoff matrix, to calculate the differential payoff, would seem to not only to promote cooperation, but also to address the internal conflict represented by either the (*C,D*) or (*D,C*) situation. In particular, the middling behaviour of staying at home and working (*C,D*) would appear to increase as the differential bias $\psi$ approaches the upper limit of the range tested (0.9). This can perhaps be

explained as follows: as the reward is global, i.e., both ANNs receive the same reward, implementing the bias towards future rewards in this way affects both ANNs, hence increasing the differential bias $\psi$ increases the reward for the middling behaviour of (*C,D*) bringing it closer to the reward for mutual cooperation, whilst at the same time decreasing the reward for the more negative behaviour of (*D,C*) bringing it closer to the reward for mutual defection. The result is that instead of four classes of rewards the organism is faced with just two classes of rewards: one with a tendency for cooperation and one with a tendency for defection.

## 6.7 Summary

In this chapter the 2-ANNs model presented in Chapter 3, representing the higher and lower brain centres, is implemented as two artificial neural networks with different weight update rules to represent the different behaviours of the higher and lower centres of the brain. We tested the 2-ANNs model as two autonomous players learning simultaneously in a shared environment competing in two general-sum games, firstly in the RBG then in the IPD game. This makes our 2-ANNs a multi-agent system. We have seen from the review of MARL in Section 6.2 that this is an area still in its infancy. There is still no definitive model of multi-agent learning, as there is with SARL and the MDP mathematical model. MARL has been proven to work for a restricted set of general-sum games, which are strictly competitive, i.e., zero-sum games. In this chapter, we have shown that convergence is reached in both the RBG and the IPD game, which are general-sum games where the players' payoffs are neither totally positively nor totally negatively correlated.

The networks weights settled into an equilibrium and with the performance of the ANNs reaching an acceptable level, i.e., division of the resource is close to half in the RBG, and both ANNs settle into a play of mutual cooperation in the IPD game. These results suggest that TD learning is representative of the higher brain processes and the Selective Bootstrap is representative of the lower brain processes. Since the Selective Bootstrap network accepted less than optimal offers, i.e., less than half of the pot in the RBG, and had a tendency to defect in the IPD game, which is indicative of myopic behaviour associated with the lower brain processes. The Temporal Difference network exhibited behaviour associated with the higher brain functions such as planning and control in that it did not accept the first offer made and appeared to hold out for a more acceptable offer in the RBG, and had a tendency to cooperate in the IPD game.

In the final set of experiments, a version of IPD was played with the TD network competing against the Selective Bootstrap network, with one or both networks signalling a bias towards future rewards. The results suggest that with a bias towards future rewards implemented as a *variable bias* cooperation is enhanced. This is the desired behaviour, if this *variable bias* technique does indeed represent precommitment, as increasing this *variable bias* enhances cooperation behaviour leading to the *LL.* With this technique there is a possibility that the final values of the weights are such as to cancel out the effect of the *variable bias*, as illustrated in Figure 6.35. This problem does not occur when a bias towards future rewards is implemented as an *extra input* node and again cooperation is enhanced. With this *extra input* technique

the situations represented by (*C,D*) and (*D,C*) are rewarded in the same way, which is not necessarily true. When the bias towards future rewards was implemented as a *differential bias* added to the global reward in the payoff matrix, cooperation behaviour was further enhanced. In addition, by increasing this *differential bias* the dilemma represented by the situations (*C,D*) or (*D,C*) is resolved since the reward for the middling behaviour represented by (*C,D*) also increases promoting cooperation. For these reasons it was considered that the *differential bias* technique was the best technique to model precommitment. This apparent relationship, between precommitment modeled, as a differential bias, and cooperation, is explored further in the context of evolution in the next chapter.

# Chapter 7

# 7 Evolutionary Adaptation of the Neural Model

## 7.1 Chapter Outline

In this chapter the 2-ANNs model, presented in Chapter 3, and developed and tested in the previous chapter, undergoes evolutionary adaptation by simulating genetic evolution using genetic algorithms (Holland, 1992). In Chapter 5, it was concluded that genetic algorithms is the evolutionary algorithm of choice in this thesis since, they are concerned with the evolution of the individual and use a near true simulation of natural evolution making it more biologically plausible than alternative evolutionary algorithms such as Evolutionary Programming or Evolutionary Strategies. In addition, in Genetic Algorithms information on intermediate generations is easily retained and since the aim of simulating an evolutionary process in this thesis is not simply to determine a clear winner, but to examine what behaviour patterns emerge this is yet another reason why Genetic Algorithms is the evolutionary algorithm of choice for this thesis.

The motivation for this chapter is to investigate the evolution of the behaviour self-control through precommitment. The results in Chapter 6 suggest implementing a bias towards future rewards enhances cooperation behaviour, which as Brown and Rachlin (1999) suggest leads to greater self-control as we learn to cooperate with our selves. In Chapter 6, this bias towards future rewards was called the *differential bias* to distinguish it from the ANN bias shown in Figure 6.23. If this differential bias does in fact enhance cooperation

behaviour then this could be interpreted as precommitment as it has the same effect as precommitment, i.e., cooperation behaviour is enhanced leading to the larger later reward, the *LL* in Figure 2.1. In addition, the results in Chapter 6, Section 6.6.2, showed that increasing this differential bias appeared to resolve some internal conflict represented by the (*C,D*) or (*D,C*) situations, again suggesting that this is a reliable technique of modeling precommitment since, as described in Section 2.2, precommitment resolves some internal conflict by restricting or denying future choices. The evolution of this differential bias as precommitment, is the purpose of this chapter.

There are two simulations of evolutionary adaptation carried out on the 2-ANNs model developed and tested in Chapter 6. Both simulations focus (as before) on the functional decomposition of the brain into "higher" brain functions associated with rational thought, and the "lower" brain functions associated with instinctive behaviour as in our 2-ANNs model. The two Artificial Neural Networks (2-ANNs) model in this thesis is subjected to evolutionary adaptation using GAs.

The first simulation investigates the suggestion made in Chapter 6 that self-control is an example of some internal conflict that is resolved by precommitment. It does this by examining what behaviour patterns emerge from the 2-ANNs model, focusing on when inconsistent patterns of behaviour occur and what (if any) is the dominant behaviour from each of the two ANNs. The types of questions that are asked are: what is the effect of this differential bias on the pattern of play? What different patterns of behaviour

emerge, for example, when is cooperation the dominant pattern of play? What are the results, in terms of fitness, of the various possible combinations of patterns of play? Does an ANN prefer a particular pattern of play? Is any one network the decision-maker? In addition, the simulation investigates if precommitment, implemented as a bias towards future rewards, results in a fitness benefit in game-theoretical situations. The types of questions that are asked are: what is the effect of this bias towards future rewards on the fitness of the individual, on the population and on future generations? What values of this bias towards future rewards work best, i.e., maximize the fitness of the individual, the population and future generations? The simulation answers these questions by examining the effect of this differential bias on the payoff, where the payoff represents the fitness of the organism. The premise being that if this differential bias is a successful mechanism in game-theoretical situations then this should result in a higher fitness for the organism. The relationship between this differential bias and cooperation is investigated in the context of the psychological data on self-control and cooperation by Brown and Rachlin (1999).

The second simulation examines the effect of this differential bias on learning. The premise being that the brain is not hard-wired for every response and learning during an organism's lifetime plays a critical part in deciding which action is the best response to a changing environment. Here the role of learning and the effect of learning on the fitness of the organism in the context of this differential bias undergoing evolutionary adaptation is investigated. The type of questions that are asked here are: what is the effect of learning on

the fitness of the individual, on the population and on future generations? What effect do different values for this bias towards future rewards have on learning? For example, does reducing the differential bias slow down learning? Does a differential bias of zero prevent learning? If the differential bias is removed, what is the effect on learning? Does having a bias towards future rewards make a difference on the results?

## 7.2 Scenario of Simulation of the evolution of a bias towards future rewards

The motivation in simulating the evolutionary process in this first simulation is not to simply determine a clear winner, but to investigate how a bias towards future rewards leads to a larger, but later reward, the *LL* in Figure 2.1, and how this bias may have evolved. In the first instance, experiments are run to see which values for this bias towards future rewards yield the higher payoff. The payoff in this experiment is the individual's fitness. The results from the experiments in Chapter 6 on modeling a bias towards future rewards, suggested that a high payoff indicates a higher percentage of cooperation behaviour. In this simulation the effect of increasing this bias towards future rewards on cooperation behaviour and hence payoff is investigated and the results are compared to Baker's results (2001), which showed that increasing the probability of reciprocation, increases the tendency to cooperate. The hypothesis, which is made in this thesis, is that this bias towards future rewards plays the same role as the probability of reciprocation in Baker's experiment (2001) in that increasing the value of this bias towards future rewards promotes cooperation behaviour.

The evolutionary ideas are implemented in a game theoretical context such as in Axelrod's evolution of cooperation (Axelrod and Hamilton, 1981), where different strategies were represented as a string of chromosomes for the Iterated Prisoner's Dilemma (IPD) game. In the Axelrod and Hamilton (1981) experiment each individual represented a strategy. Each individual in the current generation using the strategy defined in its chromosomes, competes against other strategies in the Iterated Prisoner's Dilemma game. Each game consisted of 151 rounds against the same opponent. This was the average number of rounds used in an earlier experiment of a computer tournament of different strategies submitted by academics in economics, sociology, political science and mathematics (Axelrod and Hamilton, 1981). The fitness of an individual was defined as the aggregate total against all the opponents, in all the competitions. Each strategy was ranked by its fitness. The fittest strategies were subjected to genetic evolution using genetic algorithm techniques (Holland, 1992). Axelrod characterised the strategies that evolved as either *Nice* or *Nasty*. A *Nice* strategy, is the one where the player is never the first to defect, but is capable of defecting (only in retaliation). A *Nasty* strategy includes all other strategies. A player with a Nasty strategy will defect even when not provoked. Axelrod and Hamilton (1981) found that most of the strategies that evolved were *Nice* strategies, such as the Tit-for-Tat strategy. The Tit-for-Tat strategy is where the player cooperates on the first move and from then on he or she does whatever the other player did on the previous move (Axelrod and Hamilton, 1981). In Axelrod's study on the simulation of the evolution of strategies for the IPD, it was found that the strategy Tit-for-Tat was found to achieve the highest score

and that most of the strategies that evolved in the simulation resembled Tit-for-Tat (Axelrod and Hamilton, 1981).

Rachlin (2000) suggests that the problem of self-control has been likened to an IPD game of Tit-for-Tat with oneself. The self-control problem can be constructed as an IPD game in the following way: defection (going to the pub and socializing), yields a higher immediate payoff, i.e., the *SS* in Figure 2.1, but if you continue to defect however, you would do worse in the long term rather than if you cooperate (stay at home and study), leading to the larger later reward (good grades), i.e., the *LL* in Figure 2.1. In terms of self-control this can viewed as learning to cooperate with oneself in order to maximise the total accumulated payoff. Brown and Rachlin (1999) suggest that self-control can be explained in terms of the probability of continuing to cooperate with oneself. If one cooperates with his or her self now (stays at home and studies), then the next time he or she faces the dilemma of choosing between the smaller-sooner reward (the pub) or the larger-later reward (good grades), he or she can still choose the higher future reward, i.e., the good grades. In this simulation the relationship between cooperation and implementing a bias towards future rewards is investigated in the context of evolutionary adaptation of the 2-ANNs model. In particular the focus is on the effect of increasing this bias for future rewards on behaviour variability and fitness.

### 7.2.1 Architecture and Algorithm

The simulation program is implemented on the 2-ANNs model developed and tested in Chapter 6, playing the Iterated Prisoner's Dilemma (IPD) game. As in the experiment in Section 6.6.2, the ANN simulating the lower brain

functions is implemented with the Selective Bootstrap weight update rule (Widrow et al., 1973) and the ANN simulating the higher brain functions with the Temporal Difference weight update rule (Sutton, 1988). This is justified since, the Selective Bootstrap network fared worse in both the RBG and the IPD game exhibiting behaviour generally associated with the lower brain processes, such as in the RBG accepting the first offer made. The Temporal Difference network did not necessarily accept the first offer made and achieved the higher payoff in the IPD game, with behaviour associated with the higher brain functions such as planning and control.

Most of the research in the combination of Evolutionary Algorithms and Artificial Neural Networks is concerned with finding the optimum ANN for a specific problem (Yao, 1999). In this chapter a break from the traditional approach is adopted. The techniques of GAs, RL and ANNs are combined in a novel approach in order to investigate the role of evolution in the development of a bias towards future rewards. To summarize, Genetic Algorithms (GA), work on a *populatio*n of individuals. Each individual is represented as a *genotype*, which is simply a string of genes. A *gene* can be represented as a bit string. Selection of those individuals to go on to the next generation is done by a *fitness function*. The offspring for the next generation are produced when two individuals of the population come together. Reproduction involves taking bits from each parent to form a new individual (generally referred to as *crossover*). *Mutation* is then applied to the resulting population. The combined effects of crossover and mutation mean that GAs can produce offspring that are very different from their parents.

The system configuration for the 2-ANNs model undergoing evolutionary adaptation in this simulation is shown in Figure 7.1.

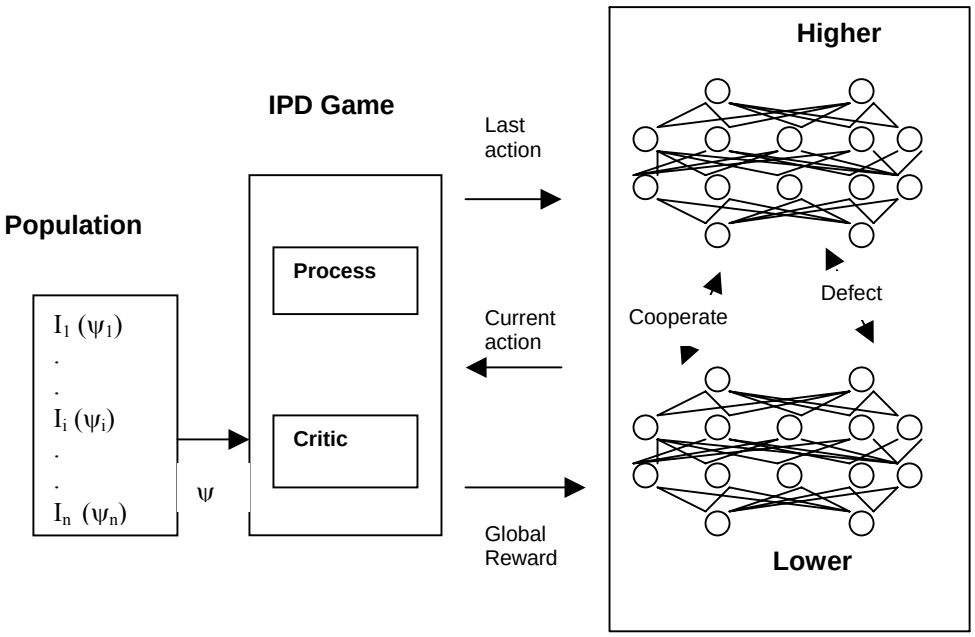



**Figure 7.1 The system configuration of the 2-ANNs neural model in the simulation of the evolution of a bias towards future rewards**
The population consists of a number of individuals (*n*) whose genotype is the bit representation of the bias towards future rewards implemented as a differential bias ($\psi$) as in Section 6.6.3 Each individual's genotype is converted into the real number value from the bit representation and is used to adjust the payoff matrix with the differential bias in the IPD Game environment. The IPD game environment contains a *process* that initializes the input/state to the opponent's previous action (to defect or to cooperate) at the start of each round. The Output/Action is the network's action (to defect or to cooperate). The *critic* assigns a global reward or penalty based on the payoff for the pattern of play from the payoff matrix, at the end of each round.

A population consists of a number of individuals each represented by a genotype of random bit strings. Each genotype contains the value for the bias towards future rewards, which is used to calculate the global reward. The IPD game environment contains a *process* that initializes the input/state to the opponent's previous action (to defect or to cooperate) at the start of each round. At the start of a game the input value of the starting ANN is initialized to a random value. The ANN that starts is selected randomly. The

Output/Action is the ANN's action (to defect or to cooperate). The environment also contains a *critic* that assigns a global reward or penalty to the players, i.e., both ANNs receive the same, at the end of a round, which is the payoff as defined in the payoff matrix. A round consists of both players deciding whether to cooperate or to defect based on the opponent's previous action for the last round. The goal is to maximize the accumulated payoff over a number of rounds.

To summarize, the IPD game consists of two players who compete with each other repeatedly. Each player can either cooperate or defect. Defection is the higher payoff for the individual player, however if both players defect, then the resulting payoff for both is worse. A game consists of one or more rounds. The goal is to maximise the accumulated payoff. A bias towards future rewards is implemented as in the experiment in Section 6.6.2, i.e., as a bias applied to the payoff matrix to calculate the differential payoff, for the following reason: (i) it would seem to promote cooperation behaviour leading to the larger, later reward, the *LL* in Figure 2.1 and (ii) it would seem to address the internal conflict represented by either of the (*C,D*) or (*D,C*) situation. The payoff matrix is the same as that used in the experiment in Section 6.6.2, where the differential bias ($\psi$) with a value between 0 and 1, is assigned to the payoff matrix for both ANNs, which is fixed for the duration of the game. This differential bias is used in the payoff matrix to calculate the payoff. The differential bias is added only to the diagonal terms in the matrix as shown in Figure 7.2.

| | Lower | Lower |
|---|---|---|
| **Higher** | 2 (C,C) | 1(C,D) +ψ |
| **Higher** | 1(D,C)-ψ | 0 (D,D) |

**Figure 7.2 The payoff matrix for the IPD used in the evolutionary adaptation of the 2-ANNs Neural Model with a bias towards future rewards**
A differential bias $\psi$ is applied to the payoff matrix. The differential bias is assigned with a value between 0 and 1 for both ANNs, which is fixed for the duration of the game. This bias is used in the payoff matrix, to calculate the differential payoff and is added only to the diagonal terms in the matrix. This represents a bias towards future reward in the following way: when you are asked to go out, but stay at home (*C,D*), you do not feel so much of a conflict, as you were determined to stay at home anyway. Therefore, you work reasonably well and get a good payoff. Similarly, if you are asked to stay at home, but go out (*D,C*), you feel doubly miserable as you also go against your own preference to the long-term reward.

This represents a bias towards future rewards in the following way: when you are asked to go out, but stay at home (*C,D*), you do not feel so much of a conflict, as you were determined to stay at home anyway. Therefore, you work reasonably well and get a good payoff. Similarly, if you are asked to stay at home, but go out (*D,C*), you feel doubly miserable as you also go against your own bias for long-term future gain. The payoff matrix in Figure 7.2 is similar to the payoff matrix used in the self-control game in the experiments by Brown and Rachlin (refer to Section 2.3), where defecting and choosing the higher current reward, represented by either of the bottom two boxes conflicts with cooperating and choosing the long term reward, represented by either of the top two boxes and hence the dilemma, which is the self-control problem as defined in Section 2.1.

In this simulation, the population consists of a number of individuals each represented by a genotype of random bit strings. Each genotype contains the value for the differential bias $\psi$ as a bit representation of a real number. The bit representation is converted from its binary value and then translated to its

real number by dividing by ten; any value greater that 1.0 is truncated to 1, an example is shown in Figure 7.3.


**Figure 7.3 An example of the genotype for an individual in the simulation of the evolution of a bias for future rewards**
The bit representation is converted to its binary value and then translated to its real number by dividing by ten, any value greater that 1.0 is truncated to 1.

Each of the genotypes for the individuals in the population are randomly initialized at the beginning of the simulation. The population size was constant with the offspring replacing the parents every generation. The genetic operator selection is implemented as a *rank selection*. The individuals compete and then are sorted by their fitness with the fittest being selected to go on to the next generation. The crossover operator implemented performs a *single-point crossover* operation on the offspring in the new generation being created. This means that the parent strings line up and a point along the strings is selected at random (the crossover point). Two offspring are created; the first containing the first bits up to and including the crossover point of one parent followed by the remaining bits of the second parent and the second containing the bits following the crossover point from the first parent and the first bits up to and including the crossover point of the second parent. The crossover mask is defined by a random selected point in the bit representation of the differential bias $\psi$. The mask is constructed from a number of ones followed by a number of zeros. An example is given in Figure 7.4.

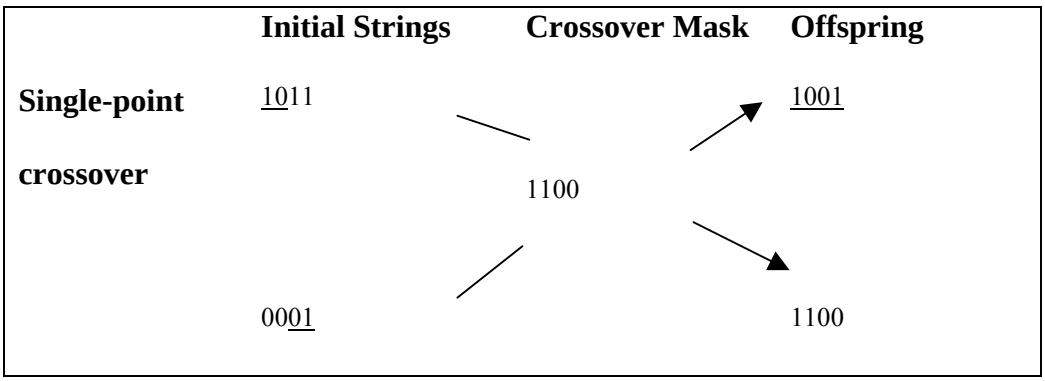



**Figure 7.4 Crossover mask for the evolution of a bias towards future rewards**
The genotype contains a bit string for the differential bias ($\psi$). The single-point crossover operator forms two offspring from two parents using a crossover mask determined by a randomly selected breakpoint in the genotype. To illustrate this, the building blocks of the first offspring are underlined.

Mutations happen infrequently and hence the level of mutation was initially set to modify only a small fraction of the population, and was fixed for the duration of the experiment.

The evolutionary algorithm for the program used in this simulation is given in Figure 7.5. The population size is fixed for the duration of a run. The genotype of each individual is randomly initialized to include the differential bias $\psi$. The rewards are calculated for the payoff matrix using the decoded differential bias. The individuals play a game of Iterated Prisoner's Dilemma using the 2-ANNs model defined previously. The accumulated payoff is retained in addition to its pattern of play. The accumulated payoff becomes the individual's fitness. The individuals are sorted and ranked according to their fitness. The fittest (1- $c_r$, where $c_r$ is the crossover rate) individuals go onto the next generation. From the current generation, pairs of individuals are selected at random to produce two more offspring using crossover and

mutation as described above. This gives a new population and the process is repeated for the desired number of generations.

Initialize population *Pop* to contain 20 individuals, which is held at 20 for the duration of the simulation. Each individual is represented by a genotype of a bit string for the differential bias ($\psi$) generated randomly.

Repeat

1. Decode each individual (genotype) as the differential bias $\psi$
2. Construct the payoff matrix using the decoded differential bias
3. Construct two ANNs with topology of 2-6-6-2 and random initial weights, and train them. The ANNs are trained by competing in a game of Iterated Prisoner's Dilemma for a specified number of rounds. This is repeated three times and the average payoff over all three games is recorded.
4. Calculate the fitness of each individual according to the average training result. Each individual is evaluated by its fitness, where fitness is the accumulated payoff from the payoff matrix. The higher the payoff, the higher the fitness.
5. Sort and rank individuals according to their fitness
6. Select the $1 - c_r$ fittest for the next generation
7. Select parents from current generation and apply search operators, crossover and mutation, to parents for generating offspring, which form the next generation

Until generation = Maximum_generation

**Figure 7.5 Evolutionary algorithm for the simulation of the evolution of a bias towards a future reward**
An initial population is created with 20 individuals. The individual is represented by a genotype of random strings of bits for the differential bias $\psi$. The two ANNs compete in the IPD game. The individual fitness is the accumulated payoff achieved after a number of rounds. The $1 - c_r$ fittest individuals, where $c_r$ is the crossover rate, are selected to go onto the next generation. The remainder $c_r$ of the next generation is constructed by selecting two individuals at random from the current generation, to produce two more offspring using crossover and mutation as described above. This gives a new population and the process continues for a fixed number of generations.

### 7.2.2 Testing Procedure

For a given trial, the evolutionary algorithm of Figure 7.5 was executed for a predefined maximum number of generations, initially set at 20. The population size was fixed at 20 individuals, as this was believed to give enough diversity and coverage of the possible permutations of the genotype.

Each individual in the population played the IPD game, where each game consisted of 250 rounds. As it was found from earlier experiments in Chapter 6, this gave the ANNs a chance to learn (refer to Figures 6.23 and 6.24). The network topology was that of the previous experiments (2-6-6-2) of: (i) one input layer of two input nodes representing, the opponent's last action (to defect or to cooperate), (ii) two hidden layers with six nodes and (iii) two output nodes for the ANN's action, which could either be to defect or to cooperate, as this proved to be the optimal configuration in terms of performance in the validation and verification of the model in Chapter 6. The network topology was fixed for the duration of the trials.

At the end of each game the individual's accumulated payoff is retained in addition to its pattern of play. At the end of 3 games the average payoff is calculated. This average payoff becomes the individual's fitness. To avoid any first player advantage/disadvantage, the starting ANN was selected at random. The individuals in the population are sorted, and ranked according to their fitness. A certain number ($1 - c_r$, where $c_r$ is the crossover rate) of the fittest individuals go onto the next generation.

A number of tests were run to determine the optimal values for the crossover rate, the mutation rate, population size and the maximum number of generations. All runs were conducted under identical conditions to allow an assessment of the variability of results. In the initial test, the crossover rate was set to 0.75, which ensured that 25% of the fittest individuals went onto the next generation by the genetic operator selection, with the remainder of

the individual for the next generation being reproduced by crossover. The mutation rate was held at 1 in 1000 bits and was fixed across generations. From the current generation, two pairs of individuals are selected to produce two more offspring using crossover and mutation as described above. These are selected randomly. This gives a new population and the process is repeated for the desired number of generations.

For each generation the composition of the generation, i.e., how many different differential biases are included in the population, the average fitness and the maximum fitness for the generation are recorded. In addition, the pattern of play was recorded for each game. The success of the GA in this simulation was measured by the prediction in Holland's article (1992) that an individual with an average fitness greater than the average fitness of the generation should have more off-spring and those individuals with an average fitness lower than the average fitness of the generation less offspring.

### 7.2.3 Results and Interpretation

In the first test the maximum number of generations was set to 20, the population size was held at 20 individuals, the mutation rate was set to 1 in 1000, meaning that each bit had a 0.001 chance of being mutated and the crossover rate was set to 0.75, meaning that 25% of the population would go on to the next generation. Since the population was sorted by fitness this meant the fittest individuals had a higher probability of dominating the future generations. To assess if increasing the value of the differential bias had some fitness benefit to the individual, the average fitness for each differential bias

represented in the population was calculated. This was compared to the cooperation percentage for each differential bias. The results are shown in Figure 7.6.

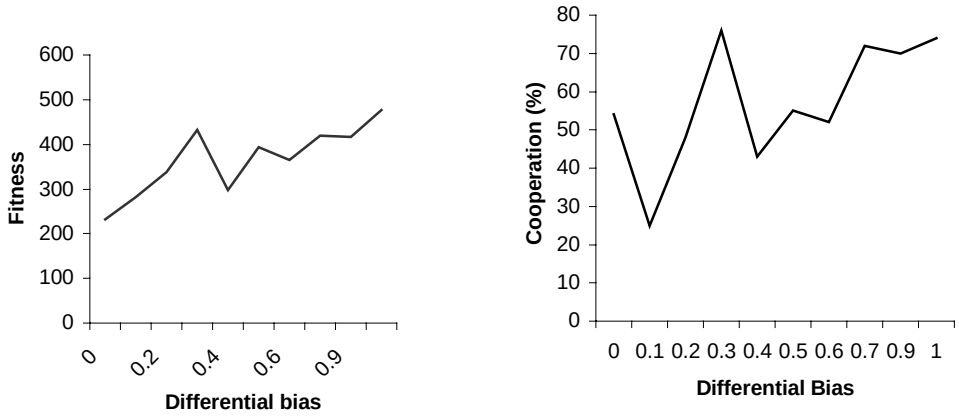



**Figure 7.6 The average fitness and the cooperation (%) by differential bias**
Generally a higher differential bias is associated with a higher average fitness, with some exceptions, e.g. a differential bias of 0.3 has a high average fitness. A high average fitness indicates a high percent of cooperation in the pattern of play. Cooperation is represented by a pattern of play of *CC* or *CD*, i.e., the top row in the payoff matrix.

To explain the exceptions to the general trend in the behaviour of the 2-ANNs model the performance of the GA was assessed. The average fitness was tracked for each generation as shown in Figure 7.7. It was expected that the average fitness should increase with the number of generations, which is suggested by the trendline, however the graph tended to be rather volatile. This behaviour can perhaps be explained by the composition of the final population. It was expected that after a number of generations the population would consist of similar, if not identical individuals (Riolo,1992). Figure 7.8, shows the composition of the final population, it is apparent that in this case this did not happen. Although the population is converging to individuals with

genotypes representing higher values for the differential bias (>0.5) there is still a number of different genotypes represented in the final population.

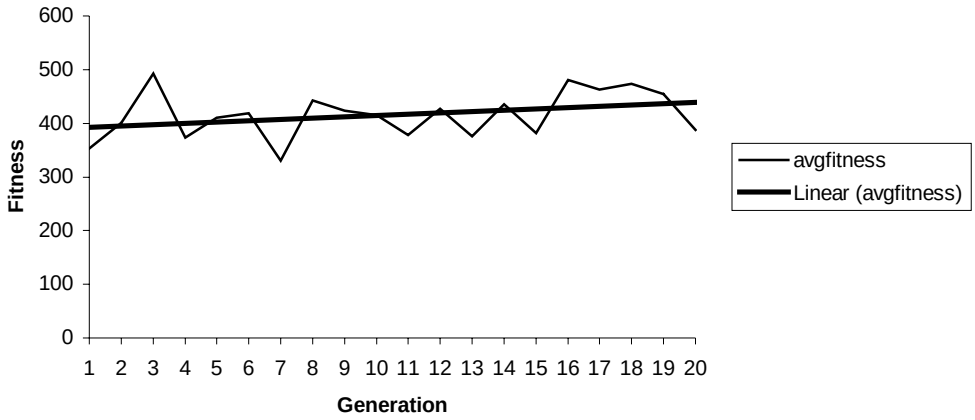



**Figure 7.7 The average fitness for a population of 20 individuals over 20 generations with a crossover rate of 0.75 and mutation rate 0.001**
It was expected that average fitness should increase with the number of generations, which is suggested by the trendline (linear) for the average fitness. However, the graph is tended to be rather volatile, which perhaps can be explained by the diversity in the composition in the final 20th generation.

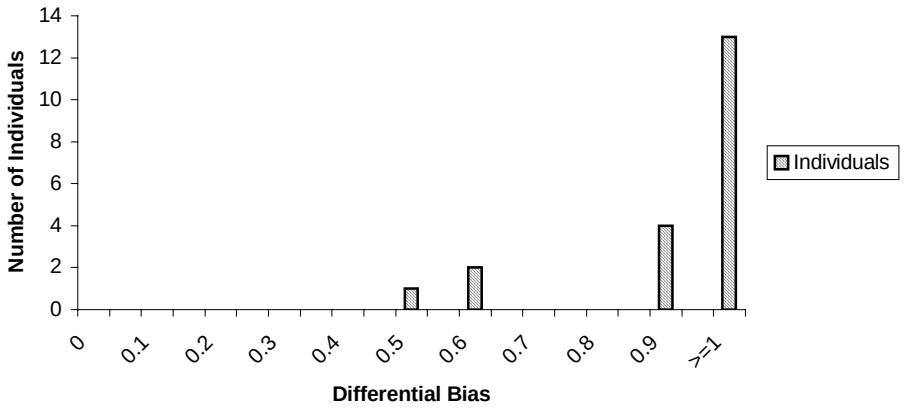



**Figure 7.8 Composition of the final population for a maximum generation of 20, with a population size of 20, a crossover rate of 0.75 and a mutation rate of 0.001**
The final population consists of individuals with genotypes with high values for the differential bias, but convergence has not been met, indicated by the fact that the final population consists of a number of different genotypes.

In addition, Holland's prediction that the individuals with above average fitness levels should have more off-spring did not necessarily hold true.

Figure 7.9 shows the average fitness for each differential bias, taken over all 20 generations, expressed as a percent of the average fitness for the final generation, and the total number of individuals with that differential bias, expressed as a percent of the total number of individuals across all generations (i.e., population size multiplied by maximum generation, which in this case is $20 \times 20 = 400$). The results show that not all possible values for the differential bias have been adequately represented (e.g. 0 and 0.1) and some have been missed completely (e.g. 0.8). In addition, Holland's prediction (1992) that those individuals with an average fitness better than the average fitness for the population have more off-spring, does not necessarily hold true. For example 0.7 has an average fitness greater than that of the population, but has less off-spring than that of 0.6, which has a lower average fitness than the population.

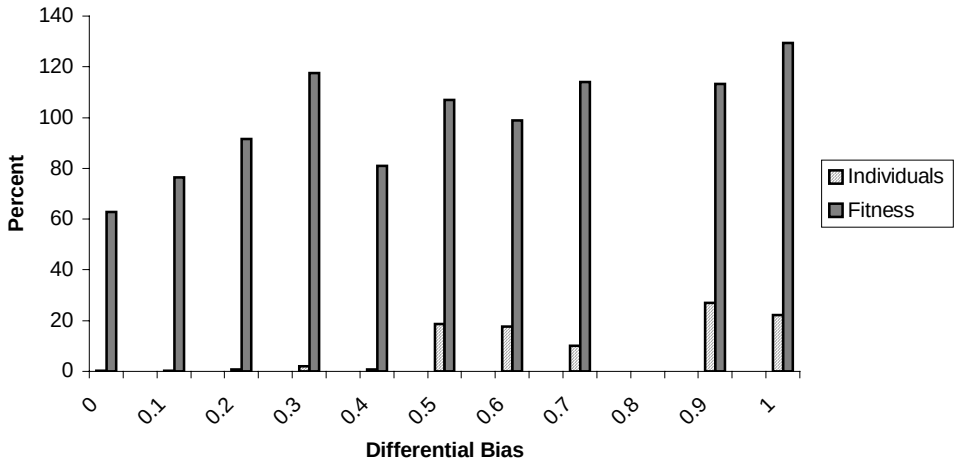



**Figure 7.9 Population composition by differential bias for a maximum generation of 20, with a population size of 20, a crossover rate of 0.75 and a mutation rate of 0.001.**
Not all possible values for the differential bias have been adequately represented (e.g. 0.3, 0.4) some have been missed completely (e.g. 0.8). In addition, it does not necessarily hold true that those individuals with an average fitness better than the average fitness for the population have more off-spring (0.7 and 0.3 have an average fitness better than the population's average fitness but have less off-spring than those with a lower average fitness such as 0.6).

To address the situation shown in Figure 7.9, where not all differential bias are adequately represented, it was decided to increase the mutation rate. Mutation adds randomness to the population composition. It does not on its own advance the search for the best solution, i.e., the fittest individual, it does however, provide an insurance against one individual dominating the population (Holland, 1992). In this next test the maximum number of generations was again set to 20, the population size was held at 20 individuals and the crossover rate was set to 0.75, as in the previous test. The mutation rate was increased to 1 in 100, which meant that each bit had a 0.01 chance of being mutated. Figure 7.10 shows the effect of increasing the mutation rate on the population composition. All possible values for the differential bias are now represented, but Holland's prediction (1992) still does not hold true.

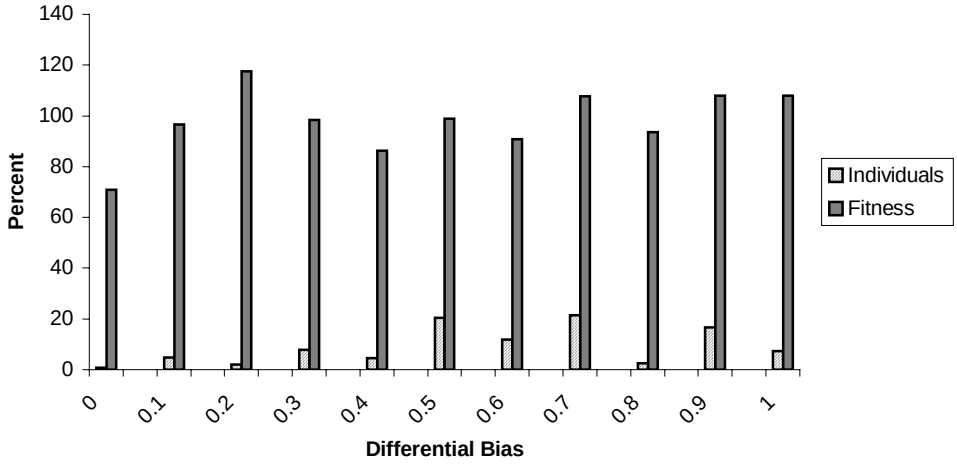



**Figure 7.10 Population composition by differential bias with an increased mutation rate of 0.01 over a maximum generation of 20, with a population size of 20 and a crossover rate of 0.75.**
All possible values for the differential bias are now represented. However, Holland's prediction (1992) does not necessarily hold true, for example 0.2 has an average fitness which exceeds the population average, but this is not represented in the total number of individuals with a genotype of 0.2, similarly 0.1.

The fact that Holland's prediction (1992) still does not hold true with an increased mutation rate can be explained by the composition of the final generation as shown in Figure 7.11. The results suggest that convergence had not been reached, indicated by the number of different values for the differential bias in the final population. This could be because those individuals with a lower than average fitness, e.g., 0.0 and 0.4, have not been discarded from future generations.

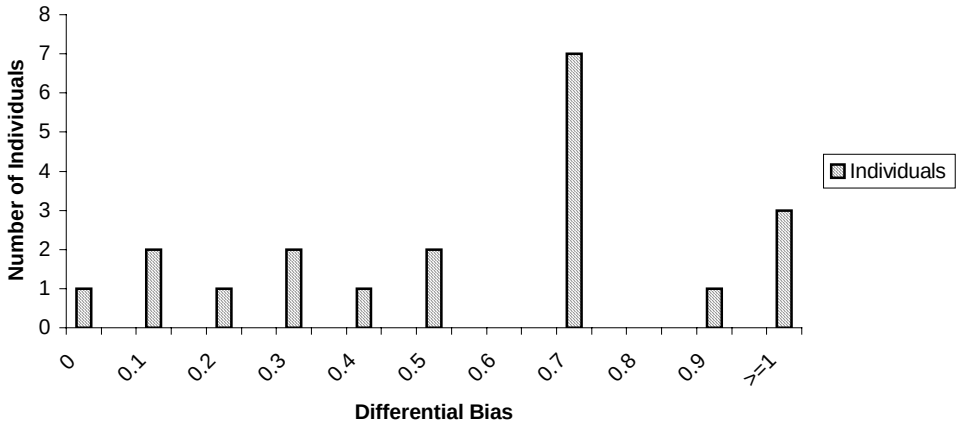



**Figure 7.11 Composition of the final population for a maximum generation of 20, with a population size of 20, a crossover rate of 0.75 and an increased mutation rate of 0.01**
Convergence has not been reached indicated by the high number of values for the differential bias in the final generation. In addition, those differential biases with a lower average fitness than the population have not been excluded, e.g. 0.0.

Based on the results in Figure 7.11 the decision was made to increase the crossover rate to 0.6. This meant that 40% of the population would be selected to go on to the next generation as opposed to 25% in the previous test. Since the population was sorted by fitness, the expectation was that more of the fittest individuals would be chosen by the selection operator. The actual results seemed to behave as expected if the composition of the final population is taken as an indication, as shown in Figure 7.12.

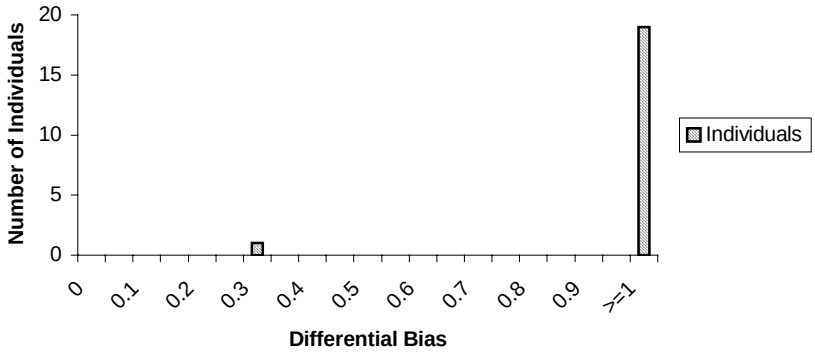



**Figure 7.12 Composition of the final population for a maximum generation of 20, with a population size of 20, a mutation rate of 0.01 and a reduced crossover rate of 0.6**
The population has converged to two values, i.e., the genetic algorithm has pushed the population into two target values. This result would seem to suggest that the individual with the higher value for the differential bias is the optimal one.

To assess if Holland's prediction holds true, the population composition over all twenty generations was tracked and is shown in Figure 7.13.

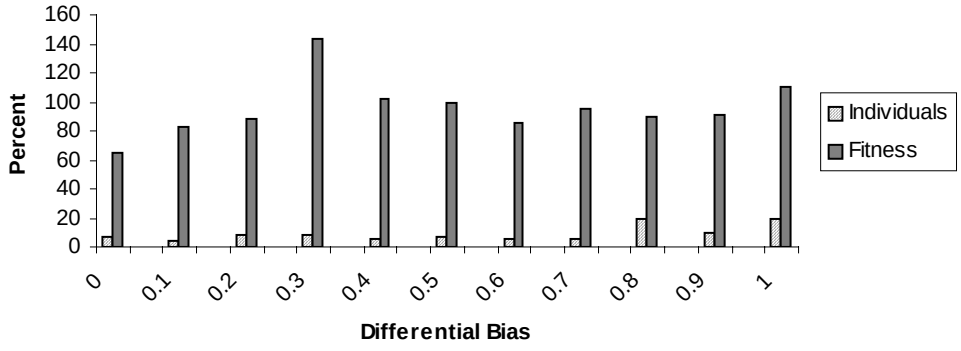



**Figure 7.13 Population composition by differential bias with a reduced crossover rate of 0.6 and a mutation rate of 0.01 over a maximum generation of 20, with a population size held at 20.**
All possible values for the differential bias are represented, but Holland's prediction (1992) still does not necessarily hold true (e.g. 0.8 and 0.3).

From the results in Figure 7.13, the genetic algorithm would seemed to have behaved as expected with those individuals with a higher than average fitness (i.e. 0.3 and 1) having more off-spring. However, there are exceptions such as

0.8, which seem to have a large number of off-spring in relation to its average fitness. The trend for the average fitness was to increase with the number of generations, as shown in Figure 7.14. The peaks and troughs in the graph were less erratic than in the results for the first test shown in Figure 7.7. The graph of the average fitness for each generation in Figure 7.14 still tended to be rather volatile as shown by the peaks and troughs in the graph.

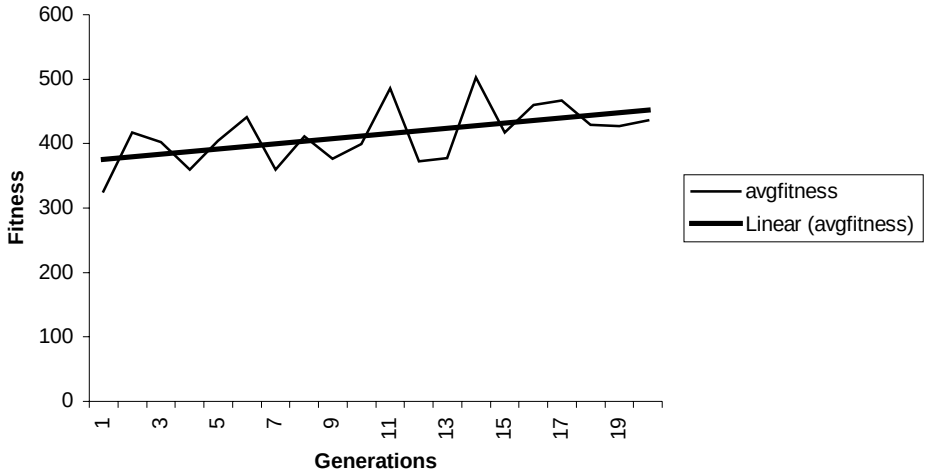



**Figure 7.14 The average fitness for a population of 20 individuals for a maximum of 20 generations with a crossover rate of 0.6 and mutation rate 0.01**
It would seem that the trend for the average fitness was to increase with number of generations, however, with the peaks and troughs this trend is not obvious, suggesting perhaps the genetic algorithm needs to run for more generations to reach convergence.

The decision was made to let the GA run for a maximum number of generations of 100 in order to enable the GA to converge and to see what trends emerged across generations, e.g., whether the average fitness continued to increase erratically with the number generations. Figure 7.15 shows the average fitness for a maximum generation of a 100. Even though there are still peaks and troughs, the graph is less volatile and the trend is for the average fitness (shown by the linear trend line) to increase with each future generation.

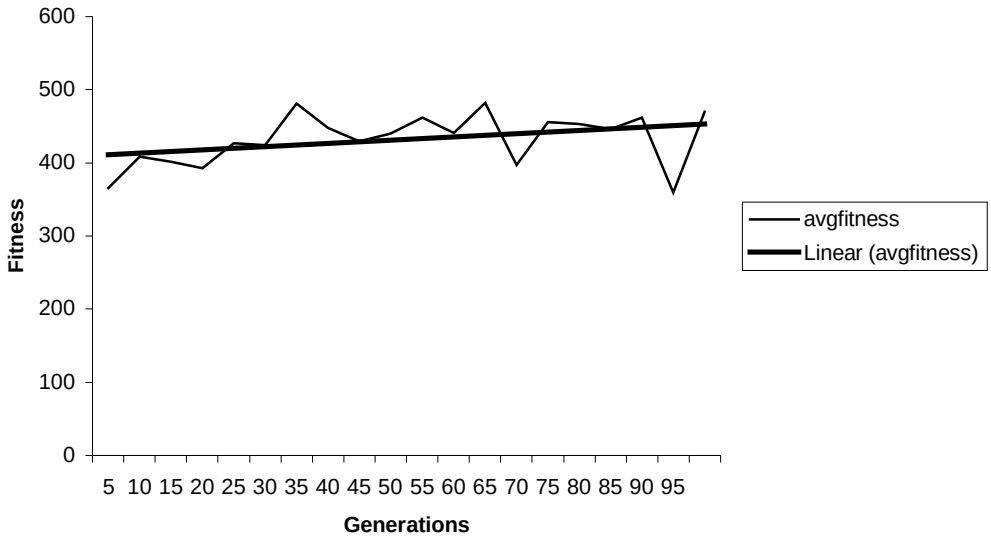



**Figure 7.15 The average fitness for a population of 20 individuals for a maximum of 100 generations with a crossover rate of 0.6 and mutation rate 0.01**
Although the graph still has peaks and troughs the graph is less volatile and it is clear that the trend for the average fitness was to increase with number of generations, shown by the linear trendline.

In addition, Holland's prediction (1992) that those individuals with a higher average fitness than the population will have more off-spring holds true. Figure 7.16 shows the population composition by differential bias. Those individuals with a higher than average fitness (i.e., those individuals with a genotype that has a value for the differential bias of greater than or equal to 0.5) tend to have more off-spring as compared to those individuals with a lower than average fitness (i.e., those individuals with a genotype of a differential bias with a value lower than 0.5). In addition, the graph shows that increasing the differential bias increases the average fitness.

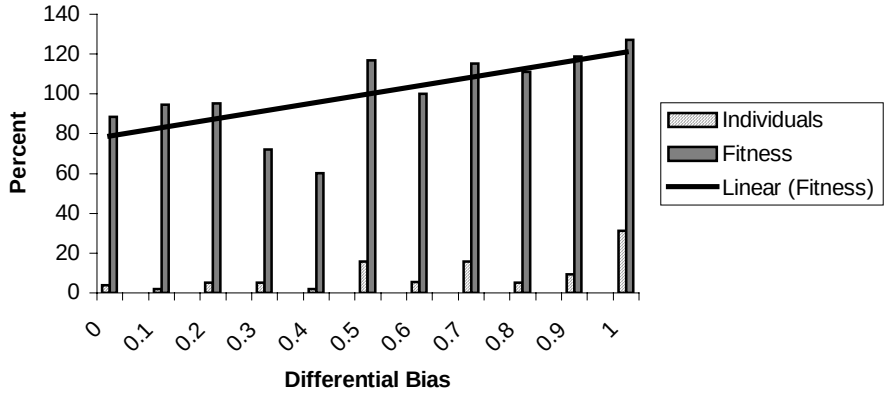



**Figure 7.16 Population composition by differential bias for a maximum generation of a 100 generations with a crossover rate of 0.6 and a mutation rate of 0.01, with a population size held at 20.**
Holland's prediction holds true with those individuals with a higher than average fitness, i.e., > 0.5 having more off-spring than those with a lower average fitness, i.e., <0.5. In addition, the average fitness increases with the differential bias.

Since a higher fitness is associated with a higher percent of cooperation (as shown in Figure 7.6), it follows that a higher differential bias results in a higher percent of cooperation, as shown in Figure 7.17.

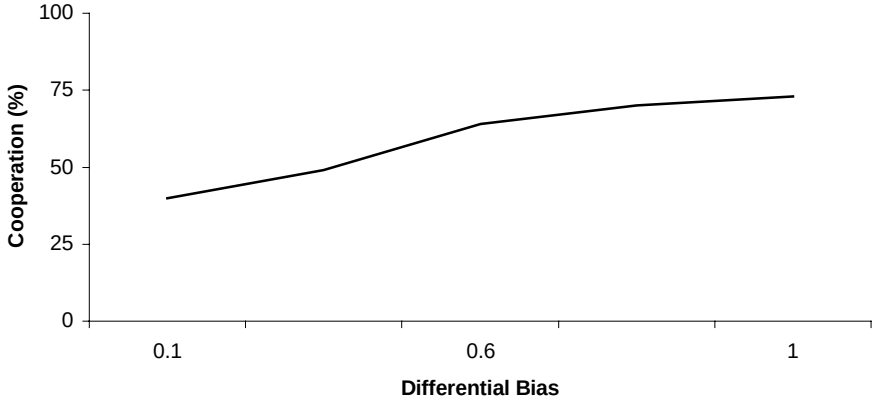



**Figure 7.17Cooperation (%) by differential bias**
Increasing the differential bias increases the cooperation percentage in the same way as increasing the probability of reciprocation increases the cooperation percentage in Baker's experiment (2001) shown in Figure 7.18.

The results in Figure 7.17 show the cooperation percentage, where cooperation is the sum of the number of patterns of play of (*C,C)* or (*C,D)* expressed as a percentage of the total number of plays, increases as the

differential bias increases. This result compares favourably to Baker's (2001) experiments, which showed that the degree of cooperation is strongly affected by the probability of reciprocation, as shown in Figure 7.18.

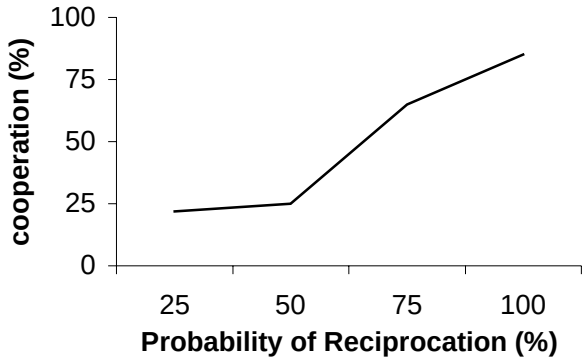



**Figure 7.18 Baker's experiment on the cooperation percentage and the probability of reciprocation**
Increasing the probability of reciprocation introduces the percent cooperation leading to greater self-control adapted from Baker (2001).

### 7.2.4 Conclusion

In the first test of twenty generations, not all possible values for the differential bias were adequately represented and there still existed much diversity in the final population, suggesting that perhaps convergence had not been reached and that the optimal value for the differential bias had not been found. The simulation was then run again with an increased mutation rate increasing diversity in the population, but again convergence was not reached. Reducing the crossover rate ensures that more of the fittest individuals go on to future generations. This certainly was the case, as shown in Figure 7.13, but although the results were an improvement on earlier simulations there was some diversity in the final population as shown in Figure 7.12, suggesting that convergence had not reached and hence an optimal value(s) for the differential

bias had not been found. In the final experiment, where the maximum number of generations was set to 100, a trend showed that a higher value for the differential bias (>0.5) resulted in a higher than average fitness. An increase in the value of the differential bias increases the percentage of cooperation behaviour as shown in Figure 7.6 and Figure 7.17. A higher average fitness is achieved when either both ANNs cooperate, i.e., the pattern of play is mutual cooperation (*C,C)* or where only the ANN representing the higher brain processes, in this case the Temporal Difference network cooperates, i.e., the pattern of play is (*C,D*). This represents the middling behaviour of the situation of when asked to go to the pub you decide to stay at home, but you do not feel too bad as you are still moving towards you later larger reward of good grades. It follows that a high percentage of cooperation behaviour is associated with a higher average fitness. In this first simulation, with the 2-ANNs model it was found that in order to maximize payoff, i.e., fitness, rather than one ANN being the decision-maker, it was necessary that both the ANNs learn to cooperate. In addition, the results suggest that a bias towards future rewards, implemented as a differential bias applied to the payoff matrix, has a fitness benefit for the individual in the game theoretical situation played out here, where payoff equates to fitness. This then has a fitness benefit for the population. This results in a higher average fitness for future generations shown by the trend line in Figure 7.15. Higher values for this bias towards future rewards (i.e., > 0.5) maximizes the fitness of the individual. It follows that this differential bias is a successful mechanism of modeling precommitment in game-theoretical situations since this results in a higher fitness for the organism. The relationship between this differential bias and

cooperation can be compared to the relationship between reciprocation and cooperation in the experiments by Baker (2001); in that increasing the differential bias increases the tendency to cooperate as shown in Figure 7.17.

## 7.3 Scenario of Simulation of the evolution of learning in the context of a bias towards future rewards

In the first simulation, experiments were run to see which values for this bias for future rewards, referred to as the *differential bias*, yielded the higher payoff. The payoff is the individual's fitness. The results showed that high values of the differential bias yielded a higher payoff and hence fitness for the individual. Results also showed that a high payoff indicates a higher percentage of cooperation behaviour. In this simulation the effect of this differential bias on learning is investigated. The premise being that the brain is not hard-wired for every response and learning during an organism's lifetime plays a critical part in deciding which action is the best response to a changing environment. Here the role of learning and the effect of learning on the fitness of the organism in the context this differential bias undergoing evolutionary adaptation is investigated. The types of questions asked in this second simulation are: does having a bias towards future rewards make a difference to the results? What is the effect of different values for this differential bias on learning? For example, does a differential bias of zero prevent learning taking place? What is the effect of learning on the fitness of the individual, on the population and on future generations? In addition, this simulation explores the relationship between learning, specifically reinforcement learning, and evolution on the behaviour of the ANNs. In particular, how do the ANNs deal with discounting, that is, the reduction in value of a reward due to delay? Also

the relationship between learning and fitness of the organism, where fitness is the payoff of the individual, is investigated.

### 7.3.1 Architecture and Algorithm

The simulation program is implemented on the 2-ANNs model in the same way as in the first simulation. The ANN simulating the lower brain functions is implemented with the Selective Bootstrap weight update rule (Widrow et al., 1973) and the ANN simulating the higher brain functions with the Temporal Difference weight update rule (Sutton, 1988).

A bias towards future rewards is implemented as in the previous simulation, i.e., as a bias applied to the payoff matrix to calculate the differential payoff, for the same reasons as in the first simulation, i.e., as it: (i) would seem to promote cooperation behaviour leading to the larger, later reward, the *LL* in Figure 2.1 and (ii) would seem to addresses the internal conflict represented by either the (*C,D*) or (*D,C*) situation. The payoff matrix is the same as that used in the first simulation. A bias with a value between 0 and 1, is assigned to the payoff matrix for both ANNs, which is fixed for the duration of the game. In addition, the population consists of a number of individuals each represented by a genotype of random bit strings. In this simulation, the string of genes includes the learning parameters for both ANNs, as described in the research by Bullinaria (2003) and the differential bias. Each genotype contains the learning parameters (step-size $\alpha$, discount rate $\lambda$) for the ANN representing the higher centre of the brain followed by the learning parameters (learning rate $\eta$) for the ANN representing the lower centre of the brain followed by the value for the differential bias $\psi$. To summarize, the genotype is the bit

representation of the real numbers for $\alpha$, $\lambda$, $\eta$ and $\psi$. The bit representation is converted to its binary value and then translated to its real number by dividing by ten; any value greater that 1.0 is truncated to 1, an example is shown in Figure 7.19.

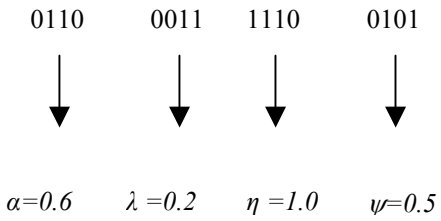



**Figure 7.19 An example of the genotype for an individual in the simulation of the evolution of a bias for future rewards**
The bit representation is converted to its binary value and then translated to its real number by dividing by ten, any value greater that 1.0 is truncated to 1.

Each of the genotypes for the individuals in the population is randomly initialized at the beginning of the simulation. The population size was constant with the offspring replacing the parents in every generation. The genetic operator selection is implemented as *rank selection*. The individuals compete and then are sorted by their fitness with the fittest being selected to go on to the next generation. The crossover operator implemented performs a *uniform crossover* operation on the offspring in the new generation being created. This means that it produces two new offspring from two parents by copying selected bits from each parent using a crossover mask. The crossover mask is defined by the breakpoint between the bit representation of the step-size $\alpha$ and bit representation discount rate $\lambda$ for the ANN representing for the higher brain and the bit representation of the learning rate $\eta$ for the ANN representing the lower centre of the brain followed by the value for the differential bias $\psi$. The mask is constructed from a number of ones followed

by a number of zeros. The beginning and end of the ones is determined by the breakpoints of the building blocks of the genotype, which in this case is ($\alpha$, $\lambda$, $\eta$, $\psi$). An example is given in Figure 7.20.

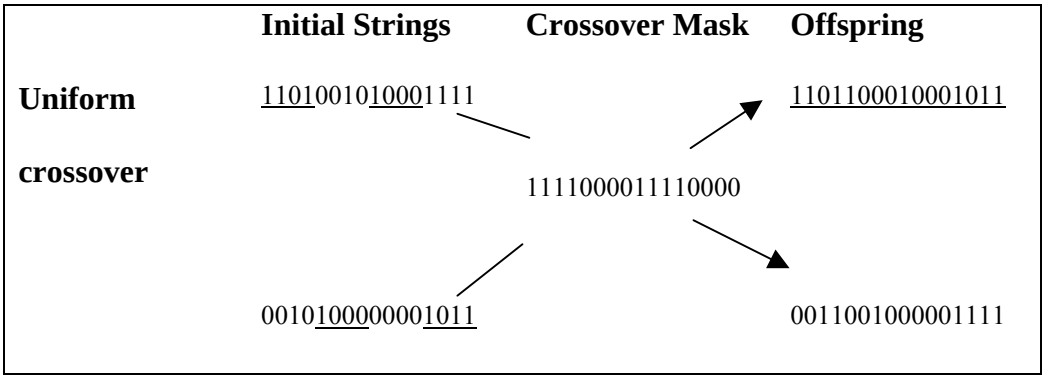



**Figure 7.20 Crossover mask for the evolution of learning in the context of a bias towards future rewards**
The genotype contains the bit strings for the learning parameters followed by a bit string for the differential bias ($\psi$). The uniform crossover operator forms two offspring from two parents using a crossover mask determined by the breakpoints of the building blocks of the genotype to determine which bit comes from which parent. In this case this is ($\alpha$, $\lambda$, $\eta$, $\psi$). To illustrate this, the building blocks of the first offspring are underlined.

Since mutations only modify a small fraction of the population, the mutation rate was set to a suitably low level, i.e., as 1 bit in 1000 individuals across generations, and was fixed for the duration of the experiment.

The evolutionary algorithm for the program used in this simulation is given in Figure 7.21. The population size is constant. The genotype of each individual is randomly initialized to include the learning parameters, step-size and discount rate for the TD network, and learning rate for the Selective Bootstrap network, and the differential bias $\psi$. The genotype of each individual is decoded into two learning rules one for the ANN representing the higher brain processes and the other for the ANN representing the lower brain processes.

Initialize population *Pop* to contain 20 individuals, which is held at 20 for the duration of the simulation. Each individual is represented by a genotype of bit strings for ($\alpha$, $\lambda$, $\eta$, $\psi$) generated randomly.

Repeat

1. Decode each individual (genotype) in the current generation into a learning rule for the ANN representing the higher brain processes and a learning rule for the ANN representing the lower brain processes and the differential bias $\psi$
2. Construct the payoff matrix using the decoded differential bias
3. Construct two ANNs with topology of 2-6-6-2 and random initial weights, and train them using the decoded learning rules.
4. The ANNs are trained by competing in a game of Iterated Prisoner's Dilemma for a specified number of rounds.
5. Repeat step 4 until three IPD games have been played for the desired number of rounds
6. Calculate the fitness of each individual according to the average training result. Each individual is evaluated by its' fitness where fitness is the average of the accumulated payoff from the three IPD games. The higher the payoff, the higher the fitness.
7. Sort and rank individuals according to their fitness
8. Select the $1 - c_r$ fittest for the next generation
9. Select parents from current generation and apply search operators, crossover and mutation, to parents to generate offspring, which form the next generation

Until generation = Maximum_generation

**Figure 7.21 Evolutionary algorithm for the simulation to investigate the role of the learning in the context of the evolution of a bias for future reward**
An initial population is created with 20 individuals. The individual is represented by a genotype of random strings of bits for the learning parameters (step-size $\alpha$, discount rate $\lambda$) for the ANN representing the higher centre of the brain followed by the learning parameters (learning rate $\eta$) for the ANN representing the lower centre of the brain followed by the value for the differential bias $\psi$. The two networks compete in the IPD game. The accumulated payoff is the total payoff achieved after a number of rounds. Repeat this process twice more. The individual's fitness is the average accumulated fitness for the three games. The $1 - c_r$ fittest individuals, where $c_r$ is the crossover rate, are selected to go onto the next generation. The remainder $c_r$ of the next generation is constructed by selecting two individuals at random from the current generation, to produce two more offspring using crossover and mutation as described above. This gives a new population and the process continues for a fixed number of generations.

The rewards are calculated for the payoff matrix using the decoded differential bias. The individuals play three games of Iterated Prisoner's Dilemma using the 2-ANNs model defined previously. For each game, the accumulated payoff is retained in addition to its pattern of play. The average

of the three accumulated payoffs becomes the individual's fitness. The individuals are sorted and ranked according to their fitness. The fittest (1- $c_r$, where $c_r$ is the crossover rate) individuals go onto the next generation. From the current generation, pairs of individuals are selected at random to produce two more offspring using crossover and mutation as described above. This gives a new population and the process is repeated for the desired number of generations. The system configuration for the 2-ANNs model undergoing evolutionary adaptation in this simulation is shown in Figure 7.22.

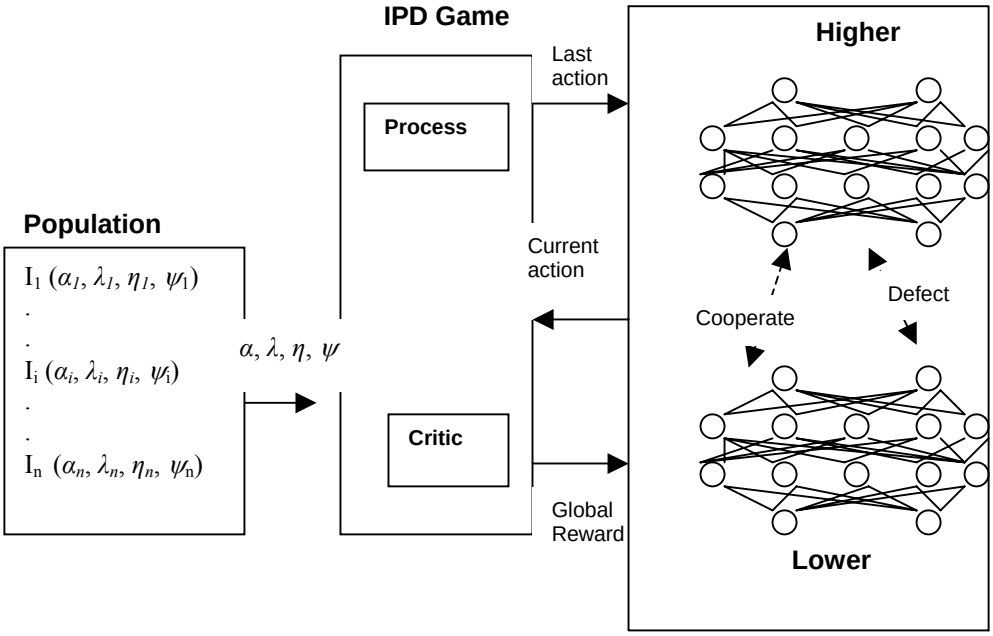



**Figure 7.22 The system configuration of the 2-ANNs neural model in the simulation of the evolution of a bias towards future rewards with learning**
The population consists of a number of individuals ($n$) whose genotype is the bit representation of the learning parameters, step-size and discount rate for the TD network, and learning rate for the Selective Bootstrap network, and the differential bias $\psi$. Each individual's genotype is converted into the real number value from the bit representation and decoded into two learning rules one for the TD network representing the higher brain processes and the other for the Selective Bootstrap network representing the lower brain processes. The rewards are calculated for the payoff matrix using the decoded differential bias in the IPD Game environment. The IPD game environment contains a *process* that initializes the input/state to the opponent's previous action (to defect or to cooperate) at the start of each round. The Output/Action is the network's action (to defect or to cooperate). The *critic* assigns a global reward or penalty based on the payoff for the pattern of play from the payoff matrix, at the end of each round.

The population consists of a number of individuals ($n$) whose genotype is the bit representation of the learning parameters followed by the differential bias $\psi$. Each individual's genotype is converted into the real number value from the bit representation and decoded into two learning rules one for the TD network representing the higher brain processes and the other for the Selective Bootstrap network representing the lower brain processes. The rewards are calculated for the payoff matrix using the decoded differential bias in the IPD game environment. The IPD game environment contains a *process* that initializes the input/state to the opponent's previous action (to defect or to cooperate) at the start of each round. The Output/Action is the network's action (to defect or to cooperate). The *critic* assigns a global reward or penalty based on the payoff for the pattern of play from the payoff matrix, at the end of each round.

### 7.3.2 Testing Procedure

The population size was fixed at 200 individuals and the maximum generation was held at 100, to allow for diversity in the initial population since the genotype now had 4 bit strings representing the step-size parameter, discount rate, the learning rate and differential bias. This was believed to give enough diversity and coverage of the possible permutations of the genotype (each bit string has ten possible values given a total of $10^4$ permutations). For a given trial, the evolutionary algorithm of Figure 7.22 was executed for a predefined maximum number of generations. Each individual in the population played the IPD game three times, where each game consisted of 250 rounds, as this gave the ANNs a chance to learn (refer to Figures 6.23 and 6.24). The individual's fitness was the average of the three payoffs achieved. This was

repeated for a maximum number of 100 generations. The network topology was that of the previous experiments as this proved to be the optimal configuration in terms of performance in earlier experiments in the validation and verification of the model and was fixed for the duration of the trials of: (i) one input layer of two input nodes representing, the opponent's last action (to defect or to cooperate), (ii) two hidden layers with six nodes, as this was found to yield the optimum performance in the IPD game in the experiments in Chapter 6, and (iii) two output nodes for the ANN's action, which could either be to defect or to cooperate (2-6-6-2).

At the end of each game the individual's accumulated payoff is retained in addition to its pattern of play. At the end of three games the average of the three accumulated payoff becomes the individual's fitness. To avoid any first player advantage/disadvantage, the starting ANN was selected at random. All runs were conducted under identical conditions to allow an assessment of the variability of results. The individuals in the population are sorted, and ranked according to their fitness. A certain number ($1 - c_r$, where $c_r$ is the crossover rate) of the fittest individuals go onto the next generation. The crossover rate was set to 0.6, to ensure that more of the fittest individuals went onto the next generation by the genetic operator selection, with the remaining being reproduced by crossover. The mutation rate was reduced to 1 chromosome in 1000 individuals to accommodate the increased population size and increased maximum number of generations, and was fixed across generations. From the current generation, two pairs of individuals are selected to produce two more offspring using crossover and mutation as described above. These are selected

randomly. This gives a new population and the process is repeated for the desired number of generations. The statistics recorded included the highest average fitness for an individual found at each generation, the average fitness for the generation and the pattern of play so that the behaviour for individuals of a particular genotype pattern can be tracked.

### 7.3.3 Results and Interpretation

Firstly, the effect on the fitness of the population as a result of evolving the learning parameters for the TD network, (step-size and discount rate) and the Selective Bootstrap network (learning rate) in the context of the evolution of the differential bias was investigated. It was found that the minimum fitness for any game across all generations was zero. In this case, the pattern of play was for both ANNs to defect (*D,D*) and thus receiving the reward for Mutual defection, which is zero (refer to the payment Matrix in Figure 7.2). The maximum fitness for any game across all generations was 500. In this case, the pattern of play was for both ANNs to cooperate (*C,C*) leading to the highest reward of mutual cooperation, which is two (refer to Figure 7.2) and thus achieved the maximum fitness of 500 (the number of rounds multiplied by the reward for mutual cooperation). Individuals, which achieved a minimum fitness of zero, tended to have genotypes where the learning parameters and differential bias were in the minimum range of valid values, i.e., zero or one. These individuals had a tendency to defect leading to mutual defection and hence a minimum fitness of zero. Individuals, which achieved a maximum fitness, had genotypes with high values for the differential bias and step-size, a low value for the discount rate and a non-zero value for the

learning rate. A much higher proportion of the population had a tendency to cooperate when they had high values for the differential bias (>0.5) and lower values for the learning parameters.

Figure 7.23 shows the trendlines (linear) for each of the parameters that were subjected to the simulation of evolutionary adaptation. A low discount rate (<0.3) and a high value for the differential bias (>0.5) would seem to promote cooperation behaviour leading to the maximum fitness value of 500 (i.e., the number of rounds per game multiplied by the reward for mutual cooperation). It would also seem that as long as the values for the learning-rate and step-size are non-zero, i.e., some learning is taking place, then the fitness of the individual increases as a result of the ANNs tending to cooperate.

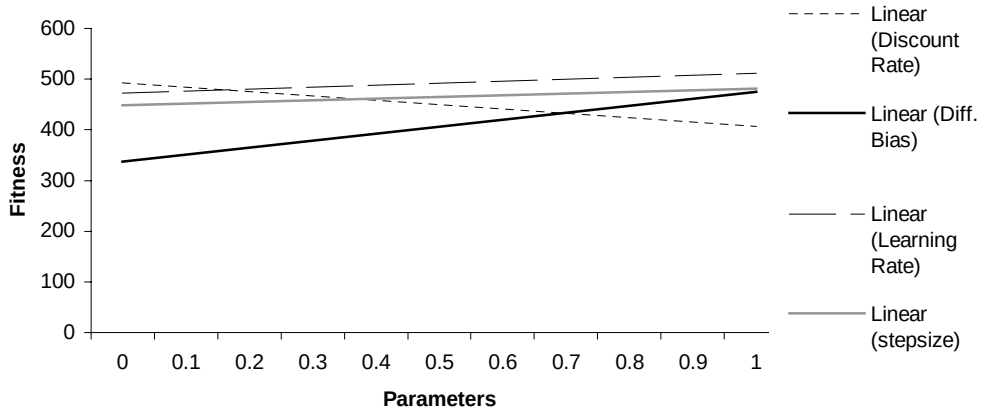



**Figure 7.23 The effect of evolving the learning parameters in the simulation of the evolution of a bias towards future rewards**
The learning parameters, i.e., the step-size and discount rate, for the Temporal Difference Feed Forward Network (TD) and learning-rate for the Selective Bootstrap in the context of the evolutionary adaptation of the differential bias are compared to the fitness of the individual for that genotype. The individual's fitness is the average of the accumulated payoff over three games. A low discount rate (<0.3) and a high value for the differential bias (>0.5) promotes cooperation leading to the maximum fitness value of 500 (the no. of rounds per game multiplied by the reward for Mutual cooperation), as long as the values for the learning-rate and step-size are non-zero.

To investigate the effect of the differential bias on learning the fitness of those individuals with a differential bias of zero was tracked. It was found that both the learning-rate and step-size behaved as expected, i.e. lower values were associated with fitter individuals. A lower value discount rate is associated with a higher fitness, which is surprising given that as the discount rate approaches 1, the ANN becomes more far-sighted and takes future rewards into account more strongly. The maximum fitness, as a result of a game of mutual cooperation was not achieved when the differential bias was zero. The results are summarized in Figure 7.24.

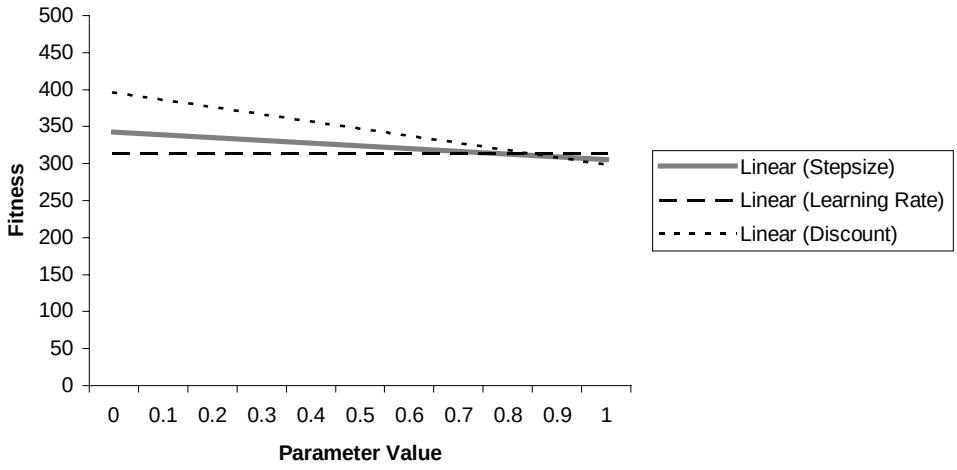



**Figure 7.24 The effect of evolving the learning parameters without a bias towards future rewards**
A lower fitness is achieved since the higher reward of Mutual cooperation is not achieved, indicating that a bias towards future rewards promotes cooperation behaviour.

### 7.3.4 Conclusion

As it can be observed, a higher differential bias is associated with a higher level of fitness. The increase in the differential bias can be explained as follows: as the differential bias approaches 1, the middling behaviour of (*C,D*) becomes more strongly favoured as the reward for this behaviour becomes

more closely aligned to the reward for mutual cooperation (the higher long term payoff). An increase in the fitness of the individual accompanied a non-zero learning rate for the Selective Bootstrap and a non-zero step-size parameter for the TD network. The role of the learning rate in the ANN is to moderate how much the weights are changed at each time step. The learning rate is usually set to some small value. If the learning rate is too large there is a possibility that the network will converge to a less than optimum equilibrium. This did not appear to be the case in this simulation, as long as some learning was taking place maximum fitness could be achieved. In addition, with a differential bias of zero, which is in effect not implementing a bias towards future rewards, learning reverted to the expected behaviour, i.e., low values for the learning parameters were associated with higher levels of fitness. With a differential bias of zero, the maximum fitness was not achieved, indicating that in order to promote cooperation behaviour leading to the higher reward of mutual cooperation there needs to be a bias towards future rewards. This result seems to suggest that learning alone is not sufficient to make a difference to the results.

## 7.4 Concluding Remarks

From the first simulation, concerned with investigating if self-control through precommitment results from an internal conflict between the higher and lower centres of the brain, it was shown that the ANNs in our 2-ANNs model exhibited different behaviours, which resulted in a conflict between Mutual cooperation and Temptation to defect, akin to the self-control problem as suggested by the empirical data of Brown and Rachlin (1999). In particular, in this first simulation it was demonstrated that increasing the level of bias

towards future rewards increased the tendency to cooperate, which is consistent with the empirical results of Baker (2001) that showed that increasing the probability of reciprocation increased the probability of cooperation. In the second simulation, it was shown that implementing a bias towards future rewards, as a differential bias is a useful mechanism to maximize the fitness of an individual. In particular, it was demonstrated in the second simulation that although it was necessary that some learning occurred, i.e., learning parameters must be non-zero, individuals with a higher value for the differential bias fared better. In addition, it was shown that a high level of differential bias, rather than learning alone, increases the tendency to cooperate, supported by the fact that individuals with a differential bias of zero did not achieve the maximum fitness associated with the reward for mutual cooperation. In order to maximize fitness, both ANNs had to play a game of mutual cooperation. It follows that a high value for the differential bias results in a high fitness benefit for the individual. The results suggest that this differential bias enhances cooperation, which could be interpreted as precommitment and hence that self-control through precommitment has an evolutionary benefit in a game-theoretical situation. In the context of the Brown and Rachlin (1999) experiment, the model is learning to cooperate with oneself.

# Chapter 8

# 8 Can self-control through precommitment be explained by evolutionary game theory?

## 8.1 Retrospective

The aim of this thesis was to attempt to explain how evolution has resulted in self-control through precommitment behaviour. We recognize we have problems with self-control and implement precommitment behaviour that limits our future choices, making it difficult to change our preferences at some later time. Three possible explanations for the evolution of this behaviour have been investigated in this thesis. The explanations are neither mutually exclusive nor exhaustive.

The first explanation for the evolution of self-control through precommitment is that it results from an internal conflict between the higher and lower centres of the brain. The internal conflict may be a result of the different centres of the brain evolving different behaviours in response to a dynamic environment. In this case the type of questions that are explored are: when did low-level intrinsic behaviours manifest themselves? Alternatively, when did behaviours that lead to long term rewards "win" in the competition for control of the organism? Did the higher centre of the brain take on the role of decision-maker in determining what behaviours are appropriate when?

The second explanation investigated the evolution of self-control through precommitment in the context of games. The theoretical premise in this case is

that if the evolution of self-control through precommitment is a successful strategy for game-theoretical situations, then there must be a fitness benefit for the organism, which is reflected in its payoff.

The final explanation views self-control through precommitment as a result of a best evolutionary compromise to a dynamic and complex environment. This is based on the theoretical premise that there is a basic hard-wiring in the organism for self-control, but it is not feasible for evolution to program the brain with a direct hard-wired response to every situation it could meet in such an environment. Instead, there is a capacity for learning. Evolution cannot always result in the optimum fitness benefit for the organism during its lifetime and learning plays a part in maximizing fitness in certain situations. Hence, natural selection has provided a mechanism for allowing learning to effectively take control when cues are strong enough, with the result reflected in a fitness benefit.

Chapter 2 provides the theoretical foundation on which this thesis is based. To summarize, it defines self-control as choosing a larger-later reward over a smaller-sooner reward. Initially our preference is for the larger-later reward, but at some later point in time our preferences are reversed, and the smaller-sooner reward is preferred. We recognize that this reversal of preferences happens and exercise self-control by precommiting to the larger-later reward, when it is the preferred choice, by either denying the smaller-sooner reward, or making it difficult to choose the smaller-sooner reward in the future. Chapter 2 goes on to discuss self-control in the context of games. We saw that

self-control can be defined as learning to cooperate with oneself. Brown and Rachlin (1999) used a version of the IPD game to illustrate this. A factor that determines if one will continue to cooperate with oneself is the probability of reciprocation, which Baker (2001) showed has a direct correlation to cooperation. In this thesis, the theoretical premise is made that a similar relationship exists for precommitment and cooperation.

In Chapter 3, the cognitive neuroscience model for self-control was presented after considering both neurophysiological models and abstract models of related behaviours. The model is adapted for this thesis and implemented as a 2-ANNs model competing in a game-theoretical situation using RL. This set the scene for the explanations listed above to be investigated in Chapters 6 and 7.

Chapters 4 and 5 provided the necessary groundwork by introducing the main techniques to be used in the context of self-control notably reinforcement learning, artificial neural networks and genetic algorithms.

After laying down this groundwork, Chapter 6, then began the quest to explore the evolution of this complex behaviour by firstly verifying the model in different game theoretical situations against empirical data on self-control. The results of the RBG game confirmed the theoretical premise that the Temporal Difference update rule exhibited behaviours associated with the higher centres of the brain such as planning and control, and that the Selective Bootstrap update rule exhibited low-level intrinsic behaviours associated with

the lower centres of the brain, such as actions that lead to immediate gratification. The results from the IPD game also confirmed this premise, as the dominant behaviour from the Selective Bootstrap network was to defect (a low-level intrinsic behaviour, associated with immediate gratification (*SS*)) and the dominant behaviour from the Temporal Difference network was to cooperate (a high-level behaviour, associated with a long-term reward (*LL*)). To this framework we added a bias towards future rewards implemented in three ways. The *variable bias* technique implemented a bias towards future rewards by simply varying the input values of the ANN's bias node between zero and one. Increasing this *variable bias* enhances cooperation behaviour, which is the desired behaviour if this technique does indeed represent precommitment. A potential problem with this technique is that there is a possibility that during training the final values of the weights may cancel out the effect of this variable bias, as shown in Figure 6.35. The second technique implemented a bias towards future rewards as an *extra input* to one or both of the ANNs in the 2-ANNs model. The *extra input* technique did not have this problem and again cooperation was enhanced. With this *extra input* technique there is no distinction between the situations represented by (*C,D*) and (*D,C*), which is not necessarily the case. The final technique implemented the bias towards future rewards as a *differential bias* added to the global reward in the payoff matrix. The results showed that cooperation behaviour was further enhanced. In addition, the dilemma represented by the situations (*C,D*) or (*D,C*) is dealt with by the different rewards, i.e., the reward for the middling behaviour represented by (*C,D*) increases, whilst the reward for the more negative behaviour represented by (*D,C*) decreases. For these reasons it was

considered that the *differential bias* technique was the best technique to model precommitment. Its behaviour suggests that this bias towards future rewards has the effect that precommitment would have, in that precommitment acts in the same way as the probability of reciprocation in Baker's experiment promoting cooperation with ones self leading to greater self-control. In Chapter 7, this was explored further in the context of evolution.

Chapter 7 explored the possible explanations as to how this bias towards future rewards has evolved, such that people must use a bias towards future rewards to control their future self. Various explanations were explored. In exploring the explanation that self-control results from an internal conflict, the following questions were investigated and the results suggested the following answers:

What behaviour patterns emerged and what were the results of these patterns of behaviour?

*The individuals that had a tendency to cooperate achieved the higher payoff and dominated the gene pool. This can be explained as follows: to maximize fitness, the ANNs had to play a game where the dominant behaviour was to cooperate, if both ANNs cooperated, then the ANNs receive the higher reward of mutual cooperation, hence the higher net payoff for the organism.*

Did a particular ANN favour a particular pattern of play?

*The dominant behaviour from the Selective Bootstrap network is to defect and the dominant behaviour from the Temporal Difference network is to cooperate. The pattern of play however, depends on the value of the bias*

*towards future rewards; the higher the value, the higher the tendency to cooperate from both networks.*

Is any one ANN the decision-maker?

*Each ANN seemed to behave as an autonomous agent, learning simultaneously, but separately in a shared environment.*

What was the effect of this bias towards future rewards on the pattern of play?

*A higher bias towards future rewards promoted cooperation. Increasing the bias towards future rewards reduced the conflict between the low-level intrinsic behaviour of defection (SS) and the high-level complex behaviour of cooperation (LL). This was done by increasing the disparity between the rewards for, the middling behaviour of staying home and studying although wishing you had gone to the pub represented by the (C,D), and the negative behaviour of the situation when asked to go to the pub you go, but feel doubly miserable as you are going against your future self and the possibility of long-term gain.*

In exploring if the evolution of self-control through precommitment is a side effect of playing games, the following questions were investigated and the results suggested the following answers:

What values for this bias towards future rewards evolve?

*To maximize fitness both ANNs had to play a game of mutual cooperation. Cooperation behaviour is associated with higher values for the bias towards*

*future rewards. Results showed that increasing this bias towards future rewards, promotes cooperation.*

In exploring if self-control through precommitment is a best evolutionary compromise to environmental complexity and variability, the effect of evolving of learning in the evolution of self-control is investigated. The question asked and, given the results, the following answer is suggested:

What is the effect of this bias towards future rewards on learning?
*Implementing a bias towards future rewards would seem to increase the rate of learning suggesting that a higher learning rate was associated with a higher level of fitness. When no bias towards future rewards was implemented the tendency to cooperate was reduced suggested by the fact that none of the individuals with a bias towards future rewards of zero achieved the maximum fitness.*

## 8.2 Conclusion

The results of the first simulation, which investigated if self-control through precommitment is a result of an internal conflict, showed that there are differences in the behaviour of the system components, i.e., the two artificial neural networks simulating the higher and lower centres of the brain. The results from this first simulation in Chapter 7 expand on the results of Chapter 6, where it was shown that the ANNs demonstrated different behaviours consistent with the expected behaviours of the brain regions modelled, e.g., the Selective Bootstrap network had a tendency to defect (a low-level intrinsic behaviour). This supports the theoretical premise, specified in Chapter 2, that

self-control can be explained in terms of multiple selves. Each self exhibits or wants a different behaviour. In summary, the results from this first simulation supports the explanation that self-control is a result of an internal conflict between the present self, guided by immediate outcomes, and the future self, guided by future prospects.

In addition, the results from this first simulation support the explanation that a bias towards future rewards (modeled as a *differential bia*s for the reasons listed in Section 8.1) is a useful mechanism in a game-theoretical situation and that there is an evolutionary benefit in a game-theoretical situation. The results showed that as the value of the bias towards future rewards increases the tendency to cooperate also increased. This proved to be a useful fitness benefit resulting in a pattern of mutual cooperation leading to a higher payoff. This supports the empirical results of Baker (2001), which showed increasing the probability of reciprocation increases cooperation. The results from this simulation showed that implementing a bias towards future rewards behaves in the same way as the probability of reciprocation, and hence increases cooperation.

In the final simulation, the explanation that learning, as opposed to evolution, is critical in formulating self-control through precommitment as a best response to a complex and dynamic environment was investigated. This is based on the theoretical premise that the brain is not hard-wired for every response and that learning, during an organism's lifetime, plays a part in making the decision as to which action is the best response to a changing

environment. The results from this final simulation suggest that although learning does play a part, it is not mutually exclusive to evolutionary factors and that evolutionary factors, as opposed to learning alone, plays a crucial role in the development of this complex behaviour.

Even in this rather coarse simulation of biological evolution, the results support an evolutionary basis for self-control through precommitment behaviour, as there is a fitness benefit associated with implementing a bias towards future rewards. This does not suggest that learning plays no role in self-control through precommitment behaviour, but rather that evolution has provided us with a the capacity to bias our preferences to future rewards in order to control our future actions. The results provide clues as to the explanation of the evolution of this complex behaviour, but much still remains to be learned. For example, there still remain missing pieces of the puzzle, as to the exact hard-wiring of the cognitive architecture for this bias towards future rewards.

## 8.3 Contributions

The major contributions of this thesis were firstly to provide a cognitive architecture that supports self-control through precommitment behaviour, and secondly give possible explanations for the origin of this complex behaviour. In this thesis the model of self-control as an internal process from the viewpoint of modern cognitive neuroscience, depicted in Figure 3.3, is brought to life in a computational model with behavioural predictability. The model goes beyond the connectionist framework as simply an information processing system that receives some input, processes it and then outputs

some results. The model simulates self-control through precommitment behaviour in a functionally decomposed system providing a deeper understanding of the psychological processes of how self-control emerges. The results from the 2-ANNs model are compared with the empirical data in psychology on self-control. The results from the two ANNs, representing the higher and the lower systems of the brain, competing in the RBG are compared to the economic literature, specifically Rubinstein (1982) and Kreps (1990), which suggest that the player who is least myopic will fare better. The results from the 2-ANNs model demonstrate this premise. The results of Brown and Rachlin (1999) show a close analogy in the structure of cooperation behaviour and the structure of self-control behaviour. Brown and Rachlin (1999) used a version of the IPD game, called the self-control game, to demonstrate that self-control can be viewed as cooperating with one's self, the higher the probability that one will cooperate with one's future self the greater the self-control. This probability was referred to as the probability of reciprocation. In Chapter 6 of this thesis a similar experiment was carried out with the 2-ANNs model competing in the IPD game, simulating the self-control game. The results emulated those of Brown and Rachlin (1999), which showed that the self-control problem could be seen as a question of: if I cooperate now, will I cooperate in the future? Given this success, a bias towards future rewards was added to the 2-ANNs model. The results suggested a positive correlation between this bias towards future rewards and cooperation behaviour emulating Baker's (2001) results, which showed a positive correlation between reciprocation and cooperation. From this point (*Chapter 7*) the neural model is subjected to evolutionary adaptation

consistent with evolutionary theory. In Chapter 7, we explored the results of Chapter 6 in the context of evolution, demonstrating in the simulation of self-control as an internal conflict and that this bias towards future rewards behaves in the same way as the probability of reciprocation and cooperation (Baker, 2001). In addition, the results in Chapter 7 of a functionally decomposed system undergoing evolutionary adaptation, support the premise that an internal conflict between the higher and lower centres of the brain exists in self-control problems, and that a bias towards future rewards is a useful mechanism to reduce this conflict. In the final simulation in Chapter 7, it was shown that it is this bias towards future rewards that determines the level of cooperation behaviour and not learning alone. These results provide a deeper understanding of the relationship between the hard-wiring of a cognitive architecture for self-control, which arises through evolutionary processes, and learning of self-control. This multi-level approach is important in model building. In this thesis, the marrying of empirical and computational research has proven to be a productive path to progress future research on the biophysical processes that underlie self-control through precommitment behaviour.

As a sideline contribution, this thesis has examined the capabilities of combining the techniques of RL, GAs and ANNs. In particular, the extent to which different forms of RL work in MAS. The model presented in Figure 3.3 and developed in this thesis, is a MAS comprised of two independent learners simultaneously learning in shared environment, using RL, playing a general-sum game. General-sum games are interesting as the payoffs are neither

wholly adverse nor wholly competitive. From a review of the literature on MARL this is an active area of research. One of the unresolved issues is the lack of a clearly defined statement on when RL can be applied to general-sum games usefully and in what form. To date there does not exist an algorithm for MARL that can be applied to the complete set of general-sum games. In this thesis we broke from the traditional framework for MARL and games, i.e., Markov games, and implemented our learners as autonomous agents learning simultaneously. This removed the limitation of centralized learning, which widens the scope for the learner's behaviour. The results demonstrated that RL could be usefully applied in this context in that the ANNs learned to behave rationally in two general-sum games that model a real-world situation, i.e., the division of the resource is close to half in the RBG, and both ANNs settle into a play of mutual cooperation in the IPD game, and that convergence was reached with the ANNs' weights settling into an equilibrium.

The following is a list of specific contributions in order of appearance.

1. In Chapter 2 a departure from traditional theories of the brain was proposed. This thesis presents a novel view of the higher and lower brain regions cooperating, i.e., working together, as opposed to the classical view of the higher brain as the controller overriding the lower brain.

2. In Section 3.3 a top-down approach to modeling self-control behaviour in the computational model is adopted. This is a novel approach to modeling the brain, but is appropriate given the complexity of the behaviour that is being modeled. The premise here is that in the behaviour self-control it is

more meaningful to describe the overall image rather than each individual neuron and for this reason a holistic approach to the brain as a functionally decomposed system is adopted.

3. In Chapter 6 the somewhat vague abstract behavioral model of self-control as an internal process taken from psychology and presented in Chapter 3 is implemented as a computational model, which to the best of my knowledge is the first time that this has been undertaken and hence a significant contribution. The psychological model explains self-control behaviour at an abstract level. The computational model, developed in this thesis, provides a greater understanding of self-control behaviour by implementing the abstract psychological model and providing a possible explanation for precommitment behaviour by adding a bias towards future rewards. The results suggest that this bias enhances cooperation behaviour and hence could be interpreted as precommitment.

4. In Chapter 6, RL and ANN are combined in a new value function approximation scheme. The higher and lower brain regions are implemented as ANNs with RL. The higher brain region is implemented with Temporal Difference learning with a simple lookup table for each state-action pair. TD learning is implemented as TD(0), as only one state preceding the current one is changed by the TD error. The lower brain region is implemented with the Selective Bootstrap weight update rule. The model framework is implemented as two players learning simultaneously, but independently, competing in general-sum games.

5. From a review of the literature on RL, summarized in Section 6.2, RL is considerably more difficult to implement in general-sum games. As part of this thesis, a feasibility study on the extent RL can be applied to such games was carried out. The results showed that RL could be applied successfully to general-sum games that model some real-world situation.

6. In addition, the computational model of two autonomous players simultaneously learning in a shared environment makes it a multi-agent system. MARL is an area of intense research activity. The results of this thesis contribute to a greater understanding of MARL by showing that convergence is reached in such a system.

7. Section 6.3 introduced an alternative to the traditional measure of learning in ANN as being a function of some error. The concept of a *mistake* was introduced as a measure of learning in the RBG. The ability of the ANN to learn was measured by the number of mistakes the ANN made.

8. In Chapter 7 all three techniques RL, ANNs and GAs are combined. This is an area in its infancy. The results of Chapter 7 contribute to a greater understanding of such hybrid systems.

## 8.4 Future Work

Following the results in this thesis there are a number of questions and avenues for future research. This thesis has focused on creating an abstract neural network model, which was realistic given the complexity of the behaviour we were trying to simulate. In future work the aim would be to give the model a stronger biological basis based on the known neurophysiological

data of reinforcment learning and reward-directed behaviour. Future research would still be based on the theoretical premise that the higher and the lower systems of the brain are largely independent and are locked in some form of internal conflict for the optimal control of the organism. However, the ANNs comprising the higher and lower brain system will be made more biologically realistic by including at the node level, models of neurons, which are more biologically plausible, such as leaky-integrate-and-fire models. The overall aim is to build a computational model that can help guide research on the biophysical processes that underlie the mechanisms suggested by the functional analysis of the more abstract model from this thesis.

## References


Ainslie G., Specious reward: A behavioral theory of impulsiveness and impulse control. *Psychological Bulletin*, 82: 463-496, 1975.

Ainslie G., *Picoeconomics: The strategic interaction of successive motivational states within the person*. Cambridge University Press, New York, 1992.

Ainslie G. and Haslam N., "Hyperbolic Discounting". In Loewenstein G. and Elster J. (Eds)., *Choice over Time,* pp. 57-92, Russell Sage Foundation, 1992.

Ariely D., *Procrastination, Deadlines, and Performance: Self-Control by Precommitment*. MIT Press, Cambridge, MA, 2002.

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

Axelrod R. and Hamilton W.D., The Evolution of Cooperation. *Science,* 211:1390-1396, 1981.

Back T., Optimal mutation rates in genetic search. In S. Forrest (Ed.), *Proceedings of the fifth International Conference on Genetic Algorithms*, pp. 2-8, Morgan Kaufmann, San Mateo, CA, 1993.

Baker F., Probability of reciprocation in repeated Prisoner's dilemma games. *Journal of behavioral Decision Making*, 14(1): 51-67, 2001.

Balch T., Learning roles: Behavioural diversity in robot teams. In Sen Sandip (Ed.), *Collected papers from the AAAI-97 workshop on multiagent learning,* 1997

Baldwin J.M., A new factor in evolution. *American Naturalist*, 30:441-451, 1896. Avail. online at http://www.santafe.edu/sfi/publications/Bookinfo/baldwin.html

Banfield G. and Christodoulou C., On Reinforcement Learning in two player real-world games. In *Proc. ICCS ASCS Int. Conf. on Cognitive Science*, 22, 2003.

Banfield, G. and Christodoulou, C., Can Self-Control be Explained through Games? In A. Cangelosi, G. Bugmann, R. Borisyuk (Eds), *Modelling Language, Cognition and Action*, *Progress in Neural Processing*, World Scientific, 16:321-330, 2005.

Barto A.G., Adaptive critics and the basal ganglia. In Houk J.C., Davis J. and Beiser D. (Eds), *Models of Information Processing in the Basal Ganglia,* MIT Press, Cambridge, MA, pp. 215-232, 1995.

Barto A.G., Sutton R.S. and Anderson C.W., Neuron like adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5): 834-846, 1983.

Baumeister R.F., Transcendence, guilt, and self-control. *Behavioral and Brain Sciences,* 18:122-123,1995.

Beiser D.G. and Houk J.C., Model of Cortical-Basal Ganglionic Processing: Encoding the Serial Order of Sensory Events. *The American Physiological Society*, 3168-3190, 1998.

Bellman R.E., A problem in the Sequential design of experiments. *Sankhya*, 16:221-229, 1956.

Bellman R.E., *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957a.

Bellman R.E., A Markov decision process. *Journal of Mathematical Mechanics,* 6:679-684, 1957b.

Binmore K., *Fun and Games: A text on Game Theory*. D. C. Heath and Co., Lexington, MA, 1992.

Bjork J.M, Knutson B., Fong G.W., Caggiano D.M, Bennett S.M, Hommer D., Incentive-Elicited Brain Activation in Adolescents: Similarities and Differences from Young Adults. *Journal of Neuroscience*, 24(8):1793-1802, 2004.

Bowling M., Convergence and No-Regret in Multiagent Learning. In *Advances in Neural Information Processing Systems*, pp. 209-216. MIT Press, 2005

Bowling M. and Veloso M. Analysis of Stochastic Game Theory for Mutiagent Reinforcement Learning. Technical Report CMU-CS-00-165, Carnegie Mellon University, Pittsburgh, 2000.

Bowling M. and Veloso M. Rational and Convergent Learning in Stochastic Games. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 1021-1026, Seattle, WA, 2001.

Brown J. and Rachlin H., Self-control and Social Cooperation, *Behavioral Processes* 47:65-72, 1999.

Brown J., Bullock D. and Grossberg S., How the basal ganglia use parallel excitatory and inhibitory learning pathways to selectively respond to unexpected rewarding cues. *Journal of Neuroscience,* 19(23):10502-10511, 1999.

Bullinaria J.A., Evolving efficient learning algorithms for binary mapping. *Neural Networks*, 16:793-800, 2003.

Burnham T. and Phelan J., *Mean Genes: From Sex to Money to Food: Taming Our Primal Instincts.* Perseus Publishing, USA, 2000.

Carver C. S. and Scheier M. F., *On the Self Regulation of Behavior.* Cambridge University Press, Cambridge UK, 1998.

Churchland P.S. and Sejnowski T.J., *The Computational Brain.* MIT Press, Cambridge MA, 1992.

Claus C. and Boutilier C., The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 746-752, AAAI Press, 1998.

Cohen J. D., Braver T.S. and Brown J.W., Computational perspectives on dopamine function in prefrontal cortex. *Current Opinion in Neurobiology,* 12:223-229, 2002.

Cosmides L. and Tooby J., Evolutionary Psychology and the Generation of Culture, Part II Case Study: A Computational Theory of Social Exchange. *Ethology and Sociobiology*, 10:51-97, 1989.

Damasio A.R., *Descartes Error: Emotion, Reason and the Human Brain*. Putnam, New York, 1994.

Damasio A.R., Tranel D. and Damasio H., Individuals with sociopathic behavior caused by frontal damage fail to respond automatically to social stimuli. *Behavioural Brain Research*, 41:81-94, 1990.

Dawkins R., *The Selfish Gene.* Oxford University Press, Oxford, UK, 1989.

Dayan P. and Abbot L.F., *Theoretical Neuroscience Computational and Mathematical Modeling of Neural Systems.* pp. 331-358, MIT press, Cambridge, MA, 2002.

Dayan P. and Balleine B. W., Reward, Motivation and Reinforcement Learning. *Neuron,* 36:285-298, 2002.

De Jong E., Non-random exploration bonuses for online reinforcement learning. In Sen Sandip (Ed.), *Collected papers from the AAAI-97 workshop on multiagent learning*, 1997.

Doya K., Complementary roles of the basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, 10(6):732-739, 2000.

Eisenberger R., Does behaviorism explain self-control? *Behavioral and Brain Sciences,* 18:125, 1995.

Eshelman L.J. and Schaffer J.D., Crossover's niche, In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 9-14, Morgan Kaufmann, San Mateo, CA, 1993.

Fantino E., The future is uncertain: Eat dessert first. *Behavioral and Brain Sciences,* 18:125-126, 1995.

Fodor, J. A., *The Modularity of Mind*. MIT Press, Cambridge, MA, 1983.

Fogarty T.C., Varying the probability of mutation in the genetic algorithm. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 104-109, Morgan Kaufmann, San Mateo, CA, 1989.

Fogel L.J., Owens A. J., and Walsh M.J., *Artificial Intelligence through Simulated Evolution*, Wiley, New York, 1966.

Frank M., Loughry B. and O'Rielly R.C., Interactions between the frontal cortex and basal ganglia in working memory: A computational model. *Cognitive Affective and Behavioral Neuroscience*, 1:137-160, 2001.

Frank R.H., *Passions within Reason: The strategic Role of the emotions*. W. H. Norton, New York, 1988.

Frank R.H., Internal commitment and efficient habit formation. *Behavioral and Brain Sciences,* 18:127, 1995.

Gabriel M. and Moore J., *Learning and computational neuroscience: Foundations of adaptive networks (edited collections)*. MIT Press, Cambridge, MA, 1990.

Gibbard A., *Wise Choices, Apt Feelings: A theory of Normative Judgment*. Oxford University Press, Oxford, 1990.

Gao Y., Huang J.X., Rong H. and Zhou Z. H., Meta-game Equilibrium for Multi-agent Reinforcement Learning. In *Australian Conference on Artificial Intelligence*, pp. 930-936, 2004.

Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

Gondek D., Greenwald A., and Hall K., QnR-Learning in Markov Games, 2001. Available at http://www.cs.brown.edu/people/amygreen/papers/qnr.ps.gz.

Greenfield S.A., *The Human Brain: A Guided Tour*. Basic Books, New York, 1997.

Greenwald A. and Hall K., Correlated-Q Learning. In *20th International Conference on Machine Learning,* pp. 242-249, Morgan Kaufman, San Francisco, 2003.

Haig D., Parental antagonism, relatedness asymmetries, and genomic imprinting, In *Proceedings of the Royal Society of London B*, 264:1657-1662, 1997.

Hamburger H., *Games as models of social phenomena*. W.H. Freeman, NY, 1979.

Harp S.A., Samad T. and Guha A., Towards the genetic synthesis of neural networks. In J.D. Schaffer (Ed.), *Proc. third Int. Conf. Genetic Algorithms and Their Applications*, pp. 360-369, Morgan Kaufmann, San Mateo, CA, 1989.

Hoch S.J. and Loewenstein G. F., Time-inconsistent preferences and consumer self-control. *Journal of Consumer Research*, 17:492-507, 1991

Holland J.H., *Adaptation in Natural and Artificial Systems*. Univ. Michigan Press, Ann Arbor, Michigan, 1975.

Holland J.H., A mathematical framework for studying learning in classifier systems. *Physica D*, 2(1-3):307-317, 1986.

Holland J.H., Genetic Algorithms. *Scientific American* 267: 66-72, 1992.

Holroyd C.B. and Coles M.G.B, The neural basis of human error processing reinforcement learning, dopamine, event related negative. *Psychological Review* 109(4):679-709, 2002.

Hu J. and Wellman M.P., Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp.242-250, Morgan Kaufman, San Francisco, 1998.

Hughes J. and Churchland P.S, My behaviour made me do it: The uncaused cause of teleological behaviorism. *Behavioral and Brain Sciences,* 18:130-131, 1995.

Jacobs R. A., Computational Studies of the Development of Functionally Specialized Neural Modules. *Trends in Cognitive Science* 3:31-38, 1999.

Jafari A., Greenwald A., Gondek D. and Ercal G., On no-regret learning, fictitious play, and Nash equilibrium. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 226-233, Morgan Kaufman, San Francisco, 2001.

Kane R., Patterns, acts, and self-control: Rachlin's theory, *Behavioral and Brain Sciences,* 18:109-159, 1995.

Kanekar S., Conceptual problems in the act-versus-pattern analysis of self-control. *Behavioral and Brain Sciences,* 18:132,1995.

Kaebling L. P., Littman M. L., and Moore A. W., Reinforcement Learning: A Survey. *Journal of AI Research,* 4:237-285, 1996.

Kitano H., Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4(4): 461-476, 1990.

Klopf A. H., Brain Function and adaptive systems - A heterostatic theory. Technical Report AFCRL-72-0164, Air Force Cambridge Research Laboratories, Bedford, MA, (1972). A summary appears In *Proceedings of the International Conference on Systems, Man, and Cybernetics*, IEEE Systems, Man and Cybernetics Society, Dallas, TX, 1974.

Klopf A.H., A comparison of natural and artificial intelligence. *SIGART Newsletter*, 53:11-13, 1975.

Klopf A.H., A neuronal model of classical conditioning, *Psychobiology*, 16:85-125, 1988.

Knutson B., Adams C.M., Fong G.W., and Hommer D., Anticipation of Increasing Monetary Reward Selectively Recruits Nucleus Accumbens, *Journal of Neuroscience*, 21:RC159:1-5, 2001.

Kochanska G., Murray K. and Harlan E.T., Effortful control in early childhood: Continuity and changes, antecedents, and implications for social development. *Developmental Psychology*, 36(2):220-232, 2000.

Kolmogorov A.N., On the representations of continuous functions of many variables by superpositions of continuous functions of one variable and addition, *Doklady Akademii Nauk*, USSR, 114(5):953-956, 1957.

Konar A., *Artificial Intelligence and Soft Computiong Behavioral and Cognitive Modeling of the Human Brain*. CRC Press LLC, 2000.

Kreps D.M., *A course in Microeconomic Theory*. Princeton University Press, 1990.

Kurkova V., Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Network*, 5(3):501-506, 1992.

Laibson D., Golden Eggs and Hyperbolic Discounting. *Quarterly Journal of Economics*, 112:443-477, 1997.

Lin C.T., Jou C. P. and Lin C. J., GA-based reinforcement learning for neural networks, *International Journal of Systems Science*, 29(3):233-247, 1998.

Littman M. L., Markov games as a framework for multi-agent reinforcement learning In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 157-163, Morgan Kaufmann, San Francisco, CA, 1994.

Littman M.L., Friend-or-foe Q-learning in General-Sum Games, In *Proceedings of the eighteenth International Conference on Machine Learning*, pp. 322-328, Morgan Kaufmann, San Francisco, CA, 2001.

Littman M.L., Dean T.L. and Kaebling L.P., On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pp. 394-402, 1995.

Liu Z., Liu A., Wang C. and Niu Z., Evolving Neural Networks using real coded genetic algorithms for multispectral image classification. *Future Generation Computer Systems,* 20:1119-1129, 2004.

Loewenstein G. F., Out of Control: Visceral influences on behavior. *Organizational Behavior and Human Decision Processes*, 65:272-292, 1996.

Maynard Smith J., *Evolution and the Theory of Games*. Cambridge University Press, UK, 1982.

Mazur J. E., An adjusting procedure for studying delayed reinforcement. In M. L. Commons, J. E. Mazur, J. A. Nevin, and H. Rachlin, (Eds.), *Quantitative analyses of behaviour: V. The effects of delay and of intervening events on reinforcement value*, pp. 55-73, Lawrence Erlbaum, Hillsdale, N.J., 1987.

McCulloch W. S. and Pitts W., A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5:115-133, 1943.

Mele A., Conceptualizing self-control, *Behavioral and Brain Sciences*, 18:136-137, 1995.

Mendel J.M., A survey of learning control systems. *ISA Transactions*, 5:297-303, 1966.

Metcalfe J. and Mischel W., A hot/cool –system analysis of delay of gratification: Dynamics of willpower. *Psychological Review*, 106(1):3-19, 1999.

Millar A. and Navarick D. J., Self control and choice in humans: effects of video game playing as a positive reinforcer. *Learning and Motivation,* 15:203-218, 1984.

Minsky M.L., *Theory of Neural-Analog Reinforcement Systems and its Application to the Brain-Model Problem*. Ph.D Thesis, Princeton University, 1954.

Minsky M.L., Steps towards artificial intelligence, In *Proceedings of the institute of radio engineers*, 49:8-30, (1961). Reprinted in E. A. Feigenbaum and J. Feldman (Eds.) *Computers and Thought*, pp. 406-450, Mcgraw-Hill, New York, 1963.

Mischel W. and Mischel H.N., Development of children's knowledge of self-control strategies. *Child Development*, 54:603-619, 1983.

Mischel W., Shoda Y and Rodriguez M., Delay of gratification in children, *Science* 244:933-938, 1989.

Montana D. and Davis L., *Training feedforward neural networks using genetic algorithms*. In *Proc. eleventh Int. Conf. Artificial Intelligence*, pp. 116-121, Morgan Kaufmann, San Mateo, CA, 1989.

Morgan C. T., King R. A., Robinson N. M., *Introduction to Psychology*. Mcgraw-Hill, Tokyo, 1979.

Moriarty D.E. and Mikkulainen D.E., Efficient Reinforcement Learning through symbiotic evolution. *Machine Learning*, 22:11-32, 1996.

Moriarty D.E., Schultz A.C. and Grefenstette J.J., Evolutionary Algorithms for Reinforcement Learning. *Journal of Artificial Intelligence Research*, 11:241-276, 1999.

Mosterin J., Overcoming addiction through abstract patterns. *Behavioral and Brain Sciences,* 18: 137-138, 1995.

Muraven M. and Baumeister R.F., Self-regulation and depletion of limited resources: Does self-control resemble a muscle? *Psychological Bulletin*, 126:247:259, 2000.

Narendra K.S. and Thathachar M. A. L., Learning automata – A survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 4:323-334, 1974.

Nash J.F., Equilibrium Points in N-person Games. In *Proceedings of the National Academy of Sciences of the United States of America* 36, pp. 48-49, 1950a.

Nash J.F., The Bargaining Problem, *Econometrica*, 18:155-162, 1950b.

Nesse R. M., Natural Selection and the Capacity for Subjective Commitment. In R. M. Nesse (Ed.), *Evolution and the Capacity for Commitment*, pp. 1-44, Russell Sage, New York, 2001.

O'Doherty J., Deichmann R., Critchley H.D. and Dolan R.J., Neural Responses during Anticipation of a Primary Taste Reward. *Neuron*, 33(5):815-826 2002.

O'Reilly R.C. and Munakata Y., *Computational Explorations in Cognitive Neuroscience*. MIT Press, 2000.

Platt J., Social Traps. *American Psychologist*, 28:641-651, 1973.

Plaud J.J., The behavior of self-control, *Behavioral and Brain Sciences,* 18:139-140, 1995.

Pavlov, P. I., *Conditioned Reflexes*. Oxford University Press, London, 1927.

Pomerleau D.A., Knowledge-based training of artificial neural networks for autonomous robot driving. In J. Connell and S. Mahadevan (Eds.), *Robot Learning*, pp. 19-43, Kluwer Academic Publishers, Boston, 1993.

Pujol J.C. and Poli R., Evolving the topology and weights of neural networks using a dual representation. *Applied Intelligence*, 8:73-84, 1998.

Rachlin H., Self-Control: Beyond commitment, *Behavioral and Brain Sciences,* 18:109-159, 1995.

Rachlin H., *The Science of Self-Control*. Harvard University Press, MA, 2000.

Rachlin H. and Green L., Commitment, choice and self-control. *Journal of the Experimental Analysis of Behavior*, 17:15-22, 1972.

Rechenberg I., *Evolution Strategy: Optimization of Technical Systems by Principles of Biological Evolution*. Frommann-Holzboog, Stuttgart, 1973.

Richards N., Moriarty D.E. and Mikkulainen D.E., Evolving Neural Networks to play Go. *Applied Intelligence*, 8: 85-96,1998.

Riolo R.L., Survival of the Fittest Bits. *Scientific American,* 267(1):114-116, 1992.

Roth A.E., Vesna P., Okuno-Fujiwara and Zamir, Bargaining and Market Behavior in Jerusalem, Ljublijana, Pittsburgh and Tokyo: An Experimental Study. *American Economic Review,* 81(5):1068-1095, 1991.

Rubinstein A., Perfect equilibrium in a bargaining model. *Econometrica* 50(1):99-109, 1982.

Rubinstein A., Is it "Economics and Psychology"?: The case of hyperbolic discounting. *International Economic Review*, 44:1207-1216, 2003.

Rumelhart D.E., Hinton G.E. and Williams R.J., Learning internal representation by error propagation. In Rumelhart D.E. and McClelland J.L. (Eds), *Parallel distributed processing: Explorations in the microstructure of cognition*, (Vol. 1) pp. 318-362, MIT Press, Cambridge, MA, 1986.

Rummery G.A., *Problem Solving with Reinforcement Learning*. Ph.D. Thesis, Cambridge University, 1995.

Samuel A. L., Some studies in machine learning using the game of checkers, *IBM Journal on Research and Development*, 3:211-229, (1959). Reprinted in E.A. Feigenbaum and J. Feldman (Eds.), *Computers and Thought,* pp. 71-105, Mcgraw-Hill, New York, 1963.

Samuelson L. and Swinkels J.M., Information and the evolution of the utility function, *Journal of Economic Literature,* 2002.

Sandholm T. W. and Crites R. H., Multiagent reinforcement learning in the Iterated Prisoner's Dilemma, *BioSystems* 37: 147-166, 1996.

Schelling T., The ecology of micromotives, *Public Interest* 25:61-98, 1971.

Schelling T., Self-command: A new discipline, In *Choice over time*, pp. 167-176, Loewenstein G.F. and Elster J. (Eds.), Russell Sage Foundation, 1992.

Schwefel H.P., *Numerical Optimization of Computer Models*. Wiley, Chichester, UK, 1981.

Shapley L., Stochastic games. In *Proc. Natl. Acad. Sci.* USA 39:1095-1100, 1953.

Shefrin H. M. and Thaler R. H., An economic Theory of Self-Control, *The Journal of Political Economy*, 89(2):392-406, 1981.

Shoham Y., Powers R. and Grenager T., *Multi-agent Reinforcement Learning: a critical survey.* A Technical Report, Stanford University, available at http://robotics.stanford.edu/~shoham, 2003.

Smolensky P., Putting together connectionism. *Behavioral and Brain Sciences* 11:59-70, 1988.

Solnick J. W., Kannenberg C., Eckerman D.A. and Waller M. B., An experimental analysis of impulsivity and impulse control in humans. *Learning and Motivation* 1:61-77, 1980.

Sozou P. D., 2003, The Evolutionary Context of Self-Control Problems, Presented at the *Workshop on the Evolutionary Biology of Learning*, Fribourg, Switzerland, 21-22 February 2003.

Sporns O., Tononi G. and Edelman G.M., Connectivity and Complexity: the relationship between neuroanatomy and brain dynamics. *Neural Networks,* 13:909-922, 2000.

Strotz R.H., Myopia and inconsistency in dynamic utility maximization. *Review of Economic Studies*, 23:165-180, 1956.

Suri R.E. and Schultz W., A neural network model with dopamine like reinforcement signal that learns a spatial delayed response task. *Neuroscience,* 91:871-890, 1999.

Sutton R.S., Learning to predict by the method of temporal differences, *Machine Learning,* 3:9-44, 1988.

Sutton R.S. and Barto A. G., A temporal-difference model of classical conditioning, *In Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pp. 355-378, Lawrence Erlbaum, Hillsdale, NJ, 1987.

Sutton R. S. and Barto A. G., *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Tan M., Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international Conference on Machine learning*, pp. 330-337, Morgan Kaufmann, Amherst, MA, 1993.

Tesauro G., Neurogammon wins computer Olympiad. *Neural Computation,* 1:321-323, 1989.

Tesauro G., TD-Gammon, a Self-Teaching Backgammon Program Achieves Master-Level Play. *Neural Computation,* 6:215-219, 1994.

Tesauro G., Programming backgammon using self-teaching neural nets. *Artificial Intelligence,* 134:181-199, 2002.

Thaler R.H. and Shefrin H.M., An Economic Theory of Self-control. *The Journal of Political Economy*, 89(2):392-406, 1981.

Thorndike E.L., *Animal Intelligence*, Hafner, Darien, Connecticut, 1911.

Trivers, R., The Elements of a Scientific Theory of Self-Deception. *Annals of the New York Academy of Sciences,* 907:114-131, 2000.

Trivers R. and Burt A., Kinship and genomic imprinting. In R. Ohlsson, (Ed.), *Genomic Imprinting, An Interdisciplinary Approach,* pp. 1-23, Springer, Heidelberg, Germany, 1999.

van der Wal J., Stochastic dynamic programming. In *Mathematical centre tracts*, 139, Morgan Kaufmann, Amsterdam, 1981.

von Neumann J. and Morgenstern, *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, 1944.

Waltz M.D. and Fu K. S., A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control,* 10:390-398, 1965

Watkins C.J.C.H, *Learning from Delayed Rewards.* Ph.D Thesis, Cambridge University, 1989.

Werbos P.J., *The Roots of Backpropagation*. John Wiley and Sons Inc., New York, 1994.

Widrow B., Gupta N. K. and Maitra S., Punish/reward: Learning with a critic in Adaptive Threshold Systems. *IEEE Trans. on Sys., Man and Cyber.*, 5:455-465, 1973.

Widrow B. and Hoff J. M. E., *Adaptive switching circuits*. IRE WESCON Convention Record, pp. 961-1104, 1960.

Yao X., Evolving Artificial Neural Networks. In *Proc. of the IEEE* 87( 9):1423-1447, 1999.

Yao X. and Shi Y, A preliminary study on designing artificial neural networks using co-evolution, In *Proc. IEEE Singapore Int. Conf. Intelligent Control and Instrumentation*, Singapore, pp. 149-154, 1995.

Zornetzer S.F., Davis J.L. and Lau C., *An introduction to neural and electronic networks edited collection*. 2nd edn., Academic Press, New York, 1994.

## Candidate's Publications During the PhD Research

Banfield G. and Christodoulou C., 2003, On Reinforcement Learning in two player real-world games, In Proc. ICCS ASCS Int. Conf. on Cognitive Science, 22 .

Banfield, G. and Christodoulou, C. 2005, Can Self-Control be Explained through Games? In A. Cangelosi, G. Bugmann, R. Borisyuk (Eds), *Modelling Language, Cognition and Action*, Progress in Neural Processing, World Scientific, 16, 321-330.

## Candidate's Invited Presentations During the PhD Research

School of Computer Science and Information Systems Research seminar, Birkbeck College, University of London, Jan. 2003

Computational modelling group, Centre for Brain & Cognitive Development School of Psychology, Birkbeck College, University of London, June 2003

LSE UCL 30 Gordon St ELSE Seminar room, Feb 2004