Master Thesis

Stochastic Games with Frequency Dependent Stage Payoffs

W Mahohoma

Master Thesis DKE 14-21

Thesis submitted in partial fulfillment
of the requirements for the degree of Master of Science
of Operations Research at the Department of Knowledge
Engineering of the Maastricht University

Thesis Committee:

Dr. G.M. Schoenmakers
Dr. J. Derks

Maastricht University
Faculty of Humanities and Sciences
Department of Knowledge Engineering
Master Operations Research

August, 2014

Acknowledgements

Foremost, I would like to thank Dr G.M. Schoenmakers, my thesis supervisor, whose continuous advice and support were of great importance during the entire project. Without his patience, motivation and dedication it would have been difficult to complete this project. His guidance was of great importance throughout the research and I cannot imagine having a better supervisor than him. Special thanks also go to the University of Maastricht, Department of Knowledge Engineering, staff and students, for creating a great studying environment. Furthermore, I thank my friends who supported and assisted in many ways throughout the research, amongst them Shamiso Mahohoma, Dr. J.A. Sellin, Themba Maphenduka and Whitlaw Mugwiji. Lastly I express sincere gratitude to NUFFIC without whose support,through the Libertas Scholarship Fund I could not have pursued both my bachelors and post graduate studies at Maastricht University.

Dedication

Dedicated to Shamiso,Wenyasha and Wendy.

Abstract

In this thesis a new game model, Stochastic Games with Function Dependent Stage Payoffs, is developed by combining Stochastic Games and Games with Frequency Dependent Stage Payoffs so as to incorporate the effects of externalities arising from players' choices. To illustrate how the model can be applied to existing games, three game models namely the Prisoners' Dilemma, Small Fish War and the Battle of the Sexes are used. For each of the examples three items are investigated using the limiting average rewards criteria: 1. effect of memory length on average payoffs, 2. threat strategies and threat points for each game, 3. sets of feasible payoffs. The investigations will be based on stationary and pure cyclic strategies.

Results show that average payoffs for the Battle of the Sexes generally increase with increasing memory length whilst for the Prisoners' Dilemma and Small Fish War there is no apparent clear pattern. Threat strategies for players in the Battle of the Sexes game involved playing the preferred action with very high probabilities in both states. For Prisoners' Dilemma and Small Fish War games threat strategies involved choosing action 1, "defecting" and fishing "without" respectively, with probability 1 for all memory lengths. Feasible payoffs regions for Small Fish War and Prisoners' Dilemma games are similar to each other for all memory lengths and strategy types. For the Battle of the Sexes game the feasible payoffs region change for both cyclic and stationary strategies with increasing memory length. In the Prisoners' Dilemma game the best scenario is for the players to purely use action "cooperate". For the Small Fish War the best compromise is for both players to fish "with restraint". The best payoffs for the Battle of the Sexes game are obtained when the two players coordinate to attend the two events equally using synchronised cyclic strategies.

Contents

1. Introduction

1.1 Background

Game theory provides different models by which real life situations in which competing agents make decisions repeatedly over time are replicated and analysed. The common objective of agents is primarily to maximise their average rewards associated with the decisions they make at each decision moment. One such model is the ordinary repeated game model, which uses a single n-matrix game to represent the rewards at each decision moment. The key assumption made in this model is that the stage payoffs remain fixed for the entirety of the game. However, empirical evidence has shown that due to externalities generated by players' choices, the payoffs fluctuate between stages [6, 5, 10]. Consequently, Joosten et. al.[6] proposed a new class of games -Games with Frequency-dependent stage payoffs- to incorporate fluctuations arising from such externalities. *Frequency-dependent (FD) games are infinitely repeated non-cooperative games in which the stage payoffs are dependent on the choices of the players at that stage, as well as on the relative frequencies with which all actions were chosen by the players at previous stages.*

The novelty of this approach is that the stage payoffs are determined by the relative frequencies of past actions and the action chosen at the current stage. To justify their approach, the authors argued that players' choices at each stage generate externalities that accumulate as the play continues and these externalities have some effect on the expected payoffs of future stages. For example, if we consider the littering game, the more the players choose to litter, the more the overall payoffs decrease because of environment degradation. The same can be said if one considers the small fish War game model, if players choose to fish "without restraint" continuously then future harvests will decrease due to dwindling fish population[3, 7, 1, 8].

Whilst this approach was an improvement towards a more realistic model, inherent weaknesses in the model still exist. Amongst other things, FD games do not capture the combined sharp effects of externalities and other natural phenomena which are stochastic in nature. The approach in FD-games assumes rather smooth and slow fluctuations of payoffs resulting only from the accumulative effects of externalities arising from the players' choices. This is not in agreement with observed patterns in real life settings [9, 7] . For instance, if one considers the exploitation of common pool problems, *"one striking observation in real life renewable resource systems is that the resource can be brought down in numbers (or quantity) rather quickly by over exploitation"* [7]. The same observation was made by Myers et. al. who observed that the salmon fish population dynamics of British Columbia and Alaska river systems follow a stochastic nature with recurrent high and low peaks [2, 9]. The implication being *"highly variable harvests with one year's harvest being a hundred times larger –or smaller- than the previous year's"*[2]. Furthermore the FD method of adjusting the stage payoffs is simplistic and problematic in that it fails to consider the sequence in which the actions are chosen. For instance fishing "without restraint" n consecutive times clearly has more drastic implications than if the agents fish "without restraint" same number of times whilst fishing "with" in between over a specified time period. As such an alternative would be to consider stochastic games which by design represent the levels of expected rewards by finite bi-matrix games.

The original stochastic games as proposed by Lloyd Shapley in the early 1950's [11] require some adjustments to capture the effect of players' action externalities on the expected payoffs. Instead of adjusting the payoffs in each state, Joosten and Meijboom [7] chose to reflect the combined effects of players' choices and natural factors in the transition probabilities between states. The project considers three variants of randomness using different mechanisms to set and update transition probabilities between states. The first version involves transition probabilities which do not depend on the action played by agents but the state. This approach captures the influence of natural phenomenon and other random events not involving the players actions on the stage payoffs. In addition to capturing natural phenomenon like in the first variant, the second variant also includes the agents' actions when modelling the transition probabilities. The third variant uses game history, current state and action pairs chosen when modelling and adjusting the transition probabilities. Whilst the stochastic games with endogenous transitions capture the high and low peaks observed in real life situations as well as reflecting the combined effect of the players' choices and natural phenomena in the transition probabilities, the model fails to capture the local oscillatory behaviour of expected payoffs in each state. It assumes that rewards are constant in each state which is rather unlikely, as argued by Joosten et. al [6] and implied by Cipra[2].

Given the importance of Game Theory in informing decision making as argued by McKevely[2], it remains important to improve existing models towards being more realistic ones. McKevey argues that *"whilst the analysis from Game Theory offers a simplified version of the real world, the story it tells when treated with caution can be instructive and compelling"*[2]. This will inform and educate policymakers dealing with similar situations to come up with sound and effective policies. Whilst there are several ways of improving proposed solutions, one way is to combine related models and borrowing the strengths from each model to come up with more realistic models. This project proposes to combine the two approaches, stochastic games with endogenous transitions and FD-games, thereby creating a new game model called Stochastic Games with function dependent payoffs. The project will however consider only the case of semi-endogenous transitions. This is the second variant of the proposed game setting in the stochastic games with endogenous transitions paper [7]. This is done so as to strike a balance between the real life systems and computational complexities arising from notation and representations. The new game model will be applied to three game types namely The Prisoners' Dilemma, Battle of the Sexes and Small Fish

War.

1.2 Study Objectives

1. Develop a game model for stochastic games with function dependent stage payoffs for two players and two states.

2. Apply the model to Prisoners' Dilemma, Battle of the Sexes and Small Fish War game models.

3. Establish feasible rewards for each of the three games using the limiting average criteria.

4. Determine equilibria for the three game classes -Prisoners' Dilemma,Battle of the Sexes, Small Fish War- under the model developed in this project.

1.3 Significance of Study

This work presents a novel approach for modelling scenarios where agents involved have to make decisions repeatedly and the rewards associated with such decisions are stochastic in nature. The model proposed in this project captures the strengths of two previously proposed models, stochastic games with semi-endogenous transitions and games with frequency dependent payoff. As such it is an improvement towards the understanding of the modelled real life scenarios. This approach contributes new knowledge to the existing knowledge base in game theory and provides an alternative approach for future researchers investigating game models considered in this project. More importantly, amongst other several applications, the model can be used to inform decision and policy making for related real life situations. Using models from the field with varying approaches, game theorists have arrived at many informative conclusions and recommendations for decision makers [2, 1]. Additionally, research done in the filed has proved to be of great importance in understanding policies used in the real world modelled situations. For instance McKevely speaks of the importance of the game theory models when drafting agreements between competing parties in situations like the fish War between U.S and Canada concerning the Pacific salmon[2].

1.4 Methodology

The methodology of the project is divided into two phases, the first phase employs literature review to develop the game model. The second phase is based on simulations, which aim at assessing and testing the model. The simulations will be done in MATLAB® to establish sets of feasible rewards, threat points and threat strategies using the limiting average payoff criteria.

1.5 Organization of Study

In Chapter 2 the model under consideration is developed by combining two previously developed models, Stochastic games and FD-games. Chapter 3 deals with the concept of strategies and explains the criteria which will be used to analyse the model. Illustrations of the model on three game classes namely Prisoners' Dilemma, Small Fish War and the Battle of the Sexes are given in Chapter 4. In Chapter 5 the results obtained during the code development, code testing and simulations are presented together with an analysis and formalisations of the results. The last chapter concludes and makes recommendations on future research based on the findings of the paper.

2. The Model

2.1 Introduction

In this chapter a new game model in which the stage payoffs are adjusted using some mechanism related to the players choices at each stage is developed. The model is a combination of two earlier models FD-games and Stochastic games. In section 2.2 we define the model and in section 2.3 we explain the factors which influence the payoffs adjustment policies in the three game models investigated in this project.

2.2 The Model

The model under consideration is a 2-state stochastic game with a state space $S := \{1, 2\}$ played by two players $N = \{1, 2\}$. As such each state s is associated with a reward bi-matrix M_s where each entry (i, j) is defined as:

$$M_s(t, i, j) = R'_s(i, j) + c_s * f(\rho_{t-1}, \varrho_{t-1})$$

Here $R'_{ij} = (r^1_s(i, j), r^2_s(i, j))$ are the payoffs quantities which are used in a normal stochastic game. The scalar c_s is an adjustment coefficient associated with state s. The third component,
$f(\rho_t, \varrho_t)$, is some function which is used to adjust the stage payoffs using variables ρ_t and ϱ_t. At stage t, the relative frequency vector $\rho_t = (\rho^1_t, \rho^2_t)$ captures the relative frequency by which player k has played action 1 up to stage t. The matrices ϱ^s_t stores relative frequencies of action pairs $\{i, j\}$ in state s for the t stages. The relative frequency vector ,ρ^i_t, will be used for the Prisoners' Dilemma and Small Fish War games whilst the action pair relative frequencies will be used to adjust the payoffs in the Battle of the Sexes game.

In addition to the payoffs matrices, there are associated probability matrices, P_s for $s \in \{1, 2\}$, with $P_s(i, j)$ being the transition probability distribution over the 2 states if action pair $\{i, j\}$ is played in state s. Since we only consider two states games, the transition probability matrices are reduced to have only one entry, $p_s(i, j)$, showing the probability of the game moving to state 1. At each stage, both players influence the course of the game system by the actions they choose from the available finite set of actions with the players guided by their play rules called strategies. In this project we assume that the number of actions in all states is constant and fixed at two. Then $A^1(s) = \{1, 2\}$ and $A^2(s) = \{1, 2\}$ denote the actions sets for state $s \in \{1, 2\}$ for players 1 and 2 respectively.

If at stage t the game is in state s and player 1 chooses action i and player 2 chooses action j (this is done simultaneously and independently of one another with the full knowledge of the other player's past play), 2 things happen consecutively:

1. Player one earns the immediate payoff $M_s^1(i, j)$ and player 2 earns immediate payoff $M_s^2(i, j)$.

$$M_s(t) = \begin{bmatrix} R'_s(1,1) + c_s * f(\rho_{t-1}, \varrho_{t-1}) & R'_s(1,2) + c_s * f(\rho_{t-1}, \varrho_{t-1}) \\ \\ R'_s(2,1) + c_s * f(\rho_{t-1}, \varrho_{t-1}) & R'_s(2,2) + c_s * f(\rho_{t-1}, \varrho_{t-1}) \end{bmatrix}$$

2. The state at the next stage is determined in a stochastic manner using entry $\{i, j\}$ in P_s i.e the game will transit to state 1 with probability $p_s(i, j)$ and to state 2 with probability $1 - p_s(i, j)$.

$$P_s = \begin{bmatrix} p_s(1,1) & p_s(1,2) \\ \\ p_s(2,1) & p_s(2,2) \end{bmatrix}$$

It is assumed that the players wish to maximise their average rewards and that they have complete information with regards to the current state of the game and expected payoffs for the current stage. It is also assumed that both the players have perfect information of the game's history. The strategies for player 1 and 2 are denoted by π and σ respectively.

2.3 Stage Payoffs Adjustment

The mechanism of payoff adjustment forms the basis of this game model. An entry point into developing a sound adjustment policy should observe that in addition to influencing the state in which the game transits to in the next stage, the choices of the players also generate externalities resulting in fluctuations in the payoffs of the next stages[2, 6]. As such, a way to adjust the payoffs would be to consider the relative frequencies of actions or action pairs for both players depending on the type of the game under consideration.Whilst the adjustment policy will differ from game to game depending on how the rewards are related to the frequencies, the policies should seek to reward 'good' behavior and punish 'bad' behavior. Other game types require additional parameters for effective adjustment policies, for instance the Battle of the Sexes game model is sensitive to time. Examples of how adjustment policies can be developed for games are illustrated in chapter 4.

One other aspect related to the payoff adjustment is how much information from the past play is used to adjust the payoffs. This approach is premised on the assumption that when modelling natural phenomena, e.g. fish population dynamics and weather, a limited history span can fully model the current state of the observed phenomena. For example Myers et. al.[9] found out that the salmon fish of the British Columbia can be modelled using four year cycles, i.e. in order to model the current population information from the previous four decision moments is adequate. Thus there is need to investigate if either the entire memory or only a part is necessary to fully replicate the real life systems dynamics. In this thesis both two approaches will be investigated, when using only part of the history to update the payoffs the history span will ne referred to as 'memory length'.

Several issues are problematic with this approach of updating the payoffs, the most obvious being the failure of the adjustment policy to take into consideration the order in which the actions were chosen. For example if we consider the Small Fish War game, continuously choosing to fish "without restraint" will not only deplete the fish population but will also most likely lead to the complete destruction of the fishery. The same applies to the Battle of the Sexes game, the continuous appearance of partners at different venues will most likely lead towards a collapse of the relationship. There are several ways which complications arising from this weakness can be addressed. One way used by Joosten et. al. [6] is to also adjust the transition probabilities using the relative frequencies by which actions are chosen. Alternatively one can also include an extra absorbing state which signals the collapse of the game system. In this project we will investigate the model without addressing this issue.

3. Strategies

3.1 Introduction

Stochastic games, like repeated games are not one shot games where the players choose an action once; instead this is a game where the players involved choose actions repeatedly. Naturally, players approach the game with a plan to fulfill some defined objective. Such plans are known as strategies and are a key concept in game theory. In addition, strategies allow the analysis of the game in terms of expected payoffs. This chapter explains the concept of strategies in general and explains in depth strategies employed in this research. An average reward criteria which will be used to analyse the game is explained under section 3.2. Threat strategies, which are used by players to ensure that their opponents are cooperating, are also introduced under section 3.3.

3.2 Strategies

In Filar and Vrieze[4] a player's strategy is a specification of a probability distribution, at each stage and state, over the available actions, conditional on the history of the play. A history of the game shows the states which the game visited and the actions which the players chose in the previous stages:

$$h_t = (s_0, a_0^1, a_0^2, s_1, a_1^1, a_1^2, ..., s_t)$$

The triplet (s_t, a_t^1, a_t^2) captures the state, the action played by player 1 and the action played by player 2 at the decision moment t respectively.

Conventionally, strategies for player 1 are denoted by π and strategies for player 2 by σ. Strategies can be classified into different categories basing on how the strategies are constructed. Filar and Vrieze[4] give three classes of strategies, namely behavior strategies, Markov strategies and stationary strategies. Behavior strategies are strategies which specify a randomization over the available actions as a function of the history. Behavior strategies can be represented by a sequence $\pi = (f_0, f_1, f_3, ...)$ were for each $t = 0, 1, 2, ...$, the decision rule f_t is a randomization over the actions as a function of the history.

For Markov strategies the decision rule f_t for every decision moment $t = \{0, 1, 2, 3..\}$ is determined by the current state s_t and the moment t. A stationary strategy is a

Markov strategy where for every t the decision rule f_t is completely determined by the current state. As such a stationary strategy for player k is be denoted by:

$$\pi^k = (f, f, f, ...)$$

where $f = (f(1), f(2))$ specifies for each state $s \in \{1, 2\}$ a probability vector $f(s)$ over the action set in that state $A^i(s)$. In other words a strategy is stationary if for each state it specifies a fixed probability vector for each state whenever that state is being visited. A stationary strategy is called pure if prescribes for every state and every possible history one action to be played with probability 1. Stationary Strategies that are not pure are called mixed strategies. The Markov strategies used in this project are pure cyclic and stationary strategies (both pure and mixed). Cyclic strategies are strategies in which a player plays a repeated sequence of actions. Both the stationary and cyclic strategies used in this thesis are state specific.

At each stage and according to the strategy pair (π, σ) each player receives a payoff. If we use $R_t^k(\pi, \sigma)$ to represent the expected stage payoff at stage t for player k given by strategy pair (π, σ), then (π, σ) determines a stochastic process on the stage payoffs. If the game is played for an infinite number of stages the limiting average reward for player k is given by:

$$\gamma^k(\pi, \sigma) = E_{\pi\sigma}(\liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} R_t^k) \tag{3.1}$$

We will assume that the goal of the players is to maximise their limiting average rewards. For some strategy pairs, (π', σ'), no player has an incentive to deviate from his or her chosen strategy after considering an opponent's choice as this will not improve the players expected average payoffs. The strategy pair (π', σ') is an equilibrium, i.e.:

$$\gamma^1(\pi', \sigma') \geq \gamma^1(\pi, \sigma') \forall \pi \tag{3.2}$$

$$\gamma^2(\pi', \sigma') \geq \gamma^2(\pi', \sigma) \forall \sigma \tag{3.3}$$

In other words, an individual will not receive an incremental benefit if he deviates from his strategy assuming that the other player does not change their strategy.

3.3 Threat Strategies

Players often need to incorporate mechanisms of punishing the other player in case of deviation from initial strategies. A way to come up with such strategies is to consider a situation were a player is trying to maximise his reward whilst his opponent is minimising that reward. For the minimising player there exist a strategy, $\pi^\star$ (for player 1) or $\sigma^\star$ (for player 2), such that the player who is trying to maximise (the one being punished) will not get a payoff above a certain value regardless of the strategy he employs. These strategies, $\pi^\star$ and $\sigma^\star$, are called retaliation or threat strategies and the associated payoffs (v_1, v_2) is called the threat or retaliation point.

$$\gamma^1(\pi, \sigma^\star) \leq v_1 \forall \pi \tag{3.4}$$
$$\gamma^2(\pi^\star, \sigma) \geq v_2 \forall \sigma \tag{3.5}$$

Thus if either of the two players deviate from an equilibrium the opponent can always invoke the threat strategy thereby ensuring that the player will not get any reward above the threat point. This in essence forces players to stick to their strategy allowing all feasible rewards above the threat point to be obtained as equilibrium rewards.

4. Application Examples

4.1 Introduction

The model proposed in this thesis is applicable to many situations modelled by ordinary repeated and stochastic games. Notable examples are the Small Fish War, Prisoners' Dilemma and the Battle of the Sexes game models. In each type of game, the two states are defined in such a way that the level of rewards for action pairs in *state 1* are higher when compared to corresponding action pairs in *state 2*. The following subsections explain how the proposed game settings can be interpreted and adapted using the following type of games: Small Fish War, Battle of the Sexes and Prisoners' Dilemma. The final section explains under what circumstances the model can be applied to other game models.

4.2 Small Fish War

A Small Fish War game replicates a situation where two agents own fishing rights to a fishery. Under the model considered in this project, the fish population at any stage is either high *(State 1)* or low *(State 2)*. At each stage the agents have two options, either to fish"without restraint" *(Action 1)* or "with restraint" *(Action 2)*. Action "with restraint" refers to the action for which the agents employ resource friendly exploiting methods which maintains the viability of the resource by allowing the resource to recover e.g. limiting the quantities harvested per each fishing cycle or using sparse nets. The action "without restraint" is when the agents are maximising their immediate earnings without paying concern to the fish population e.g using fine mazed fishing nets and not limiting quantities harvested. Clearly the action "without restraint" is dominant and drives the fish population downwards whereas the action "with restraint" allows the resource to recover.

$$\begin{array}{cc} & \begin{array}{cc} \text{Action 1} & \text{Action 2} \end{array} \\ \begin{array}{c} \text{Action 1} \\ \\ \text{Action 2} \end{array} & \begin{bmatrix} \frac{11}{2}, \frac{11}{2} & 6, \frac{7}{2} \\ & \\ \frac{7}{2}, 6 & 4, 4 \end{bmatrix} \\ & \text{State 1} \end{array}$$

$$\begin{array}{ccc} & \text{Action 1} & \text{Action 2} \\ \text{Action 1} & \left[\begin{matrix} \frac{11}{4}, \frac{11}{4} \\ \\ \frac{7}{4}, 3 \end{matrix}\right. & \left.\begin{matrix} 3, \frac{7}{4} \\ \\ 2, 2 \end{matrix}\right] \\ \text{Action 2} & & \end{array}$$

State 2

The above setup assumes that the payoffs in each state are fixed thus the agents are assured of a certain quantity whenever they use the same action in some state. However empirical evidence [6] has shown that this is not the case. The fish population as shown by the empirical study conducted by Myers et. al.[9] is not constant but rather dynamic. Thus besides the huge jumps between the highs and lows (states) we should also introduce mechanisms to replicate the fluctuations within states. One way is to observe that fishing "without restraint" results in nondiscriminatory catching of fish including those of the reproductive age which are supposed to replenish the fish population [7]. It follows that fish population is negatively related to the continuous use of action 1 and thus we can choose to lower the stage payoffs by some factor multiplied by the relative frequencies by which players have used action "without restraint". Thus if at stage t we use the relative frequency vector ρ_t^k to represent the relative frequencies for which player k has played action 1 then the stage payoffs for stage $t+1$ are denoted by:

$$M_s(t+1) = \begin{bmatrix} R'_{1,1} - c_s * (\rho_t^1 + \rho_t^2) & R'_{1,2} - c_s * (\rho_t^1 + \rho_t^2) \\ \\ R'_{2,1} - c_s * (\rho_t^1 + \rho_t^2) & R'_{2,2} - c_s * (\rho_t^1 + \rho_t^2) \end{bmatrix}$$

Restrictions on the payoffs can be imposed after observing that action 1 dominates action 2 and that by design the corresponding payoffs in state 2 are less than those in state 1. If $r_s^k(i,j)$ is the payoff for player k when player 1 is playing action i and player 2 is playing action j in state s then:

$$r_s^1(1,1) \geq r_s^1(2,1)\ ,\ r_s^1(1,2) \geq r_s^1(2,2) \text{ and } r_1^1(i,j) \geq r_2^1(i,j) \tag{4.1}$$

$$r_s^2(1,1) \geq r_s^2(1,2)\ ,\ r_s^2(2,1) \geq r_s^2(2,2) \text{ and } r_1^2(i,j) \geq r_2^2(i,j) \tag{4.2}$$

To have a meaningful model, we should also impose restrictions on the transition probabilities considering the implications of each action pair within states as argued for by Joosten et. al. [7]. We further assume symmetry on the players actions impact on the resource and observe that both players fishing "without restraint" is worse than if one player uses this action. Furthermore one player using action 1 (fishing "without restraint") is clearly worse than if none of the player uses this action in any of the state. Additionally, it is logical to make the assumption that the system is more vulnerable when the fish population is low. Consequently corresponding transition probabilities in state 2 are at most equal to those in state 1. Let $p_s(i,j)$ be the probability of the game moving to state 1 given state s and action pair $\{i,j\}$, then:

$$0 < p_s(1,1) \leq p_s(1,2) = p_s(2,1) \leq p_s(2,2) < 1 \ \forall \ s \in \{1,2\} \tag{4.3}$$

$$p_1(i,j) \geq p_2(i,j) \ \forall \ i,j \tag{4.4}$$

The following example is an instance of a Small Fish War game modeled under the model developed in this project and it will also be used later in the analysis of the model in following chapters.

Example SFW1: Two agents hold fishing rights to a fishery and they have an option of fishing "with restraint" (action 1) or "without restraint" (action 2). Additionally the fish population is either high -State 1- or low -State 2-. Using the above proposed payoff adjustment regime with adjustments coefficient matrix c=[1 0.5], the stage payoffs for stage $t+1$ are captured as follows:

$$\text{Player 1} = \underset{\text{State 1}}{\begin{bmatrix} 5.50 - 1.0 * (\rho_t^1 + \rho_t^2) & 6.00 - 1.0 * (\rho_t^1 + \rho_t^2) \\ 3.50 - 1.0 * (\rho_t^1 + \rho_t^2) & 4.00 - 1.0 * (\rho_t^1 + \rho_t^2) \end{bmatrix}} = \text{Player } 2^\top$$

$$\text{Player 1} = \underset{\text{State 2}}{\begin{bmatrix} 2.75 - 0.5 * (\rho_t^1 + \rho_t^2) & 3.00 - 0.5 * (\rho_t^1 + \rho_t^2) \\ 1.75 - 0.5 * (\rho_t^1 + \rho_t^2) & 2.00 - 0.5 * (\rho_t^1 + \rho_t^2) \end{bmatrix}} = \text{Player } 2^\top$$

The coefficients c_1 and c_2 can be set arbitrarily with $c_1 > c_2$, we chose $c_1 = 2 * c_2$ to get all payoffs in state 1 twice as big as in state 2. This maybe intuitionally be justified that the lesser the population the less the players can catch hence the small the impact of their actions assuming all other variables remaining unchanged. The transition probabilities of such a game can also be set arbitrarily as long they obey the restrictions specified above. Whilst there are many possible valid combinations of the values, we use the following example to demonstrate one such instant of the combinations. Good combinations should however replicate the real dynamics of the system and this is not in the scope of the thesis. Simulations will also be used to investigate if the values of transition probabilities have any bearing on the results obtained.

$$\underset{\text{State 1}}{\begin{bmatrix} 0.3 & 0.6 \\ 0.6 & 0.8 \end{bmatrix}} \quad \underset{\text{State 2}}{\begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}}$$

4.3 Battle of the Sexes

The classical Battle of the Sexes is a two-player coordination game which can be fully described by the following scenario. Suppose a couple that agree to meet in evening, but cannot remember if they are to go to the movie or a football match. The wife prefers going to a movie more than going to watch a football match. The husband on the other hand prefers watching football more than going to watch a movie. However the best situation for both partners is to go to the same event together. Then the situation can be described by the following bi-matrix game where the wife chooses the row and the husband chooses the column:

$$\begin{array}{cc} & \begin{array}{cc} \text{Movie} & \text{Football} \end{array} \\ \begin{array}{c} \text{Movie} \\ \text{Football} \end{array} & \begin{bmatrix} 7,5 & 4,4 \\ 2,2 & 5,7 \end{bmatrix} \end{array}$$

A two state stochastic Battle of the Sexes game arises from the repeated single state game if we take into consideration the mood in the relationship at each decision moment. We use two states to capture the mood level in the relationship with state 1 being the situation when the mood is high and state 2 representing a low mood. Rewards for state 2 are obtained by multiplying those of state 1 by a fraction. A fraction of 0.5 is the one used in the project unless otherwise stated.

It is important to note that the desirable behavior is defined by action pairs, not individual actions, i.e. it will be desirable for the players to attend the same event. We can impose restrictions on the transition probabilities by observing that appearing at the same event is more desirable and drives the system towards the high state whereas appearing at different events does the opposite. If we assume symmetry, i.e. both players derive the same rewards in the preferred events and vice versa then the following inequalities hold for the transition probabilities:

$$1 > p_s(1,1) = p_s(2,2) \geq p_s(1,2) = p_s(2,1) > 0 \;\forall\; s \in \{1,2\} \tag{4.5}$$

$$p_1(i,j) \geq p_2(i,j) \;\forall\; i,j \tag{4.6}$$

With regards to adjusting the payoffs we can note the following about the players' action pairs in relation to the expected payoffs:

- Attending events together for the couple is the best scenario, however this should be balanced between the two events. Thus we can choose to look at the frequencies by which the partners attend the same event together and adjust by reducing over attendance of one event. The reduction takes care of possible disgruntlements from one partner whose preferred event is under-attended:

$$e_t = \sum_{s=1}^{2} \mid \varrho^s_{1,1}(t) - \varrho^s_{2,2}(t) \mid \tag{4.7}$$

- Attending different events (sad moments) leads to a reduction in payoffs and the reduction is inversely proportional to time, i.e attending events during the early stage of the game is more detrimental than if the same happens during the later stages of the game. As such the following quantity will be deducted from the payoffs for stage $t+1$:

$$f_t = \sum_{s=1}^{2} \frac{\varrho^s_{1,2}(t) + \varrho^s_{2,1}(t)}{\min\{t,m\}} \tag{4.8}$$

- Attending same events (happy moments) has an incremental effect on future stage payoffs. Thus with time the payoffs for both players when attending individually preferred events alone will increase as a function of previous attendance of events together. The following term will be added to rewards in entries $\{i, j\} = \{1, 2\}$ in both states:

$$g_t = \sum_{s=1}^{2} \varrho^s_{1,1}(t) + \varrho^s_{2,2}(t) \tag{4.9}$$

The three adjustment quantities are combined giving the following structure quantities which are used to update the relevant entries in the reward matrices:

$$\phi_{i,j}(t) = \begin{cases} e_t + f_t - g_t & \{i, j\} = \{1, 2\} \\ e_t + f_t & otherwise \end{cases}$$

If we incorporate quantity $\phi_{i,j}(t)$ for stage payoffs in state $s \in \{1, 2\}$, then the payoffs at stage $t + 1$ are:

$$M_s(t+1) = \begin{bmatrix} R'_s(1,1) - c_s * \phi_{i,j}(t) & R'_s(1,2) - c_s * \phi_{i,j}(t) \\ R'_s(2,1) - c_s * \phi_{i,j}(t) & R'_s(2,2) - c_s * \phi_{i,j}(t) \end{bmatrix}$$

The above setup is demonstrated in the following example which will also be used in the following chapters.

Example BS1:A couple has to decide where to go on a date with the preference from the two being to spend the evening together. However the man prefers watching football whilst the woman prefers going to watch a movie. If the payoff adjustment is considered using c=[1 0.5] the payoffs at stage $t + 1$ are:

$$M_1^1(t+1) = \begin{bmatrix} 7.0 - 1.0 * \phi_{1,1}(t) & 4.0 - 1.0 * \phi_{1,2}(t) \\ 2.0 - 1.0 * \phi_{2,1}(t) & 5.0 - 1.0 * \phi_{2,2}(t) \end{bmatrix}$$

$$M_1^2(t+1) = \begin{bmatrix} 5.0 - 1.0 * \phi_{1,1}(t) & 4.0 - 1.0 * \phi_{1,2}(t) \\ 2.0 - 1.0 * \phi_{2,1}(t) & 7.0 - 1.0 * \phi_{2,2}(t) \end{bmatrix}$$

$$M_2^1(t+1) = \begin{bmatrix} 3.5 - 0.5 * \phi_{1,1}(t) & 2.0 - 0.5 * \phi_{1,2}(t) \\ 1.0 - 0.5 * \phi_{2,1}(t) & 2.5 - 0.5 * \phi_{2,2}(t) \end{bmatrix}$$

$$M_2^2(t+1) = \begin{bmatrix} 2.5 - 0.5 * \phi_{1,1}(t) & 2.0 - 0.5 * \phi_{1,2}(t) \\ 1.0 - 0.5 * \phi_{2,1}(t) & 3.5 - 0.5 * \phi_{2,2}(t) \end{bmatrix}$$

The accompanying transition probabilities which are arbitrarily set to obey the above proposed restrictions are as follows:

$$\underset{P_1}{\begin{bmatrix} 0.8 & 0.4 \\ \\ 0.4 & 0.8 \end{bmatrix}} \quad \underset{P_2}{\begin{bmatrix} 0.6 & 0.2 \\ \\ 0.2 & 0.6 \end{bmatrix}}$$

4.4 Prisoners' Dilemma

In the original one state repeated Prisoners' Dilemma, two suspects have committed a crime together and are interrogated separately. Each suspect has two choices, cooperating with his partner and keeping quite (action 2) or choosing to defect and implicate his partner (action 1). If both players cooperate each suspect is convicted for a minor crime. Defecting by both players also implies a reduced sentence to both suspects which is however bigger than the sentence when they both cooperate. If one of the suspects defects while the other one cooperates then the one who defected serves a lesser sentence than the one who cooperated who in turns serves a larger sentence. The following payoff matrix reflects the given situation:

$$\begin{array}{cc} & \begin{array}{cc} \text{Defect} & \text{Cooperate} \end{array} \\ \begin{array}{c} \text{Defect} \\ \text{Cooperate} \end{array} & \begin{bmatrix} 4,4 & 9,2 \\ 2,9 & 6,6 \end{bmatrix} \end{array}$$

Using the rationale which justifies the repeated game i.e. the continued association of the suspects in future crimes, one critical observation is made with regards to the trust between the two. Cooperating reinforces the trust between the two and this leads to the duo committing more serious crimes together. On the contrary if both defect or if either of the suspects defect, the trust between the two is negatively affected and the chance for the pair committing a big crime together is less. This can be interpreted as a stochastic game with two states. State 1 is a situation where the trust level between the two is high enough for them to commit big crimes hence the higher rewards as compared to state 2. In state 2 trust level is low and the suspects have committed small crimes hence the small payoffs as compared to state 1. The stage payoffs are adjusted using relative frequencies to capture the shocks which are brought about by bad behavior, i.e. defecting by the players. Coefficients c_s will be used to differentiate the quantities deducted in state 1 and 2 the rationale being that the system responds differently when in each state.

$$M_s(t+1) = \begin{bmatrix} R'_s(1,1) + c_s * f(\rho^1_{t-1}, \rho^2_{t-1}) & R'_s(1,2) + c_s * f(\rho^1_{t-1}, \rho^2_{t-1}) \\ \\ R'_s(2,1) + c_s * f(\rho^1_{t-1}, \rho^2_{t-1}) & R'_s(2,2) + c_s * f(\rho^1_{t-1}, \rho^2_{t-1}) \end{bmatrix}$$

In order to fully describe the Prisoners' Dilemma as modelled under the stochastic game with frequency dependent payoffs model, a mechanism of setting up the transition probability needs to be set up. As in the Small Fish War above, assuming symmetry, both players cooperating means thats the game is more likely in state 1 in the next stage as the trust levels are kept high. On the contrary, defecting drives the game system towards state 2 and this is worse when it is done by both players. It also follows that the system is more vulnerable when in state 2, consequently the restrictions developed for the transition probability for the small fish also holds for this game.

Example PD1: Two suspects who are jointly charged with a crime are being interrogated by the police. Both have a chance of either to defecting, i.e confessing *(Action 1)* or cooperating, i.e. remaining silent *(Action 2)*. Taking into account the payoff adjustment for *c=[1 0.5]*, the stage payoffs for stage $t+1$ are:

$$\text{Player 1} = \underset{\text{State 1}}{\begin{bmatrix} 4.0 - 1.0 * (\rho_t^1 + \rho_t^2) & 9.0 - 1.0 * (\rho_t^1 + \rho_t^2) \\ \\ 2.0 - 1.0 * (\rho_t^1 + \rho_t^2) & 6.0 - 1.0 * (\rho_t^1 + \rho_t^2) \end{bmatrix}} = \text{Player } 2^\top$$

$$\text{Player 1} = \underset{\text{State 2}}{\begin{bmatrix} 2.0 - 0.5 * (\rho_t^1 + \rho_t^2) & 4.5 - 0.5 * (\rho_1^1 + \rho_t^2) \\ \\ 1.0 - 0.5 * (\rho_t^1 + \rho_t^2) & 3.0 - 0.5 * (\rho_1^1 + \rho_t^2) \end{bmatrix}} = \text{Player } 2^\top$$

The payoffs for state 2 can be obtained by multiplying the state 1 payoff matrix by a fraction. In this project we consider a fraction of 0.5. The accompanying transition probabilities will be as follows:

$$\underset{\text{State 1}}{\begin{bmatrix} 0.3 & 0.6 \\ \\ 0.6 & 0.8 \end{bmatrix}} \quad \underset{\text{State 2}}{\begin{bmatrix} 0.2 & 0.4 \\ \\ 0.4 & 0.6 \end{bmatrix}}$$

It should be noted that the structure of the rewards adjustments and the transition probabilities are almost identical to the Small Fish War since the game systems dynamics are essentially the same.

The above examples have demonstrated how the model introduced in chapter 2 can be used in many problems which have been investigated before. It is important at this juncture to mention that whilst the model can be applied to several other related games, this project will only investigate the above three types of games. Attempts to model other games under this model should first address how different states may arise. Secondly it is important to also establish externalities generated by the players' actions and how they are related to payoffs.

5. Simulations: Results and Analysis

5.1 Introduction

In this chapter design descriptions of the functions used to fulfill the project objectives are given. This will be done in section 5.2. Section 5.3 explains the tests done on the code and issues arising. The results obtained during the simulations are presented in three separate sections. Section 5.4 investigates the effect of memory length on stage payoffs dynamics as well as average payoffs. Section 5.5 deals with threat strategies and threat points, investigating how they relate to memory length used to update the payoffs. Section 5.6 presents the results for the structure of feasible average payoffs areas for the three game examples used in this project.

5.2 MATLAB® Functions

This subsection explains the design of MATLAB® functions which were used for the simulations of play involving stationary and cyclic strategies. Only the main files will be explained under this section. Several other files which were used to do tasks like plotting and results will not be included. The actual files and functions are attached together with this report in the Appendix.

5.2.1 Payoff Adjustment

To adjust the payoffs at each stage we use the function *PayoffAdjustor*. As input the function needs game type to determine the type of adjustment to be used. The game type is given as a string: 'PD' for Prisoners Dilemma, 'SFW' for Small Fish War and 'BS' for Battle of the Sexes. Additionally the function also needs the relative frequency of action 1 if the game type is Prisoners Dilemma or the Small Fish War,which is given as a vector with 2 entries. The first (second) entry is the relative frequency which player 1 (2) has played action 1. If the game type is Battle of the Sexes then action pairs relative frequencies are given instead. The function also needs the adjusting coefficients as input which is a vector with 2 entries. The first (second) entry adjusts payoffs in state 1 (2). As output it gives four 2×2 matrices which are used to adjust payoff for each players reward matrices.

5.2.2 Stationary Play

The stationary play routines which were designed for this project are capable of handling both pure and mixed stationary strategies. There are basically two ways in which the simulations are carried out, the first version executes a single simulation. This is done using the function *StationaryPlay*, which uses a single stationary strategy pair and calculates stage payoffs for T stages for each memory length. As stated in section 2.3 memory length refers to the number of previous stages whose history is used to update the payoffs. The strategy for each player is a 2×1 vector with the first(second) entry being the probability of playing the first action in state 1(2). As output it gives the two sets of stage payoffs, one for each player. For each player the payoffs are in a $n_m \times T$ matrix with n_m being the number of different memory lengths used to adjust the payoffs. In this project we will use the following memory lengths m=[1 2 5 10 20 50 75 ∞], so in this case $n_m = 8$. The second version, *StaSimulate*, involves an exhaustive iteration through input supplied sets of strategies for each player with the first method executed at each simulation with a strategy pair derived from the strategy sets. The strategy sets are created by generating a row vector of n_k^s points from a given strategy range, $range_k^s$ for player k state in s. The range is an interval between 0 and 1.

5.2.3 Cyclic Play

Play involving pure cyclic strategies have two versions similar to those of stationary play with the difference being the structure of strategies used. The cyclic strategies considered in this project are state specific. Consequently, the one simulation version,*CyclicPlay* function, of cyclic play has as part of input two $2 \times l_k$ matrices,π and σ, one for each player's strategy with l_k being the length of players k's strategy. The first(second) row specifies actions to be played if the game is in state 1(2). This version gives as output a series of stage payoffs and average payoffs for both players. The second version outputs two sets of average payoffs, one for each player, for the given strategy sets. Each set of strategies is a $r_i^s \times l_i$ matrix with r_i^s being the number of cyclic strategies for player i in state s. This routine is implemented in the function *CycSimulateR* and requires the *CyclicPlay* function.

5.3 Code Testing

The first step after developing the MATLAB® code was to test if the code was working as expected and efficiently. With regards to producing expected results, the testing was done by comparing and verifying results to those obtained from calculations done by hand. Based on several trials the code passed this test as it was producing expected results. With respect to efficiency, both the one simulation functions, *CyclePlay* and *StationaryPlay*, produced results well under a second. For the multi-simulation functions, *StatSimulate* and *CycSimulate*, efficiency with regards to time was a challenge. This is expected as both routines are based on a brute force

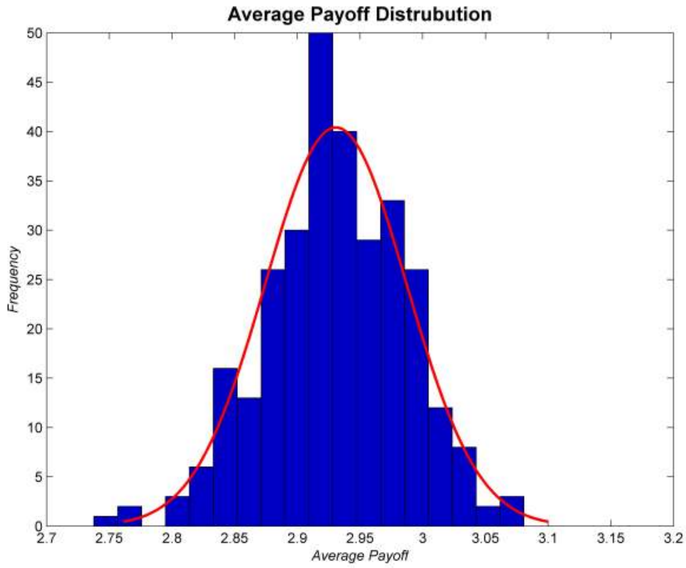
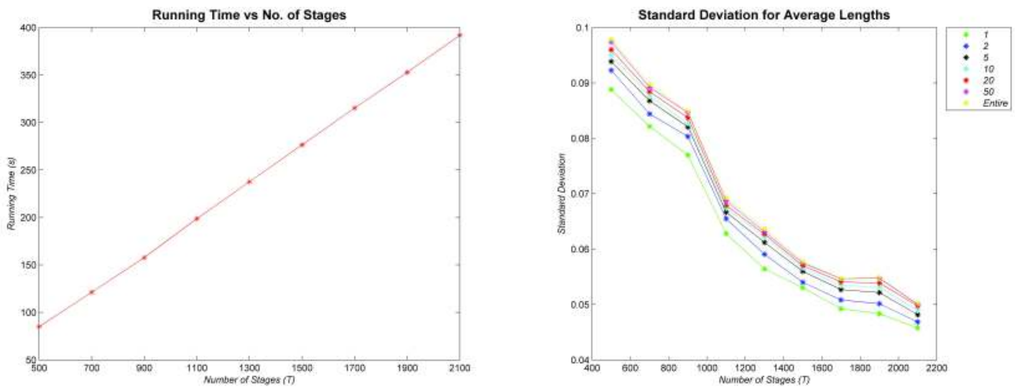


Figure 5.1: $T = 1300$, distribution for average payoffs for strategy pair $\pi = [0.7\,0.3]\sigma = [0.3\,0.7]$.

search through the provided strategy sets.

After testing, simulations were carried out in MATLAB® to address several issues amongst them: investigate effect of memory length on average expected payoffs, to establish sets of feasible average payoffs and determine threat points for each type of game.

An important observation made during the trial simulations was that for a given strategy pair the average payoff for each simulation varied in size with the difference explained by different sequences of states and/or actions generated in each simulation for the same strategy pair. This presented challenges when determining the relationship between average payoffs and memory length as well as when establishing threat points. In order to get a good approximation of the true average payoff associated with a strategy pair very large values for the number of stages, T, are supposed to be used to minimise the observed differences. However this comes with computational challenges thus there was need to balance between running time and the quality of results. Investigations were done to determine the number of stages which assured good approximates within reasonable amount of time. Consequently 300 simulations based on the same strategy pair for a fixed number of stages were done. The process was repeated for 10 different values of T and the results were analysed. The average payoffs were found to be following a normal distribution. Figure 5.1 shows the histogram of average payoffs for the strategy pair $\pi = [0.7\,0.3]\sigma = [0.3\,0.7]$ on the Battle of the Sexes example given in section 4.3 for $T = 1300$. From figure 5.2b the standard deviations decrease with increasing the number of stages. From $T = 1500$ the rate at which the standard deviation decreases sharply and the associated improvement in estimating the true average payoff with increasing T is insignificant. Figure 5.2a shows that the running time are linear with increasing T. Consequently, T was set



(a) (b)

Figure 5.2: (a)computing and (b) standard deviations of average payoffs for strategy pair $\pi = [0.7\,0.3]\sigma = [0.3\,0.7]$

at 1500 for the analysis of average payoffs and threat points. Given that even for $T = 1500$ there were still variance in the average payoff associated with a strategy pair and that the average payoffs followed a normal distribution we used the central theorem to approximate the average payoffs using a 95% confidence interval. The sample size for each strategy pair was set at 200.

Table 5.1: Standard deviations for different stages, Example BS1, $\pi = [0.7\,0.3]\sigma = [0.3\,0.7]$

T	$Time$	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
500	85.0480	0.088795	0.092271	0.093862	0.095026	0.095996	0.097384	0.097824	0.098730
700	121.2315	0.082164	0.084406	0.086780	0.087687	0.088429	0.089213	0.089776	0.090715
900	157.8010	0.076988	0.080344	0.082076	0.082688	0.083735	0.084642	0.084870	0.085975
1100	198.7120	0.062793	0.065501	0.066664	0.067404	0.067883	0.068712	0.069250	0.069735
1300	237.6486	0.056431	0.059073	0.061209	0.062195	0.062706	0.063137	0.063686	0.064583
1500	276.5111	0.053045	0.054016	0.055989	0.056541	0.057011	0.057489	0.057715	0.057656
1700	315.3298	0.049232	0.050820	0.052694	0.053449	0.054083	0.054654	0.054709	0.055595
1900	352.8572	0.048351	0.050153	0.052162	0.053169	0.053877	0.054740	0.054874	0.055421
2100	391.9091	0.045761	0.046856	0.048169	0.048956	0.049742	0.050016	0.050160	0.051220

5.4 Results: Expected Stage and Average Payoffs

In order to investigate the effect of memory length on the stage payoff adjustments, a sequence of 1500 actions and states generated for each strategy pair. This was done to remove differences arising from different sequences of actions and states of each simulation. Using this sequence, stage payoffs where then calculated for each of the memory lengths considered in this thesis. The memory lengths used are 1, 2, 5, 10, 20, 50, 75 and ∞ . The results for each game are presented below.

5.4.1 Battle of the Sexes

We use the example from section 4.3 to investigate the effect of the memory length on stage payoffs and consequently average payoffs for Battle of the Sexes games. In each simulation of 1500 stages, the payoffs obtained by each player for each memory length was recorded. To illustrate the dynamics during each simulation , the payoffs obtained by player 1 during the simulation are plotted in figures 5.3. Similar trends are observed for player 2's payoffs and more figures are found in the appendix. To give a clear picture the plot are zoomed for only 50 stages from $t = 501$ to $t = 550$. The strategy pair associated with the two plots is $\pi = [0, 0], \sigma = [1, 1]$.

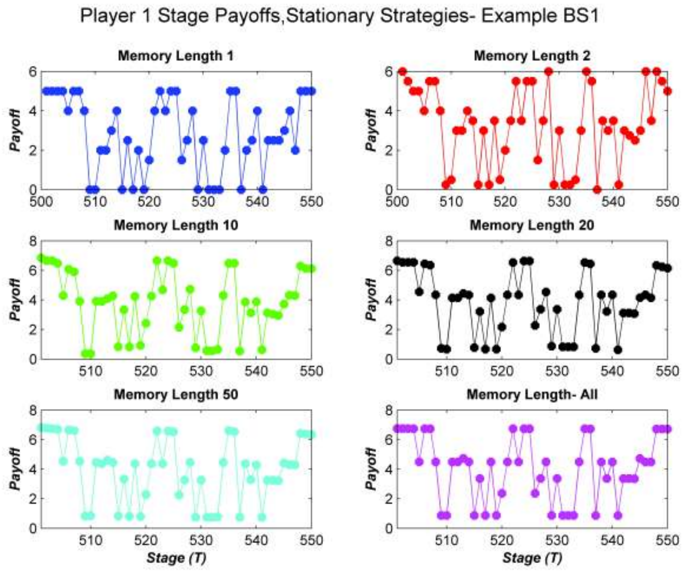


Figure 5.3: Player 1's stage payoffs for $\pi = [0, 0], \sigma = [1, 1]$

From the plot of player 1's stage payoffs, figure 5.3, it can be seen that for memory length 1, the stage payoffs have a lower bound of 0 and an upper bound of 5. For memory length of 2, fewer 0's are obtained and there are two 6's obtained which does not happen for memory length 1. For other memory lengths the main difference with memory lengths of 1 and 2, is that no 0's are obtained and also that in some stages payoffs larger than 6 are obtained. However the difference amongst themselves is not clearly evident to the naked eye. A good picture of the differences is shown taking average stage payoffs for each memory length for every strategy pair considered. The results for a selected few strategy pairs are shown in tables 5.2 and 5.3.

From tables 5.2 and 5.3 the first pattern to notice is that players receive a constant expected average reward regardless of the memory length when both employ pure stationary strategies that lead to the action pair {1,1} or {2,2} being chosen in all states at all stages . As can be seen from the top subfigure in figure 5.4 the expected average payoff based on a 95% confidence interval remains within the range 5.2486±0.082356 for player 1 when the players are using $\pi = [1.0, 1.0], \sigma = [1.0, 1.0]$. Player 2's payoff for the same strategy pair also remains fixed within the range 3.4993±0.54904 for all memory lengths as shown in the third row of table 5.2. This pattern is a result of the fact that for this particular strategy pair the quantity $\phi_{i,j}(t) = 1$ for {1,1} or {2,2} since for all memory lengths $e_t = 1$ and $f_t = 0$ as none of actions pairs [{1,2},{2,1}] are chosen at any of the stages.

Any other strategy pair leads to the general trend which suggests that increasing memory length is associated with an increase in the expected average reward for the Battle of the Sexes game. This trend is illustrated in figure 5.4 bottom subfigure. In this case the players are using the strategy pair $\pi = [0,0], \sigma = [1,1]$ yielding average payoffs lying in the range 0.7511±0.015564 for both players when memory length of one is used to adjust the stage payoffs. The average payoff increases for the same strategy pair with increasing memory length reaching a maximum within the range of 1.4968±0.031066 when using full memory to adjust the payoffs. The updating component dealing with sad moments explains this trend. For mixed strategies increasing the memory length, m, reduces this quantity as the relative frequencies are proportionally the same since there are based on the same strategy pair.

The same pattern is observed when the players are using cyclic strategy pairs. Table 5.4 displays average payoffs for a selected 9 strategy pairs. Cyclic strategy pairs in which the players use the same action for each state lead to average payoffs that remain the same with increasing memory length. This is expected as such strategies are equal to pure stationary strategy. Any other cyclic strategy pair yields average payoffs which increase with increasing memory length. A special case is when the players are using $\pi = [1212; 1212], \sigma = [1212; 1212]$, this yields expected average rewards of 4.3757±0.067399 for memory length 1 and 5.0748±0.077872 for memory 5 which are well below the average payoffs for other memory lengths. This is shown in figure 5.5 subfigure 2. For even memory lengths the average payoffs are almost equal and for odd memory lengths the average payoffs increase with increasing memory length and it approaches the payoffs for even memory length. This is because for even numbers both the reducing components, $e_t = 0$ and $f_t = 0$, and the increasing component(which is not relevant since the actions pairs being chosen are always {1,1} and {2,2}) is always 1. On the contrary, for odd memory lengths the quantity e_t decreases with increasing memory length approaching zero for bigger values of m leading to increasing average payoffs. This is because at each updating moment the quantity e_t reduces to:

$$e_t = \frac{1}{\min\{t, m\}}$$

This clearly decreases with increasing memory length approaching 0 for large values of both m and t.

Table 5.2: Player 1's average payoffs for stationary Strategies, Example BS1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[0.0,0.0]	[0.0,0.0]	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214
[1.0,1.0]	[0.0,0.0]	2.2508±0.047614	2.6252±0.055543	2.8496±0.060296	2.9242±0.061883	2.9614±0.062675	2.9834±0.063143	2.9882±0.063245	2.9962±0.063415
[1.0,1.0]	[1.0,1.0]	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356
[0.0,0.0]	[1.0,1.0]	0.7511±0.015564	1.1257±0.023342	1.3502±0.028008	1.4248±0.029560	1.4620±0.030335	1.4840±0.030796	1.4888±0.030895	1.4968±0.031066
[0.9,0.9]	[0.1,0.1]	2.7047±0.078190	3.0259±0.080685	3.2386±0.083487	3.3237±0.085615	3.3748±0.086875	3.4131±0.087846	3.4234±0.087990	3.4481±0.090151
[1.0,0.0]	[0.0,1.0]	1.7482±0.077542	2.1223±0.085295	2.3465±0.089948	2.4211±0.091499	2.4582±0.092269	2.4802±0.092728	2.4850±0.092828	2.4930±0.092994
[0.7,0.3]	[0.3,0.7]	2.9339±0.104570	3.2269±0.109160	3.4463±0.114220	3.5402±0.116040	3.5985±0.116030	3.6441±0.115650	3.6571±0.115630	3.6910±0.115630
[0.5,0.5]	[0.5,0.5]	2.9505±0.102640	3.2527±0.105370	3.4765±0.107290	3.5730±0.108480	3.6334±0.110050	3.6812±0.112210	3.6951±0.112820	3.7307±0.112070
[0.8,0.3]	[0.3,0.8]	2.9373±0.110370	3.2260±0.112690	3.4378±0.115360	3.5252±0.116560	3.5762±0.117450	3.6101±0.116910	3.6173±0.116740	3.6246±0.118250

Table 5.3: Player 2's average payoffs for stationary Strategies, Example BS1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[0.0,0.0]	[0.0,0.0]	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356	5.2486±0.082356
[1.0,1.0]	[0.0,0.0]	2.2508±0.047614	2.6252±0.055543	2.8496±0.060296	2.9242±0.061883	2.9614±0.062675	2.9834±0.063143	2.9882±0.063245	2.9962±0.063415
[1.0,1.0]	[1.0,1.0]	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214	3.5011±0.055214
[0.0,0.0]	[1.0,1.0]	0.7511±0.015564	1.1257±0.023342	1.3502±0.028008	1.4248±0.029560	1.4620±0.030335	1.4840±0.030796	1.4888±0.030895	1.4968±0.031066
[0.9,0.9]	[0.1,0.1]	2.7047±0.078190	3.0259±0.080685	3.2386±0.083487	3.3237±0.085615	3.3748±0.086875	3.4131±0.087846	3.4234±0.087990	3.4481±0.090151
[1.0,0.0]	[0.0,1.0]	1.7482±0.077542	2.1223±0.085295	2.3465±0.089948	2.4211±0.091499	2.4582±0.092269	2.4802±0.092728	2.4850±0.092828	2.4930±0.092994
[0.7,0.3]	[0.3,0.7]	2.9339±0.104570	3.2269±0.109160	3.4463±0.114220	3.5402±0.116040	3.5985±0.116030	3.6441±0.115650	3.6571±0.115630	3.6910±0.115630
[0.5,0.5]	[0.5,0.5]	2.9505±0.102640	3.2527±0.105370	3.4765±0.107290	3.5730±0.108480	3.6334±0.110050	3.6812±0.112210	3.6951±0.112820	3.7307±0.112070
[0.8,0.3]	[0.3,0.8]	2.7769±0.101740	3.0655±0.105380	3.2773±0.109830	3.3647±0.112130	3.4158±0.114510	3.4496±0.116050	3.4568±0.116720	3.4641±0.119720

Table 5.4: Player 1's average payoffs for cyclic strategies of length 4, Example BS1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[1212;1212]	[1212;1212]	4.3757±0.067399	5.2494±0.080510	5.0748±0.077872	5.2490±0.080507	5.2488±0.080500	5.2485±0.080496	5.2373±0.080326	5.2475±0.080479
[1111;1111]	[2222;2222]	2.2518±0.045624	2.6264±0.053216	2.8509±0.057759	2.9256±0.059275	2.9627±0.060030	2.9848±0.060475	2.9896±0.060572	2.9976±0.060732
[2222;2222]	[1111;1111]	1.7526±0.074562	2.1272±0.082018	2.3517±0.086486	2.4263±0.087977	2.4635±0.088721	2.4855±0.089164	2.4903±0.089261	2.4983±0.089424
[1122;2221]	[1121;2222]	4.0461±0.053049	4.4989±0.064838	4.6039±0.059295	4.6859±0.056036	4.7331±0.052069	4.7605±0.047777	4.7658±0.046234	4.7695±0.054398
[1111;2122]	[1111;2222]	3.6739±0.035141	3.8946±0.033086	4.0056±0.024284	4.0517±0.019639	4.0684±0.017461	4.0732±0.017431	4.0737±0.017992	4.0741±0.043469
[1222;1121]	[1222;1121]	4.6921±0.086842	5.1294±0.093260	5.2542±0.086593	5.3188±0.081088	5.3410±0.075484	5.3467±0.073044	5.3473±0.072718	5.3481±0.075254
[1221;2122]	[2112;1212]	1.8266±0.038766	2.1379±0.037555	2.3348±0.040617	2.3974±0.041615	2.4288±0.042254	2.4489±0.042602	2.4526±0.042716	2.4604±0.043028
[2122;1121]	[1111;2211]	1.5957±0.034849	1.8870±0.031045	2.0633±0.031680	2.1240±0.031429	2.1539±0.031629	2.1716±0.031533	2.1753±0.031596	2.1822±0.032215
[1222;2212]	[1111;2121]	1.5850±0.018312	1.9027±0.023615	2.1038±0.026060	2.2249±0.028872	2.2762±0.029814	2.3224±0.031110	2.3332±0.031191	2.3604±0.031976

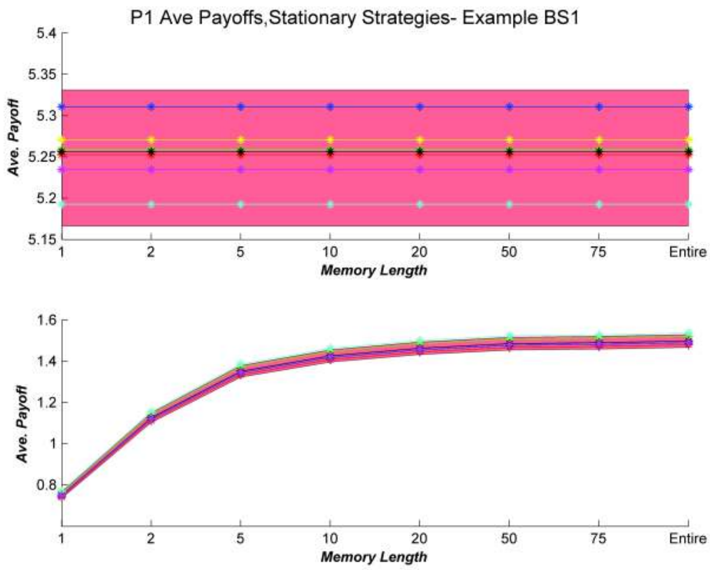


Figure 5.4: Subfigure 1 the players are using strategy pair $\pi = [1.0, 1.0], \sigma = [1.0, 1.0]$. For subfigure 2 the players are using $\pi = [0.0, 0.0], \sigma = [1.0, 1.0]$. Each line represents average payoffs obtained in a single simulation of 1500 stages. The coloured band is the 95% confidence interval of the average payoffs associated with the strategy pair.

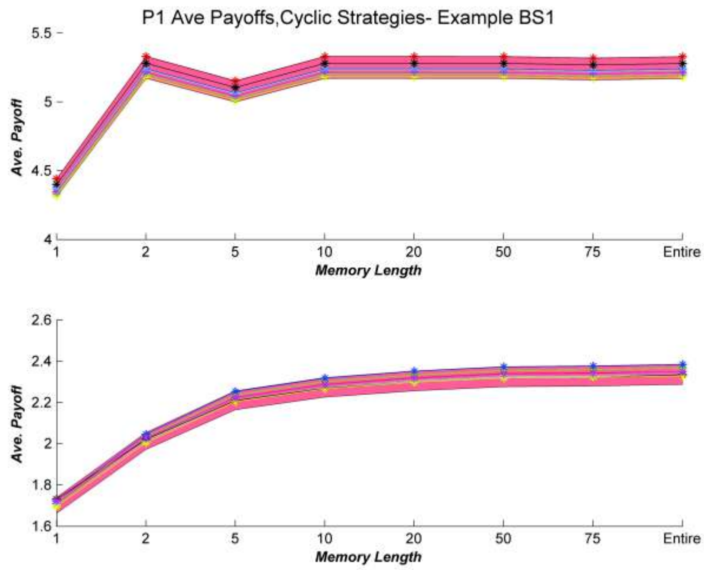


Figure 5.5: Subfigure 1 the players are using strategy pair $\pi = [1212; 1212], \sigma = [1212; 1212]$. In the second subfigure the players are using $\pi = [1221; 2122], \sigma = [2112; 1212]$.Each line represents average payoffs obtained in a single simulation of 1500 stages. The coloured band is the 95% confidence interval of the average payoffs associated with the strategy pair.

5.4.2 Prisoners' Dilemma

We examined example PD1 from section 4.4 to establish the effect of the memory length on average payoffs for a Prisoners' Dilemma game.

In each simulation of 1500 stages, the payoffs obtained by each player for each memory length was recorded. An example of stage payoffs obtained by each player at each stage is illustrated for strategy pair $\pi = [0.8, 0.1], \sigma = [1.0, 1.0]$ in figure 5.6. The figure zooms out a small portion of the stages and shows that the payoffs changes with changing memory length. For memory length 1, player one receives a reward of 3 in several stages which rarely happen for memory length of 2 and does not happen for the other memory lengths. The patterns are also clearly different despite the fact that the payoff sequences are based on the same states and action sequences.

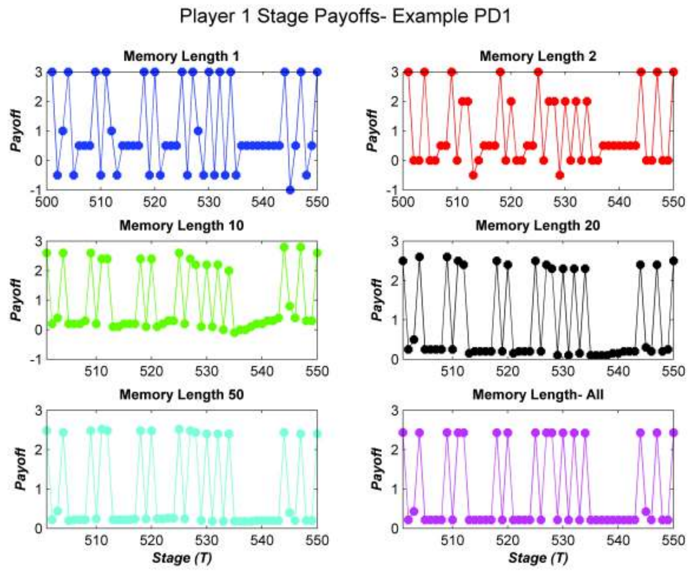


Figure 5.6: Player 1's stage payoffs for $\pi = [0.8, 0.1], \sigma = [1.0, 1.0]$

A summary of the average stage payoffs for a selected nine stationary strategy pairs is shown in table 5.5 for player 1 and table for 5.6 for player 2. For pure stationary strategies, when used by both players in both states, the average payoffs obtained by both players remain the same with increasing memory length. If purely cooperating the players are assured an average payoff which is in the range 5.2461±0.083938 each regardless of the memory length. The average payoffs for the strategy pair $\pi = [0.0, 0.0], \sigma = [0.0, 0.0]$ is plotted in subfigure 1 of figure 5.7. The same pattern is observed when both players are purely defecting and this yields an average payoff within a 95% confidence interval of 1.2241±0.022021. This is explained by noting that for pure stationary strategy pairs, the relative frequencies for action one which is used to reduce the payoffs will be the same for all memory lengths, zero for $\pi = [0.0, 0.0], \sigma = [0.0, 0.0]$, one for $\pi = [1.0, 1.0], \sigma = [1.0, 1.0]$ and 0.5 for $\pi = [0.0, 0.0], \sigma = [1.0, 1.0]$. Thus the same quantity is deducted for the payoffs at each stage for all memory lengths leading to equal average payoffs.

Table 5.5: Player 1's average payoffs for stationary Strategies, Example PD1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[0.0,0.0]	[0.0,0.0]	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938
[1.0,1.0]	[1.0,1.0]	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021
[0.0,0.0]	[1.0,1.0]	0.7508±0.014256	0.7508±0.014256	0.7508±0.014256	0.7508±0.014256	0.7508±0.014256	0.7508±0.014256	0.7508±0.014256	0.7508±0.014256
[0.0,0.5]	[1.0,1.0]	0.8292±0.010081	0.8178±0.009610	0.8063±0.009007	0.8019±0.008500	0.7996±0.008131	0.7984±0.008437	0.7981±0.008758	0.7982±0.019644
[0.8,0.2]	[0.8,0.2]	3.0179±0.099024	2.9881±0.099427	2.9737±0.099132	2.9685±0.099101	2.9659±0.099180	2.9643±0.099314	2.9639±0.099310	2.9614±0.101510
[0.8,0.0]	[1.0,1.0]	1.1121±0.034863	1.1054±0.034272	1.1012±0.033878	1.0998±0.033640	1.0990±0.033619	1.0985±0.033649	1.0983±0.033664	1.0974±0.035685
[0.5,0.5]	[0.5,0.5]	3.2077±0.138330	3.1848±0.138130	3.1656±0.138020	3.1587±0.137990	3.1553±0.137910	3.1535±0.137960	3.1532±0.137890	3.1532±0.141490
[0.8,0.1]	[0.8,0.1]	3.1328±0.094043	3.1006±0.094032	3.0861±0.094237	3.0810±0.094170	3.0785±0.093862	3.0767±0.093682	3.0762±0.093695	3.0739±0.096428
[0.6,0.1]	[0.6,0.1]	3.5555±0.114090	3.5315±0.114120	3.5179±0.114010	3.51330±0.11381	3.5109±0.113480	3.5094±0.113460	3.5089±0.113440	3.5075±0.117710

Table 5.6: Player 2's average payoffs for stationary Strategies, Example PD1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[0.0,0.0]	[0.0,0.0]	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938	5.2461±0.083938
[1.0,1.0]	[1.0,1.0]	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021	1.2241±0.022021
[0.0,0.0]	[1.0,1.0]	6.0015±0.114050	6.0015±0.114050	6.0015±0.114050	6.0015±0.114050	6.0015±0.114050	6.0015±0.114050	6.0015±0.114050	6.0015±0.114050
[0.0,0.5]	[1.0,1.0]	4.8316±0.193410	4.8204±0.193430	4.8087±0.193700	4.8043±0.193540	4.8020±0.193540	4.8008±0.193590	4.8005±0.193580	4.8006±0.193700
[0.8,0.2]	[0.8,0.2]	3.0134±0.097107	2.9837±0.096849	2.9692±0.096618	2.9641±0.096749	2.9614±0.096603	2.9598±0.096612	2.9594±0.096752	2.9569±0.099860
[0.8,0.0]	[1.0,1.0]	3.7953±0.069655	3.7886±0.069878	3.7844±0.070019	3.7830±0.069965	3.7822±0.070022	3.7817±0.069985	3.7815±0.069946	3.7806±0.071950
[0.5,0.5]	[0.5,0.5]	3.2140±0.133090	3.1911±0.133330	3.1719±0.133280	3.1650±0.133190	3.1616±0.133380	3.1598±0.133590	3.1595±0.133710	3.1595±0.136190
[0.8,0.1]	[0.8,0.1]	3.1329±0.085667	3.1007±0.085324	3.0863±0.085073	3.0811±0.085031	3.0786±0.085093	3.0768±0.085099	3.0763±0.085107	3.0740±0.089688
[0.6,0.1]	[0.6,0.1]	3.5566±0.120240	3.5326±0.119940	3.5190±0.120010	3.5144±0.120270	3.5120±0.120560	3.5105±0.120720	3.5101±0.120870	3.5086±0.123450

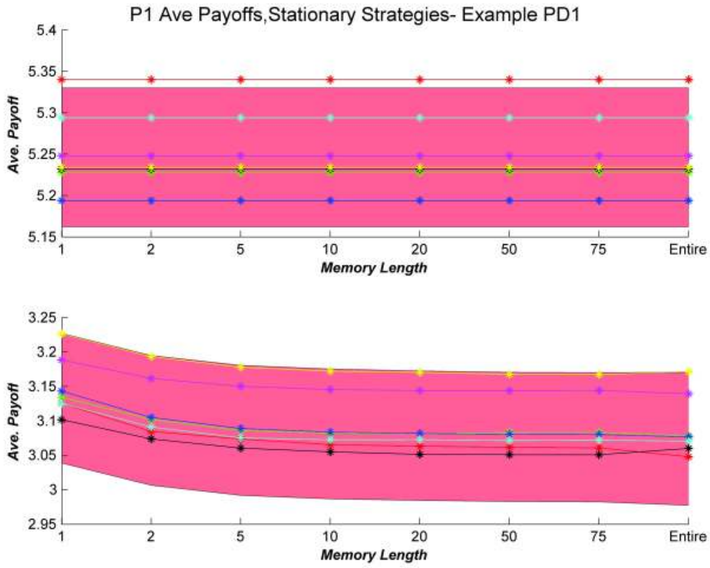


Figure 5.7: Top: $\pi = [0, 0], \sigma = [0, 0]$. Bottom: $\pi = [\frac{8}{10}, \frac{1}{10}], \sigma = [\frac{8}{10}, \frac{1}{10}]$. Each line represents average payoffs obtained in a single simulation of 1500 stages. The coloured band is the 95% confidence interval of the average payoffs associated with the strategy pair.

If one of the players employs a mixed stationary strategy then the average payoffs for each player will slightly decrease with increasing memory length. Subfigure 2 in figure 5.7 shows the average payoffs for strategy pair $\pi = [\frac{8}{10}, \frac{1}{10}], \sigma = [\frac{8}{10}, \frac{1}{10}]$ which has an interval of 3.1328 ± 0.094043 for memory length 1. For the same strategy pair both players will get average payoffs in the interval 3.0740 ± 0.089688 if the payoffs are adjusted with full memory length. This rather unexpected anomaly, though insignificant was further investigated to ascertain how it arises. The investigations focused on the average reductions and it was found that apparently, reductions in some cases may increase with increasing memory lengths whilst in some cases the average reduction may decrease with increasing memory length. Strategy pairs associated with the former coincidentally involve playing action 1 with high probabilities whereas the latter involves playing action 1 with relatively low probabilities. Figure 5.9 subfigures top and centre shows reductions associated with strategy pairs $\pi = [\frac{6}{10}, \frac{5}{10}], \sigma = [\frac{6}{10}, \frac{5}{10}]$ and $\pi = [\frac{8}{10}, \frac{1}{10}], \sigma = [\frac{8}{10}, \frac{1}{10}]$ respectively and these seem to increase with increasing memory length. The bottom subfigure in figure 5.9 shows average reductions associated with the strategy pair $\pi = [0, \frac{5}{10}], \sigma = [1, 1]$ which decrease with increasing memory length. No plausible explanations could be given for this observation.

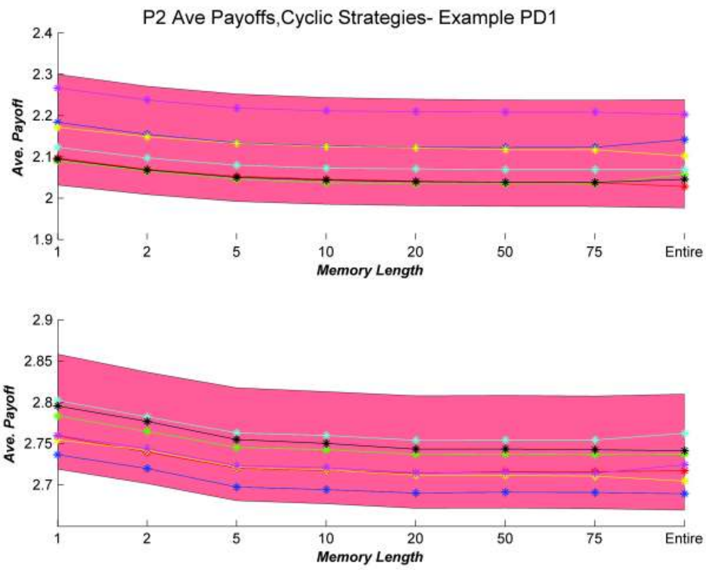


Figure 5.8: Top: $\pi = [1221; 1111], \sigma = [1121; 2111]$. Bottom: $\pi = [1112; 1122], \sigma = [2111; 1112]$.Each line represents average payoffs obtained in a single simulation of 1500 stages. The coloured band is the 95% confidence interval of the average payoffs associated with the strategy pair.

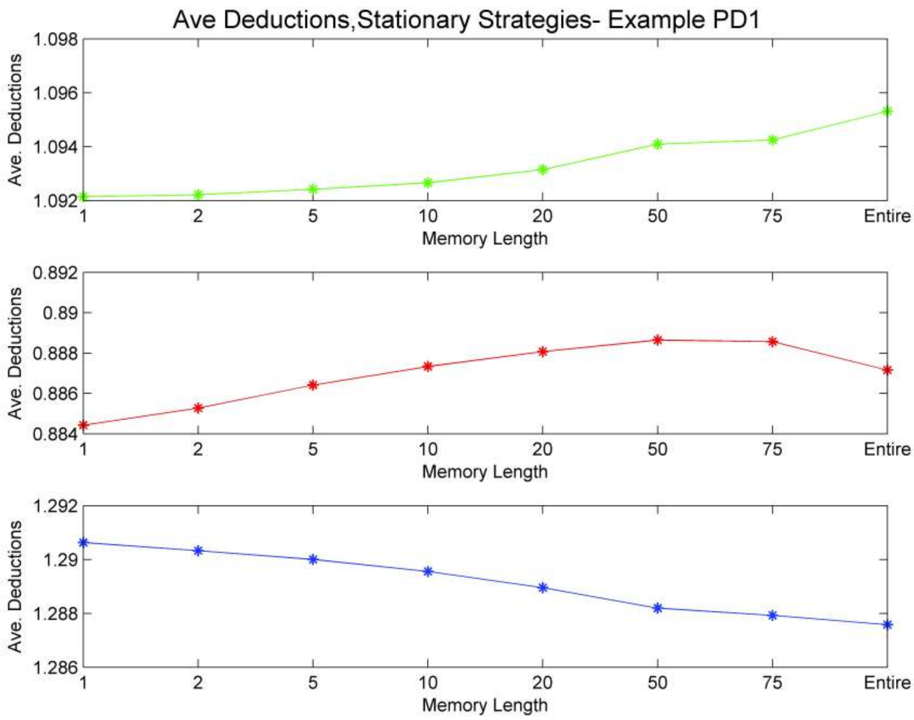


Figure 5.9: Average reductions for stationary strategy pairs. Top: $\pi = [\frac{6}{10}, \frac{5}{10}], \sigma = [\frac{6}{10}, \frac{5}{10}]$. Centre: $\pi = [\frac{8}{10}, \frac{1}{10}], \sigma = [\frac{8}{10}, \frac{1}{10}]$. Bottom: $\pi = [0, \frac{5}{10}], \sigma = [1, 1]$.

For the Prisoners' Dilemma example cyclic strategy pairs have similar patterns observed for stationary strategy pairs. If players use cyclic strategies with the same action for each state, this is similar to pure stationary strategies, then all the players will receive the same average rewards for all memory lengths. Any other cyclic strategy pair will produce similar results as those of mixed stationary strategies however with tighter intervals since only the states sequences will contribute towards the differences. Figure 5.8 shows average payoffs for cyclic strategies of length four. The same reasoning used to explain the trend stationary strategies, pure and mixed, also applies for cyclic strategies.

5.4.3 Small Fish War

The example presented in section 4.2 is investigated for to establish the relationship between memory length and average payoffs for a Small Fish War games. Using preliminary results from trial simulations, a few strategy pairs were selected and simulations for of 1500 stages each were done for each strategy pair. Figure 5.6 displays a portion of the stage rewards obtained by player 1 for the stationary strategy pair $\pi = [0, 0.0], \sigma = [1.0, 0.1]$.

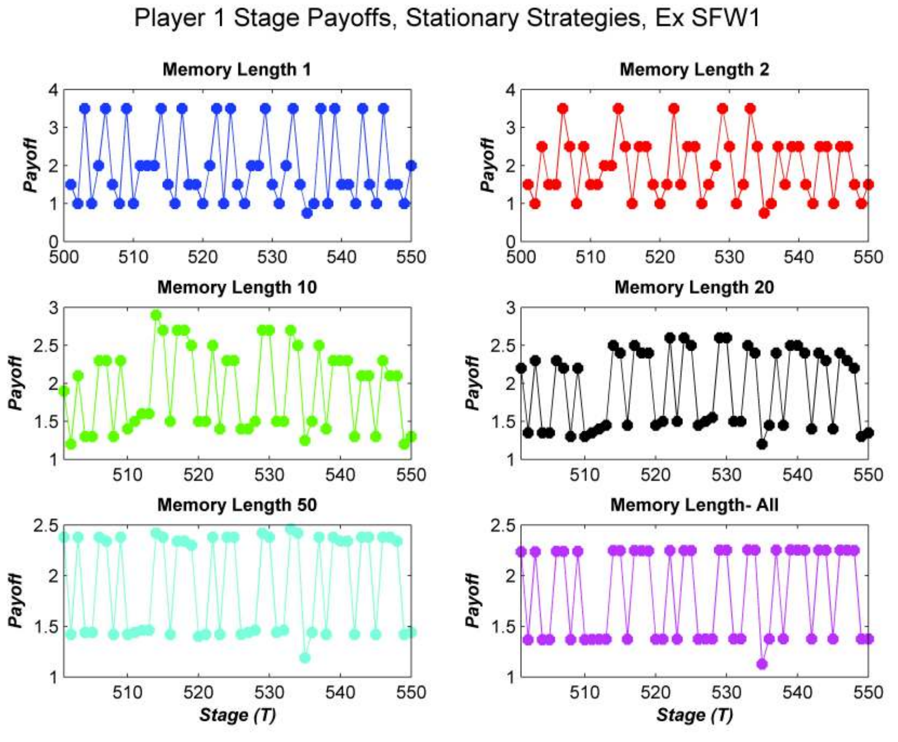


Figure 5.10: Player 1's stage payoffs for $\pi = [0.0, 0.0], \sigma = [1.0, 0.1]$

As expected and similar to the other two games, mixed stationary strategies lead to different stage payoffs for different memory lengths. A close inspection of figure 5.10 shows that the patterns of the stage rewards are different from each other. A summary of player 1's average stage payoffs for selected nine stationary strategy pairs is shown in table 5.7 and table 5.8 for player 2. The average payoffs are given as a 95% confidence interval as in the other two games to take care of changing actions and states sequence in each simulation.

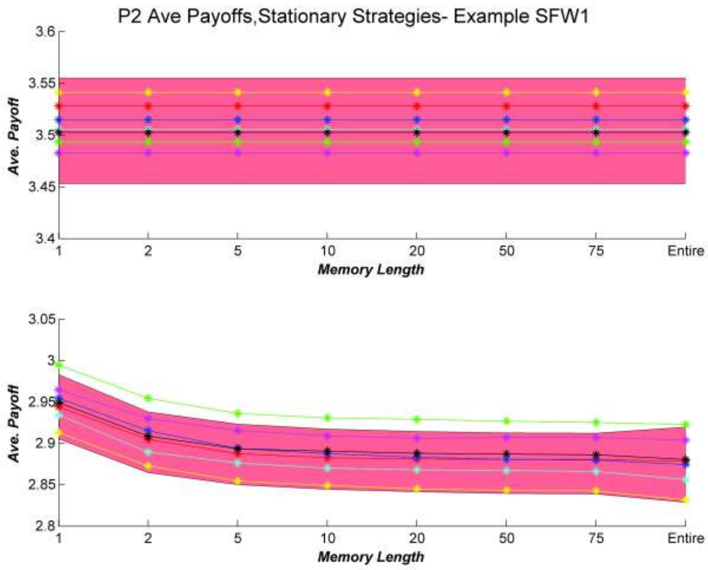


Figure 5.11: Subfigure 1: $\pi = [0,0], \sigma = [0,0]$. Subfigure 2: $\pi = [1, \frac{1}{10}], \sigma = [1, \frac{1}{10}]$. Each line represents average payoffs obtained in a single simulation of 1500 stages. The coloured band is the 95% confidence interval of the average payoffs associated with the strategy pair.

The Small Fish War updating mechanism is similar to the one used for the Prisoners' Dilemma and thus the trends of the average payoffs are identical for the two games. Average payoffs for pure stationary strategy pairs are the same for all memory lengths' for each strategy pair simulation. For instance if the players use the strategy pair $\pi = [0,0], \sigma = [0,0]$, subfigure 1 in figure 5.11 shows that player 2 will receive an average payoff of 3.5041±0.051059 for all memory lengths. Player 1 will also receive the same amounts as shown in table 5.8 row one. If both players choose to fish without restraint, $\pi = [1,1], \sigma = [1,1]$, then both players will receive an average payoff of 2.1413±0.039476 apiece. Just as in the case involving the Prisoners' Dilemma, the pattern is explained by noting that the relative frequency for the first action will always be the same for all memory lengths at each adjusting moment. If the strategy pair involves some mixed strategy by any of the players then the expected average payoffs may decrease or increase, though insignificantly, with increasing memory length. Figure 5.11 subfigure 2 shows instances of payoffs obtained when the players are employing strategy pairs $\pi = [1, \frac{1}{10}], \sigma = [1, \frac{1}{10}]$. As stated earlier when explaining the trends for the Prisoners' Dilemma no plausible explanation can be given for this trend and further work needs to be done to ascertain causes of this trend.

Similar patterns for average payoffs to those based on stationary strategy pairs are observed for the Small Fish War's game if players use cyclic strategy pairs.

Table 5.7: Player 1's average payoffs for stationary Strategies, Example SFW1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[0.0,0.0]	[0.0,0.0]	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059
[1.0,1.0]	[1.0,1.0]	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476
[0.0,0.0]	[1.0,0.1]	2.3756±0.017150	2.3746±0.017144	2.3741±0.016916	2.3738±0.016947	2.3736±0.017066	2.3735±0.017207	2.3734±0.017390	2.3731±0.024980
[0.9,0.2]	[0.0,0.0]	3.8638±0.072032	3.8607±0.072247	3.8585±0.072192	3.8576±0.072096	3.8572±0.072117	3.8568±0.072240	3.8566±0.072291	3.8563±0.073695
[0.6,0.0]	[0.6,0.0]	3.1415±0.048790	3.1164±0.048267	3.1035±0.047955	3.0992±0.047727	3.0968±0.047699	3.0954±0.048048	3.0951±0.048304	3.0928±0.055729
[0.5,0.5]	[0.5,0.5]	2.8421±0.073161	2.8192±0.072222	2.8000±0.072168	2.7933±0.072102	2.7899±0.072102	2.7879±0.071906	2.7874±0.071912	2.7856±0.074445
[1.0,0.1]	[1.0,0.1]	2.9452±0.036716	2.9023±0.033990	2.8876±0.034158	2.8819±0.033777	2.8790±0.033807	2.8771±0.033845	2.8766±0.033988	2.8752±0.042291
[0.9,0.2]	[0.9,0.2]	2.8980±0.047268	2.8647±0.044820	2.8506±0.044074	2.8454±0.044141	2.8429±0.044190	2.8412±0.044203	2.8407±0.044401	2.8390±0.048489
[0.5,0.1]	[0.5,0.1]	3.1242±0.055737	3.1035±0.055274	3.0904±0.055344	3.0860±0.055185	3.0837±0.054989	3.0822±0.054880	3.0819±0.054939	3.0807±0.060440

Table 5.8: Player 2's average payoffs for stationary Strategies, Example SFW1

π	σ	1	**2**	**5**	**10**	**20**	**50**	**75**	∞
[0.0,0.0]	[0.0,0.0]	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059	3.5041±0.051059
[1.0,1.0]	[1.0,1.0]	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476	2.1413±0.039476
[0.0,0.0]	[1.0,0.1]	3.9069±0.073431	3.9060±0.073612	3.9054±0.073622	3.9052±0.073615	3.9050±0.073785	3.9049±0.073873	3.9048±0.073937	3.9044±0.076134
[0.9,0.2]	[0.0,0.0]	2.4193±0.030147	2.4162±0.029429	2.4139±0.029333	2.4131±0.029189	2.4126±0.029187	2.4122±0.029485	2.4121±0.029645	2.4118±0.034926
[0.6,0.0]	[0.6,0.0]	3.1433±0.048416	3.1182±0.047218	3.1053±0.046519	3.1010±0.046598	3.0986±0.046744	3.0972±0.047111	3.0969±0.047415	3.0946±0.056314
[0.5,0.5]	[0.5,0.5]	2.8403±0.070323	2.8174±0.069871	2.7983±0.070133	2.7916±0.070073	2.7881±0.070191	2.7861±0.070267	2.7856±0.070467	2.7839±0.074668
[1.0,0.1]	[1.0,0.1]	2.9442±0.038822	2.9013±0.036669	2.8867±0.036575	2.8809±0.036360	2.8780±0.036420	2.8761±0.036564	2.8756±0.036695	2.8742±0.045403
[0.9,0.2]	[0.9,0.2]	2.8972±0.049306	2.8639±0.047025	2.8497±0.047039	2.8445±0.046833	2.8421±0.046858	2.8404±0.046854	2.8399±0.046842	2.8381±0.050896
[0.5,0.1]	[0.5,0.1]	3.1236±0.054144	3.1030±0.053453	3.0899±0.053108	3.0855±0.053060	3.0832±0.052931	3.0817±0.052746	3.0814±0.052751	3.0801±0.058195

If both the strategy sets for the two players have the same action for each state then all the players will receive constant average rewards for all memory lengths. Subfigure 2 in figure 5.12 shows that for the strategy pair $\pi = [2222; 2222], \sigma = [2222; 2222]$ the players receive an average payoff of 3.5173. Any other cyclic strategy pair will produce similar results as those of mixed stationary strategies. For the strategy pair $\pi = [1221; 1111], \sigma = [1121; 2111]$, the players receive payoffs similar to those in subfigure 1 of 5.12. The same reasoning used to explain the trend stationary strategies, pure and mixed, for the same game also apply for cyclic strategies.

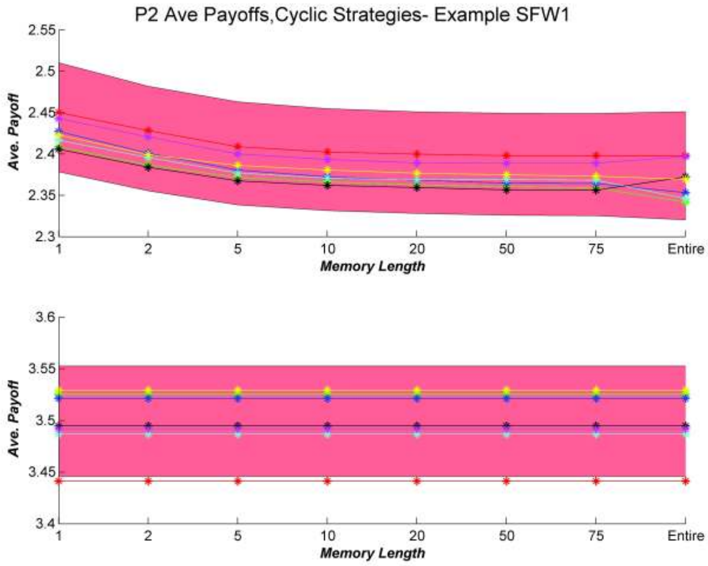


Figure 5.12: Subfigure 1: $\pi = [1221; 1111], \sigma = [1121; 2111]$. Subfigure 2: $\pi = [2222; 2222], \sigma = [2222; 2222]$. Each line represents average payoffs obtained in a single simulation of 1500 stages. The coloured band is the 95% confidence interval of the average payoffs associated with the strategy pair.

5.5 Results: Threat Points

Effect of memory length on threat point values was done using stationary strategy pairs and setting the range of each of the players strategy to [0 1] in both the states. Using the *linspace* function, n points were generated from each range thus creating n^4 possible strategy pairs. Average payoffs were obtained for each strategy pair. To obtain the threat points of player 1(2) the n^2 payoffs generated from each combination of player 2(1)'s strategy pair were grouped together and the maximum was stored. This was done for all player 2(1)'s strategy pairs and for each memory length. The minimum maximum was set as the threat point and the accompanying strategy from the minimising player was stored as the threat strategy. Given that the running time of the routine is of order $\mathcal{O}(n^4)$, a balance had to be made between the quality of results and time needed to obtain the results. Consequently n was set at 11 which implied a step size of 0.1 for the strategy pairs. The threat points obtained in each simulation for all memory lengths was always changing in magnitude due to different actions and states sequences. As such the threat point of the game is given as a 95% approximate calculated from a sample size of 200. For each strategy pair, 1000 stages were used to obtain the average payoffs after which the maximum as for each memory length was taken as the threat point approximate of that simulation . Below are the results for several games. Given that the games considered in this project are symmetrical the threat points should be equal and the threat strategies should be similar.

5.5.1 Battle of the Sexes

The Battle of the Sexes game investigated under this section is **Example BS1**.Table 5.9 shows the threat points $\overline{V}$ (given as a 95% confidence interval estimate) and the threat strategies, $\pi^\star$ and $\sigma^\star$, for each memory length.

Table 5.9: Threat point 95% confidence interval for the game Example BS1

Memory Length	$\overline{V}$	$\pi^\star$	$\sigma^\star$
1	3.1573±0.0499	[0.8,0.7]	[0.2,0.3]
2	3.4048±0.0491	[0.9,0.8]	[0.1,0.2]
5	3.5299±0.0451	[0.9,0.9]	[0.1,0.1]
10	3.5836±0.0468	[0.9,0.9]	[0.1,0.1]
20	3.6113±0.0483	[0.9,0.9]	[0.1,0.1]
50	3.6276±0.0473	[0.9,0.9]	[0.1,0.1]
75	3.6311±0.0477	[0.9,0.9]	[0.1,0.1]
∞	3.6388±0.0481	[0.9,0.9]	[0.1,0.1]

For the Battle of the Sexes game, the threat point increases with increasing memory length. This trend follows the observed trend for average payoffs and thus is explained with the same line of reasoning: for similar mixed stationary strategy pairs the reductions are larger for smaller memory lengths and as such the associated payoffs increase with increasing memory length. Table 5.9 shows that when minimising the other players payoffs, the player concerned should play their preferred action with a very high probability. For memory length 1, both players play their preferred action with probability 0.8 in state 1 and use probabilities 0.7 in state 2. Memory length of 2 requires the players to play their preferred action with probability 0.9 in state 1 and 0.8 in state 2 as retaliation strategies. For the other memory lengths used in this project, both players play their preferred actions with a probability of 0.9 in both states. Since the threat point obtained in this project is an approximate due to the fact that we took steps of 0.1 when generating the strategy sets, the best response to the threat strategy will always change due to varying actions and states sequences. A more precise threat point with the associated threat strategies can be obtained if the step taken when dividing the strategy range in each state is reduced, however this come with computational challenges. Since both players' threat strategies involve stationary strategies where each player play their preferred action with high probability, the threat points are sensitive to the choice of the reduction coefficients c.

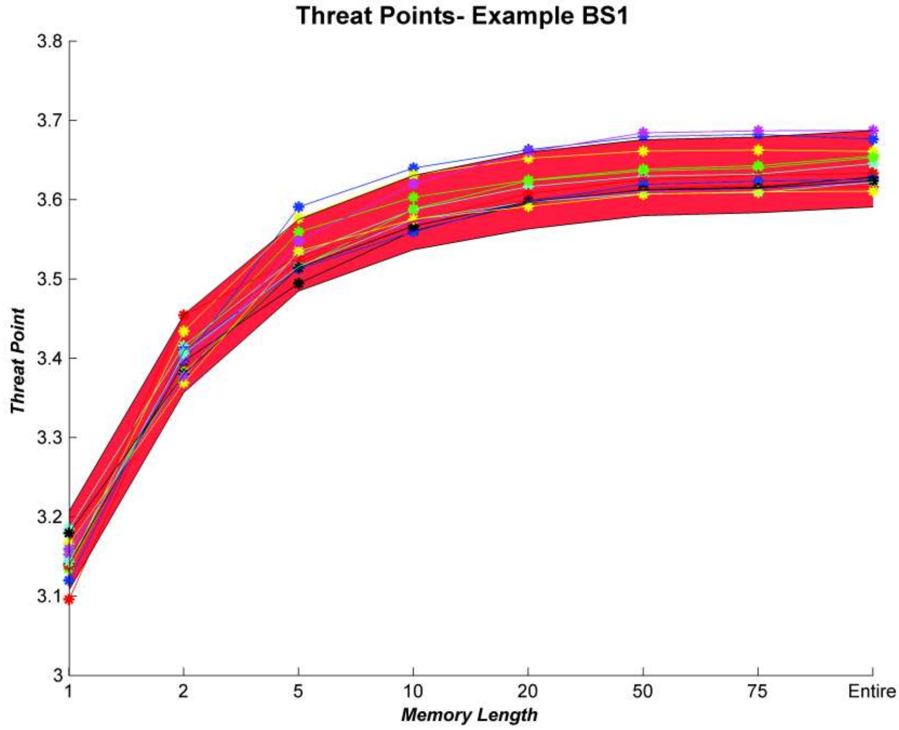


Figure 5.13: Threat point 95% confidence (filled region)interval for Example BS1. The lines are threat points obtained in a single simulation.

5.5.2 Prisoners' Dilemma

We consider the game described under Example PD1 and use the same procedure as described in the beginning of this section to determine threat points. Since the game is symmetric the threat point values are equal. As stated earlier the average payoffs for a strategy pair will be marginally different for each simulation thus several simulations are done for the threat strategy and the entire set of strategies for the maximising player. The threat point of the game is then calculated and given as a 95% confidence interval based on 200 simulations. The results for example PD1 are displayed in table 5.10.

Table 5.10: Threat point values Example PD1

Memory Length	$\overline{V}$	$\pi^\star$=$\sigma^\star$
1	1.2414±0.0211	[1.0,1.0]
2	1.2376±0.0217	[1.0,1.0]
5	1.2356±0.0219	[1.0,1.0]
10	1.2350±0.0222	[1.0,1.0]
20	1.2346±0.0223	[1.0,1.0]
50	1.2344±0.0223	[1.0,1.0]
75	1.2344±0.0223	[1.0,1.0]
∞	1.2353±0.0223	[1.0,1.0]

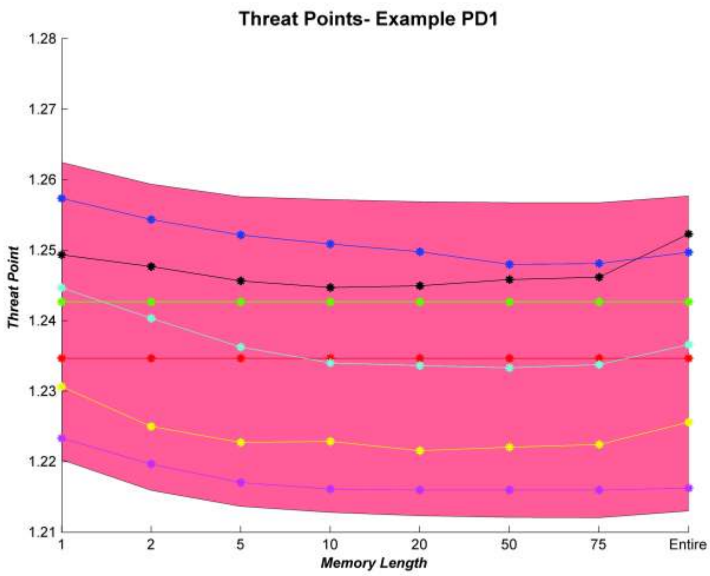


Figure 5.14: Threat point 95% confidence (filled region)interval for Example PD1. The lines are typical threat points obtained in a single simulation.

The threat point magnitude follows the trend observed for stage average payoffs. The payoffs are insignificantly different as they lie within the same 95% confidence interval. The differences in some cases (some lines from figure 5.14 are not horizontal) though insignificant, may be explained by the fact that when determining the threat points, we did not use a fixed best reply to the threat strategy, but rather we used the entire strategy range and took the maximum in each simulation. Thus the best replies differed in each simulation hence the small differences.

5.5.3 Small Fish War

Reconsider the Small Fish War game described under **Example SFW1**. Table 5.11 shows the threats point value, V expressed as a 95% confidence interval of 200 samples and associated threat strategies,$\pi^\star$ and $\sigma^\star$, employed by the two players when retaliating.

Table 5.11: Threat point values for Example SFW1

Hist Length	$\overline{V}$	$\pi^\star = \sigma^\star$
1	2.2457±0.0419	[1.0,1.0]
2	2.2367±0.0415	[1.0,1.0]
5	2.2323±0.0412	[1.0,1.0]
10	2.2307±0.0413	[1.0,1.0]
20	2.2298±0.0417	[1.0,1.0]
50	2.2291±0.0418	[1.0,1.0]
75	2.2290±0.0417	[1.0,1.0]
∞	2.2299±0.0422	[1.0,1.0]

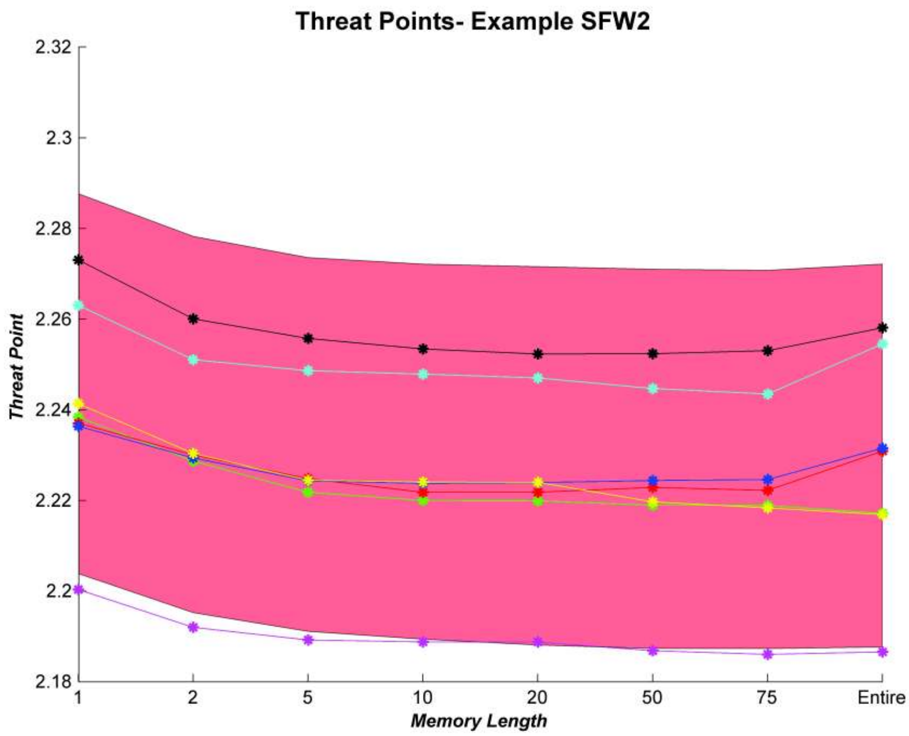


Figure 5.15: Threat point 95% confidence (filled region)interval for Example SFW1. The lines are typical threat points obtained in a single simulation.

Table 5.11 shows that the threat points 95% confidence interval follows a similar trend like those observed for the Prisoners' Dilemma game. The confidence interval overlaps for all memory lengths and the difference are thus insignificant. The reasons why there seem to be differences were suggested in the previous section. For threat strategies both players have to resort to choosing action 1 (fishing without restraint) if the other player deviates from their initial strategies. The threat strategies found in this game are similar to the results obtained by Joosten et.al.[7] thus there is no apparent change in behaviour of the players in their threat strategies.

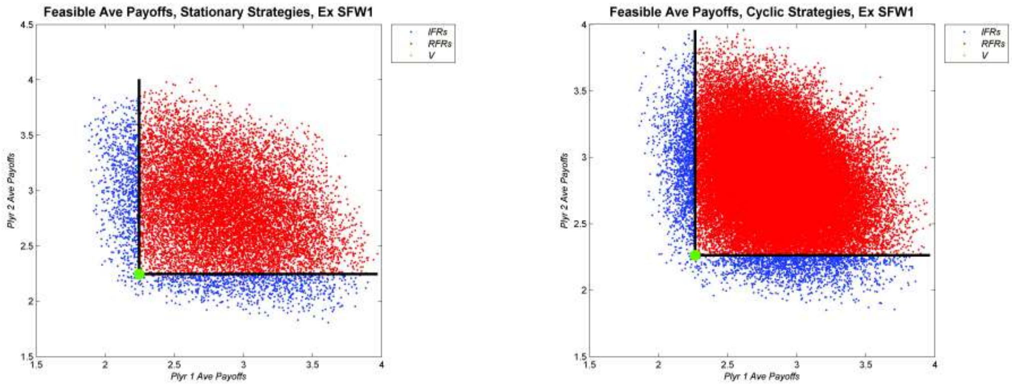
5.6 Feasible Rewards

The third and final objective of the project was determining the set of feasible rewards for the examples investigated in this thesis as well as distinguishing between irrational and rational feasible rewards. Feasible rewards were determined for both cyclic and stationary strategies. To obtain a set of feasible rewards for stationary strategies the strategy range was set to [0 1] for all states and players. To get a good approximation of the feasible payoff region the strategies are generated by creating uniformly spread strategies in each range. A balance also had to be reached between computing time and quality of results thus there is need to limit the number of strategies in each state. After several trials the number of strategies in each range was set at 11 implying a step of 0.1. For cyclic strategies, the entire set of strategy length l_{str} was created using the function *StrategyGenerator* and payoffs were calculated for all possible combinations of strategy pairs. Trimming down the strategy set in an attempt to reduce computational time however with a good representation of the entire strategy set proved to be a tedious task. Consequently only strategy lengths of up to 4 were considered. For strategy length 4 the computing time was 35224 seconds. Distinguishing between rational(RFRs) and irrational(IFRs) average payoffs is done using threat points obtained using the routine described in the section 5.5.

5.6.1 Small Fish War: Example SFW1

For the SFW game, example SFW1, the feasible payoffs region for memory length 1 is shown in figures 5.16a and 5.16b. The shapes of the feasible regions are similar for both classes of strategies and all memory lengths. As can be seen from both the figures, 5.16a and 5.16b, the threat point is low in both cases making most of the feasible average rewards rational feasible rewards. Thus if players incorporate threats into their strategies, playing action 1(fishing "without restraint") always, they can easily get any average pair reward above the game value (2.2457,2.2457)±0.0419 each. The best compromise for both players is to fish "with restraint", both using [0 0], and this yields expected average payoffs within the range 3.5006±0.053500 for each player. This payoff combination is the average of the entries $M_{s,2,2}$ in both states. Any deviation by either one of the players will lead to the other player invoking the threat strategy. If both players using action 1 purely, then each player will get an expected average payoffs within the range 0.9194±0.017594. In addition to the very low average payoffs, this might as well lead to the collapse of the system.

For other memory lengths the average payoffs region stays the same but with slightly lower threat points. The reason for this was discussed before in the previous section dealing with threat points. Investigations made during the simulations showed that changing adjusting coefficients, c, has an effect on the shape of the feasible average rewards. For coefficients of $c = [2\ 1]$ and other variables remaining the same leads to noticeable changes in the average payoffs area shapes. For the strategy pair, $\pi = [0, 0], \sigma = [0, 0]$, the average payoffs remains in the interval 3.5006±0.53500 whilst the lower vertex of the feasible payoff region is stretched towards the origin as the reductions associated with this strategy pair increases from 2 to 4 since $\varrho_t^1 +$



(a) (b)

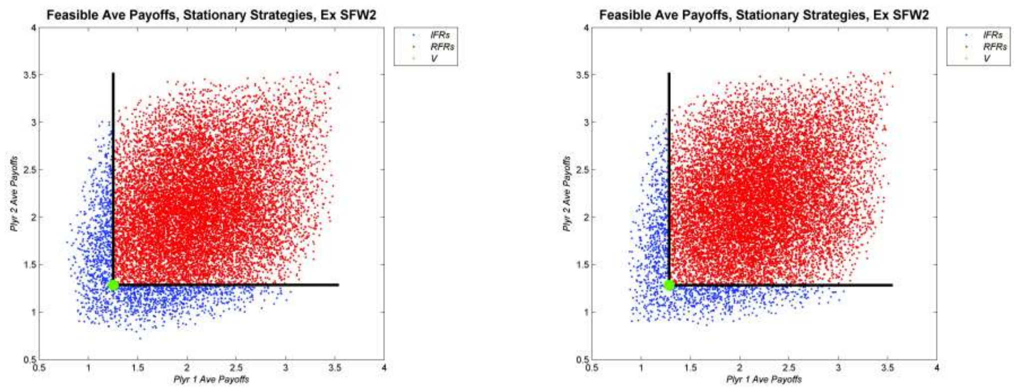
Figure 5.16: Cyclic strategies of length 4, memory length 1. The 95% confidence interval of the threat point is V=2.2457±0.0419

$\varrho_t^2 = 2$ always. The other two vertices associated with the strategy pair where one player purely chooses action 1 whilst the other purely chooses action 2 always almost disappears as players get smaller payoffs such strategy pairs. In this case associated reductions increase to 2 since $\varrho_t^1 + \varrho_t^2 = 1$ always. Consequently, average rewards obtained from the strategy pair $[\pi = [0, 0], \sigma = [0, 0]]$ requires no threat strategies as it yields the maximum possible payoff for each player. Figures 5.17a and 5.17b shows the average payoffs associated with $c = [2\ 1]$ and other variables remaining the same for example SFW1.

The results obtained in this game are similar to the results obtained by Joosten et.al.[7] in terms of best strategies for the players and behavior. Joosten et. al. found that the payoffs associated with using action 1 always yielded rewards which were slightly higher if the players used action 2. Furthermore the threat strategies in Joosten et.al.[7] are the same as those found in this project. Since the threat strategies are the same we may infer that the players strategies may not necessarily change. However it becomes necessary for the players to use the action 2 (fish “with restraint”) if the effects of externalities are very high (large values for c) as the payoffs associated with action 1 are very low for huge values of c where as the payoffs associated with always using action 1 remain the same..

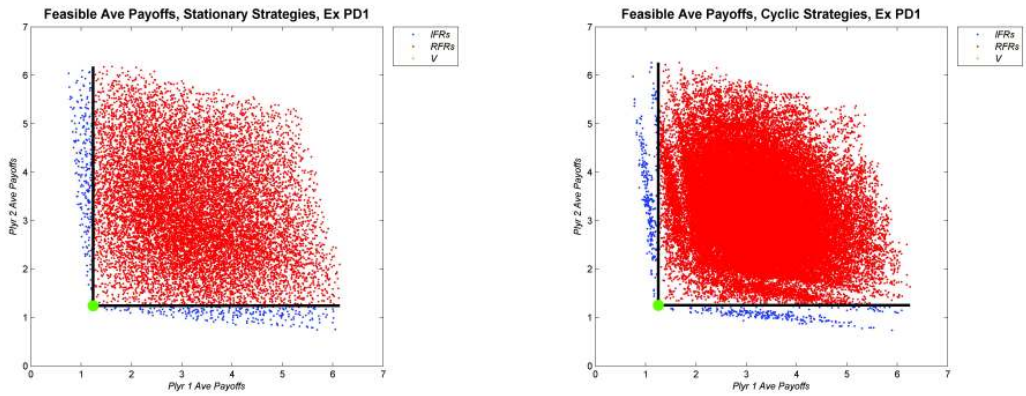
5.6.2 Prisoners’ Dilemma: Example PD1

The feasible average payoffs region for the Prisoners’ Dilemma game is similar to that of the Small Fish war game and is convex in shape. This shape is the same for all memory lengths with the difference being slight movements towards and away from the origin. The threat point for all memory lengths are very low making most of the points rational feasible rewards. The threat point coincides with the average payoffs which the players obtain if they both play the action 1 with probability 1. The top vertex of the region in figure 5.18a is obtained when both payers choose the second action with probability one. This yields an expected average payoff of



(a) (b)

Figure 5.17: (a)full memory length,(b)memory length 5, $c = [2\ 1]$.



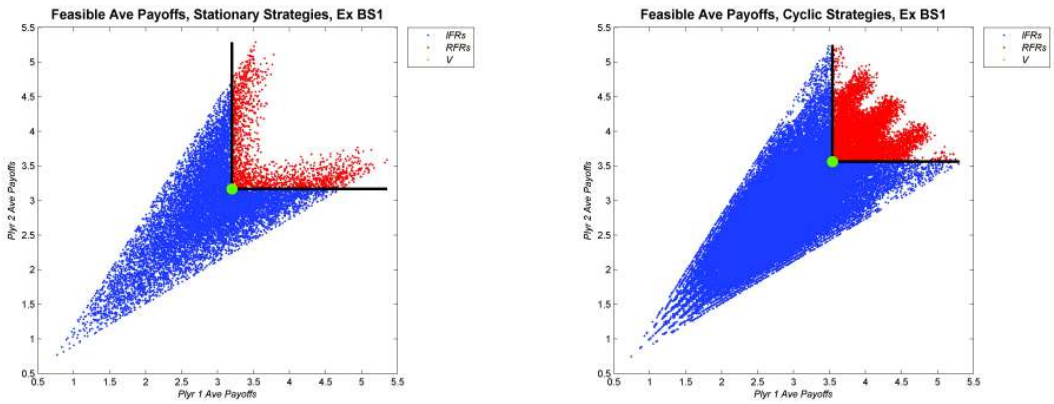
(a) (b)

Figure 5.18: Cyclic strategies of length 4, memory length 1. The 95% confidence interval of the threat point is V=1.2414±0.0211

5.2461 ± 0.083938 for each player, which is the best compromise for both players. If one of the players' purely chooses action 1 whilst the other purely plays action 2, the players are excepted to earn 6.0015 ± 0.114050 and 0.7508 ± 0.014256 respectively. This average payoff pair is outside the rational feasible payoff for both players. The same shapes are obtained for cyclic strategies and figure 5.18b shows the feasible payoffs for memory length 1.

5.6.3 Battle of the Sexes: Example BS1

Unlike the other two games investigated in this thesis the Battles of the Sexes game has different results for the feasible average payoffs. To begin with, the rational feasible average payoff region is smaller as compared to the other two games. Furthermore the feasible average payoff region shapes are different for stationary strategies and cyclic strategy pairs and these also change with changing memory length. Figure 5.19a shows average payoffs for memory length one when the players are using



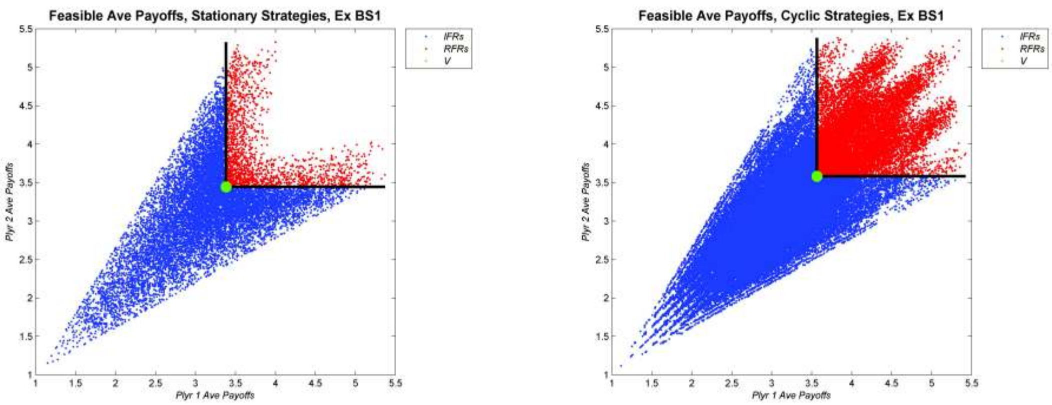
(a) (b)

Figure 5.19: Cyclic strategies of length 4, memory length 1. The 95% confidence interval of the threat point is 3.1573±0.0499

stationary strategies. The two peaks at the top of the cone are obtained when one of the player plays his preferred action with probability 1 whilst the other player plays their preferred action with probability 0. The gap, curving inwards between the two peaks towards the origin, results from players employing mixed stationary strategy pairs result in only entries $\{i, j\} \in [\{1, 1\}, \{2, 2\}]$ being chosen all the time. This will result in the average payoffs falling inwards due to two factors. The first is that the mixed strategy pair will result in the entries $\{i, j\} \in [\{1, 2\}, \{2, 1\}]$ being chosen in some stages which by default have smaller payoffs for each player. Additionally the mixed strategies will also cause reductions from attending different events as well as unbalanced attending of the events. This holds for all other memory lengths as well. This will lead to an average which is less than those at the peaks for each player, hence the curving inwards.

For cyclic strategy pairs, the shape is cone-like but with almost a flat top. Figure 5.19b shows the feasible payoff for cyclic strategy pairs of length 4. This is a consequence of synchronised alternating between the entries $\{i, j\} \in [\{1, 1\}, \{2, 2\}]$ which happens with certainty. Thus the averages defined by the line joining the two peaks can be obtained as long as the cyclic strategies lead to the two entries being chosen. Other cyclic strategy pairs will result in the other entries being chosen and this produces average payoffs which are smaller than any combination of the entries $\{i, j\} \in [\{1, 1\}, \{2, 2\}]$.

Increasing the memory length to 2 leads to similar shapes for stationary strategy pairs however with slight differences at the peak. Figure 5.20a shows the average payoffs for stationary strategy pairs with the payoffs being adjusted using memory length 2. The inner part of the cone head is slightly stretched upwards. For cyclic strategies the change in the shape is more defined as seen in figure 5.20b. The peaks are stretched out with the middle being the biggest, with the peak resembling the average payoffs the players obtain if they use cyclic strategies where the entries $\{i, j\} \in [\{1, 1\}, \{2, 2\}]$ are chosen 50% of the time each. The immediate peak to the right (left) is when players are synchronised but with entry {1,1} ({2,2}) being chosen



(a) (b)

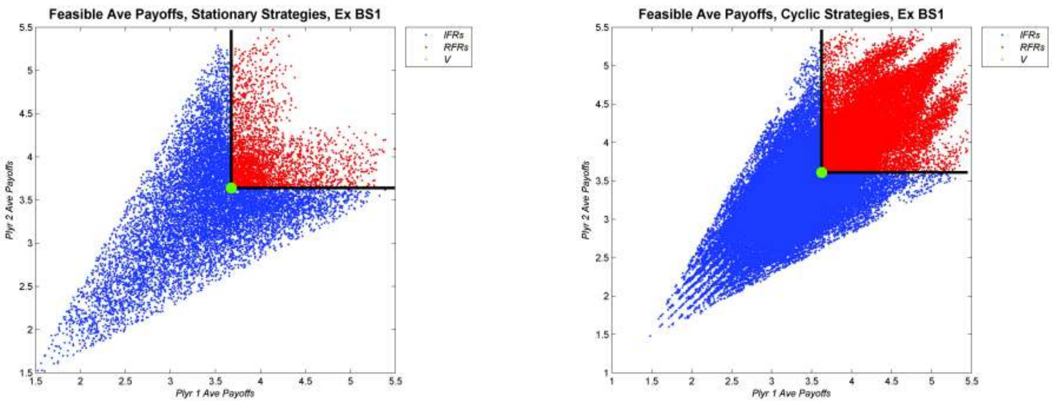
Figure 5.20: Cyclic strategies of length 4, memory length 2. The 95% confidence interval of the threat point is 3.4048±0.0491

$\frac{3}{4}$ of the time and entry 2,2} ({1,1}) being chosen $\frac{1}{4}$ of the time. The outer peaks are produced when one player plays their preferred action with probability 1 and the other chooses the preferred action with probability 0. The observed increase of the average payoffs results from reduced reductions. For coordinated cyclic strategy pairs (these are the strategy pairs which gives the peaks), $e_t = 0$ at all adjusting moments as relative frequencies for the action pairs $\{i, j\} \in [\{1, 1\}, \{2, 2\}]$ are both 0.5. On the contrary, using the same coordinated cyclic strategies for memory length 1 $e_t = 1$ always as only 1 action pair has been played in the previous stage. The other two adjusting components stay the same for both memory lengths but are not relevant for the coordinated strategy pairs.

Subsequent increase of the memory length does lead to small changes in shapes of feasible payoffs regions for both stationary and cyclic strategy pairs. However the top of the cone is stretched upwards giving slightly higher payoffs for longer memory lengths. Figures 5.21a and 5.21b shows feasible average payoffs for the example BS1 game for full memory length which are similar to those obtained when using memory length 2. For all memory lengths and adjustments coefficients of c=[1 0.5] all rational feasible payoffs can only be obtained as equilibrium average payoffs if players incorporate the necessary threat strategies.

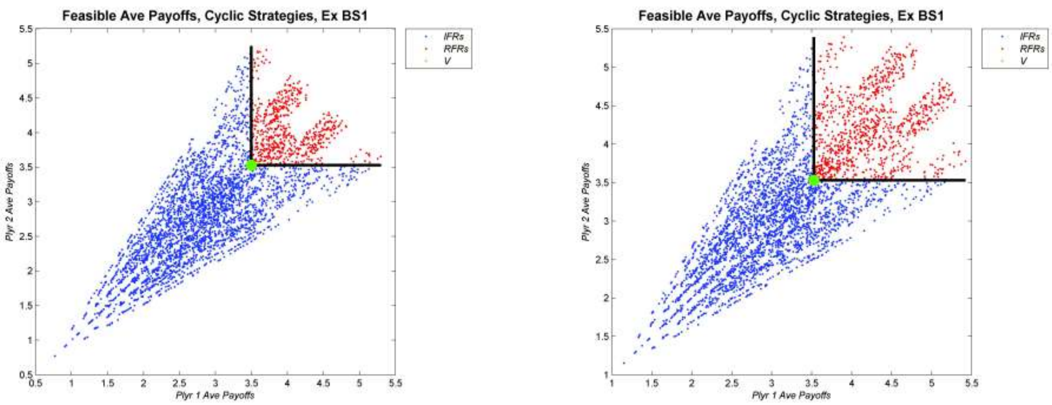
Using different cycle lengths for cyclic strategies leads to different results. Figure 5.22a shows that for cyclic strategy of length 3 and memory length 1 the top of the cone is different from that of strategy length 4 (figure 5.19b). In this case the two middle peaks are still obtained when the players synchronise between entries $\{i, j\} \in [\{1, 1\}, \{2, 2\}]$. The central peak present in average payoffs for strategies of length 4disappears as any synchronization will always lead to either of the entries {1,1} or {2,2} being chosen $\frac{2}{3}$ of the time.

Increasing memory length yields different results, the two middle peaks are stretched upwards as for cycle length 4 due to reduced reductions in the e_t component. Figure 5.22b shows the feasible payoffs for cyclic strategies of length 3 obtained using memory length of 2 to adjust the payoffs.



(a) (b)

Figure 5.21: Cyclic strategies of length 4, full memory length. The 95% confidence interval of the threat point is 3.6388±0.0481



(a) (b)

Figure 5.22: Cyclic strategies are of length 3. The payoffs are adjusted using memory length 1 and full respectively. The 95% confidence interval of the threat point values are (a) 3.1573±0.0499 and (b) 3.6388±0.0481

5.7 Summary

The chapter first presented the design of MATLAB® routines and functions used in this project. Challenges and shortfalls identified during the testing were presented. One of the main issue noted during tests concerns poor efficiency with respect to time of the two routines that carry out exhaustive simulations of given strategy sets *CyclicSimulateRoutine* and *StationarySimulateRoutine*. This limited the maximum number of strategies in each set to 11 for stationary strategies and 16 for cyclic strategies. For both PD and SFW examples, pure stationary strategy pairs led to constant payoffs for all memory lengths whilst mixed strategy pairs led to average payoffs which slightly change with increasing memory length. For mixed strategies where action 1 is played with relatively high probability, the average payoffs apparently decrease with increasing memory length whilst average payoffs seem to increase slightly if action 1 is played with relatively low probability. The average payoffs for the Battle of the Sexes generally increase with increasing memory length. Threat strategies for players in the Battle of the Sexes game involved playing the preferred action with very high probabilities in both states. For Prisoners' Dilemma and Small Fish War games, threat strategies involved choosing action 1 with probability 1 for all memory lengths. Feasible payoffs regions for Small Fish War and Prisoners' Dilemma games are similar to each other for all memory lengths. For the Battle of the Sexes game the feasible payoffs region change for cyclic strategies with increasing memory length. In the Prisoners' Dilemma game the best scenario is for the players to purely use action "cooperate". For the Small Fish War the best compromise is for both players to fish "with restraint" and this yields payoffs which are bigger than when both players resort to fishing "without restraint". In addition to relatively high average payoffs this will also ensure that the system remains viable. The best payoffs for the Battle of the Sexes game are obtained when the two players coordinate to attend the two events equally.

6. Conclusions

6.1 Conclusions

A new game model, Stochastic Games with Frequency Dependent Stage Payoffs, was proposed and developed in this project. The way by which two state stochastic games may arise from one state repeated game was explained for three game models: Battle of the Sexes, Prisoners' Dilemma and the Small Fish War. The game model was motivated by observing that the stage payoffs are not fixed but fluctuate for many repeated games. These fluctuations are a result of externalities generated by the players' choices as well as natural factors which are stochastic. To incorporate the effects of these externalities, relative frequencies by which players have used action 1 for the Prisoners' Dilemma (defecting) and the Small Fish War (fishing "without restraint") were used. For the Battle of the Sexes game the payoffs are adjusted using the relative frequencies of action pairs. Each of the three games applied to this model was then investigated using the limiting average reward criteria to determine three aspects.

The first aspect investigated is the effect of memory length on the average stage payoffs and associated stage payoffs fluctuations. For the Prisoners' Dilemma and the Small Fish War games the average payoffs remained the same for all memory lengths when both players used pure stationary strategies. If any of the players uses a mixed stationary strategy, the average payoffs slightly changes with increasing memory length depending on the probability by which action 1 is played. The same trend is observed for cyclic strategies. For the Battle of the Sexes game, the average payoffs for mixed stationary strategies increase with increasing memory length. For pure strategies the pattern depends on the entries which are implied by the strategy pair. Cyclic strategy pairs yield similar patterns with a special case arising when the players synchronise their cyclic strategies to choose their preferred actions in all the stages. In this case both players obtain payoffs which are almost equal for even memory lengths. For odd memory lengths the average payoffs increase with increasing memory length and subsequently approaching those of even memory lengths.

The project also investigated threat points and strategies for the three game classes. The threat points of the Prisoners' Dilemma and Small Fish War games remain almost the same with increasing memory length. Threat strategies for both players in the two games involve purely playing the action 1 (defecting for the Prisoners' Dilemma and fishing without restraint for the Small Fish War). For the Battle of the Sexes, the threat point increase with increasing memory length and threat strategies involve the players choosing their preferred action with high probability.

The final aspect investigated the nature of feasible average payoffs areas for each type of the three game in which the model was applied. Again the SFW and PD examples yielded similar results with the shape of the feasible average payoffs being convex and similar for both cyclic and stationary strategies. The best compromise in the Prisoners' Dilemma is for the players to purely use action "cooperate". For the Small Fish War the best compromise is for both players to fish "with restraint". In addition to relatively high average payoffs this will also ensure that the system remains viable. The Battle of the Sexes example had different results for the cyclic and stationary strategies. Cyclic strategy pairs yield slightly higher payoffs for all memory lengths with best results obtained when the two players coordinate to attend the two events equally.

6.2 Future Work

The main feature of this model was to incorporate fluctuations arising from externalities associated with the players actions. However several issues need to be addressed for the model to fully replicate the real life scenarios it models. Further research should be done to establish good approximations of the following variables: the adjustment coefficients, rewards levels and transition probabilities. In this research only two classes of strategies were used, it is interesting to see what type of results are obtained when players use other types of strategy pairs, for example behavior strategies. Future research should also expand the model in terms of the number of players, states and actions. Lastly the routines developed for the simulations require improvements for them to handle larger strategy sets and give results in less time.

Bibliography

[1] Michale Breton and Michel Yevenunye Keoula. A great fish war model with asymmetric players. *Ecological Economics*, 97(0):209 – 223, 2014.

[2] B A Cipra. Game theory, population dynamics,and the fish that got away. *SIAM* News, 32(4), 1999.

[3] Elena Denisova and Andrey Garnaev. Fish war: Cooperative and non-cooperative approaches. *Czech Economic Review*, 2(1):028–040, March 2008.

[4] Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

[5] R Joosten. A small fish war: an example with frequency-dependent stage payoffs. *International Journal of Game Theory*, 1(0506), 2005.

[6] R Joosten, T Brenner, and U Witt. Games with frequency-dependent stage payoffs. *International Journal of Game Theory*, 31(4):609–620, 2003.

[7] R Joosten and R Meijboom. Stochastic games with endogenous transitions, November 2010.

[8] David Levhari and Leonard J. Mirman. The Great Fish War: An Example Using a Dynamic Cournot-Nash Solution. *Bell Journal of Economics*, 11(1):322–334, Spring 1980.

[9] Ransom A Myers, Gordon Mertz, Jessica M Bridson, and Michael J Bradford. Simple dynamics underlie sockeye salmon (oncorhynchus nerka) cycles. *Canadian Journal of Fisheries and Aquatic Sciences*, 55(10):2355–2364, 1998.

[10] G. Schoenmakers, J. Flesch, F. Thuijsman, and O.J. Vrieze. Repeated games with bonuses. *Journal of Optimization Theory and Applications*, 136(3):459–473, 2008.

[11] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

A. MATLAB® Functions

A.1 Function PayoffAdjuster

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [M1_1Adjustor M1_2Adjustor M2_1Adjustor M2_2Adjustor] = ...
PayoffAdjuster(rho,c,Type)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Developed by Wellington Mahohoma
%THE FUNCTION CALCULATES THE PAYOFF ADJUSTMENTS FOR STAGE t+1
%Input:
%rho: if Type is PD or SFW then relative frequency for action 1, for BS
%action pairs relative
%frequency
%c: adjusting coefficients
%Type: game type

%Output
% Adjustments matrices for players payoffs for stage t+1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%For either Small Fish Wars or Prisoners' Dilemma gam
if strcmpi('SFW',Type)==1||strcmpi('PD',Type)==1 e
        M1_2Adjustor=c(2)*(sum(rho))*ones(2,2);
        M1_1Adjustor=c(1)*(sum(rho))*ones(2,2);
        M2_1Adjustor=M1_1Adjustor;
        M2_2Adjustor=M1_2Adjustor;
else strcmpi('BS',Type)==1 %for Battle of the Sexes game.
        tt=sum(rho);
red=((rho(2)+rho(3)+rho(6)+rho(7))/tt^2)+(abs(rho(1)+rho(5)-rho(4)-rho(8))/tt);
        ad=(rho(1)+rho(4)+rho(5)+rho(8))/tt;

        M1_1Adjustor=c(1)*[red red-ad; red red];
        M2_1Adjustor=c(1)*[red red-ad; red red];
        M1_2Adjustor=c(2)*[red red-ad; red red];
        M2_2Adjustor=c(2)*[red red-ad; red red];

end


end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A.2 Function StationaryPlay

```
function  [SP_1,SP_2]=...
StationaryPlay(M1_1,M1_2,M2_1,M2_2,Pi,Sigma,P_1,P_2,t0,T,c,memo_length,Type)
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Developed by Wellington Mahohoma
        %THE FUNCTION CALCULATES STAGE PAYOFFS FOR A GIVEN STARTEGY PAIR
        %Input:
```

```
        %M1_1,M1_2: rewards payoff matrices for player 1 in state 1 and 2 .
        %M2_1,M2_2: rewards payoff matrices for player 2 in state 1 and 2 .
        %P_1,P_2: Transtion probability matrices.
        %t0: initial state.
        %T: number of stages.
        %pi: stratetegy for player 1.
        %sigma: stratetegy for player 2.
        %c: adjustments coefficients
        %Type ('BS','PD' and 'SFW'): game type.

        %Output:
        %AvePay1,AvePay2: Average payoffs for players 1 and 2 repectively
        %Strats: Strategy pairs for the simulations.
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %PREALLOCATIONG LOCAL VARIABLES
        States=zeros(1,T);
        Actions=zeros(2,T);
        Action1_Freq=zeros(2,T);
        ActionPairs=zeros(T,8);
        memo_length=[memo_length 2*T];
        SP_1=zeros(size(memo_length,2),T);
        SP_2=zeros(size(memo_length,2),T);


        %initialising the state to t0
        nextState=t0;
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %% GENERATING SEQUENCE OF ACTIONS AND STATES
        for i=1:1:T

                state=nextState;%Updating current state of the game.


                actionToss1=rand; %To be used to decide action for player 1
                actionToss2=rand; %To be used to decide action for player 1

        if state==1 %GAME IN STATE 1
                States(1,i)=1;
                P1_i=(actionToss1>=Pi(1,1))+1; %Player 1's action for stage t
                P2_i=(actionToss2>=Sigma(1,1))+1; %Player 2's action for stage t
                Actions(:,i)=[P1_i;P2_i]; %Storing the actions for players.
                PrNextState=P_1(P1_i,P2_i); %probability to state 1 for stage i+1.


                if P1_i==1&&P2_i==1 %Player 1 plays 1 and player 2 plays 1
                  ActionPairs(t,:)=[1,0,0,0,0,0,0,0];
                  Action1_Freq(:,t)=[1;1];
                  elseif P1_i==1&&P2_i==2 %Player 1 plays 1 and player 2 plays 2.
                        ActionPairs(t,:)=[0,1,0,0,0,0,0,0];
                        Action1_Freq(:,t)=[1;0];
                  elseif P1_i==2&&P2_i==1 %Player 1 plays 2 and player 2 plays 1.
                        ActionPairs(t,:)=[0,0,1,0,0,0,0,0];
                        Action1_Freq(:,t)=[0;1];
                  else %Player 1 plays 2 and player 2 plays right.
                        ActionPairs(t,:)=[0,0,0,1,0,0,0,0];
                        Action1_Freq(:,t)=[0;0];
                  end

        else %THE GAME IS IN STATE TWO
                States(1,i)=2;
                P1_i=(actionToss1>=Pi(1,2))+1;
                P2_i=(actionToss2>=Sigma(1,2))+1;
                Actions(:,i)=[P1_i;P2_i];
                PrNextState=P_2(P1_i,P2_i); %prob to state 1 for stage i+1.

         if P1_i==1&&P2_i==1 %Player 1 plays 1 and player 2 plays 1
                ActionPairs(t,:)=[0,0,0,0,1,0,0,0];
                Action1_Freq(:,t)=[1;1];
```

```
                elseif P1_i==1&&P2_i==2 %Player 1 plays 1 and player 2 plays 2.
                        ActionPairs(t,:)=[0,0,0,0,0,1,0,0];
                        Action1_Freq(:,t)=[1;0];
                elseif P1_i==2&&P2_i==1 %Player 1 plays 2 and player 2 plays 1.
                        ActionPairs(t,:)=[0,0,0,0,0,0,1,0];
                        Action1_Freq(:,t)=[0;1];
                else %Player 1 plays 2 and player 2 plays right.
                        ActionPairs(t,:)=[0,0,0,0,0,0,0,1];
                        Action1_Freq(:,t)=[0;0];

         end
        end

                %SETTING THE STATE FOR THE NEXT STAGE
                StateDecider=rand;
                if PrNextState==0 %Next state is 2.
                        nextState=2;
                elseif PrNextState==1 %Next state is 1.
                        nextState=1;
                else %The  next state is decided by comparing StateDecider to PrNextState.
                        nextState=(StateDecider>=PrNextState)+1;
                end
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CALCULATING STAGE PAYOFFS
        for y=1:1:size(mem_length,2)
                A_S1=M1_1;
                A_S2=M1_2;
                B_S1=M2_1;
                B_S2=M2_2;

                P1_StagePayoffs=zeros(1,T);
                P2_StagePayoffs=zeros(1,T);

        for t=1:1:T
                %Assiging the relevant payoff matrices using current state for stage t.
                if States(t)==1
                        A=A_S1;  B=B_S1;
                else
                        A=A_S2;  B=B_S2;
                end

                %Payoffs for stage t
                P1_StagePayoffs(1,t)=A(Actions(1,t),Actions(2,t));
                P2_StagePayoffs(1,t)=B(Actions(1,t),Actions(2,t));

        %Calculating relative frequency for action 1/action pairs depending on the game
        if strcmpi('BS',Type)==1
                if mem_length(1,y)>t
                        RelFreq=sum(ActionPairs(1:t,:),1);
                else
                        g=t-mem_length(1,y)+1;
                        RelFreq=sum(ActionPairs(g:t,:),1);
                end
        else
                if mem_length(y)>t %Calculating the relative frequency for acti 1
                        RelFreq=sum(Action1_Freq(:,1:t),2)/t;
                else %Calculating the relative frequency for action 1 for t>mem_length.
                        g=t-mem_length(1,y)+1;
                        RelFreq=sum(Action1_Freq(:,g:t),2)/mem_length(1,y);
                end
        end

                %Determining payoff adjustmets for  stage  t+1
                [AH_SA,AL_SA,BH_SA,BL_SA] = PayoffAdjuster(RelFreq,c,Type);

                %Adjusting the payoffs for stage t+1
```

```
                A_S1=M1_1+AH_SA;
                A_S2=M1_2+AL_SA;
                B_S1=M2_1+BH_SA;
                B_S2=M2_2+BL_SA;

        end
        %Storing the stage payoffs for for memory length mem_length(y).
        SP_1(y,:)=P1_StagePayoffs;
        SP_2(y,:)=P2_StagePayoffs;
        end

end
```

A.3 Function StatSimulate

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [AP1,AP2,Strats]= ...
StatSimRouHist(M1_1,M1_2,M2_1,M2_2,P_1,P_2,t0,T,mem_length,R1,R2,n,m,c,Type)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%THE FUNCTION CALCULATES FOR A GIVEN STARTEGY SETS AVERAGE PAYOFFS FOR EACH PLAYER.
%Input:
%M1_1,M1_2: rewards payoff matrices for player 1 in state 1 and 2 respectively.
%M2_1,M2_2: rewards payoff matrices for player 2 in state 1 and 2 respectively.
%P_1,P_2: Transtion probability matrices.
%t0: initial state.
%T: number of stages.
%R1: strategy range for player 1 in states 1 and 2 respectively.
%R2: stratetegy range for player 2 in states 1 and 2 respectively.
%n,m: number of startegies to be generated in the states for players 1 and 2
%        respectively. Both should
%        be 2*1 vectors
%c: adjustments coefficients
%Type ('BS','PD' and 'SFW'): game type.

%Output:
%AP1,AP2: Average payoffs for players 1 and 2 repectively
%Strats: Strategy pairs for the simulations.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CREATING STRATEGY SETS
%Strategies for player 1 in state 1
pi1=linspace(R1(1,1),R1(1,2),n(1));
%Strategies for player 1 in state 2
pi2=linspace(R1(2,1),R1(2,2),n(2));
%Strategies for player 2 in state 1
sigma1=linspace(R2(1,1),R2(1,2),m(1));
%Strategies for player 2 in state 2
sigma2=linspace(R2(2,1),R2(2,2),m(2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PREALLOCATING VARIBLES
rounds=n(1)*n(2)*m(1)*m(2); %number of all possible combinations
rows=size(mem_length,2)+1; %number of memory lengths used
AP1=zeros(rows,rounds); %Average payoffs for player 1
AP2=zeros(rows,rounds); %Average payoffs for player 2
Strats=zeros(4,rounds); %Strategy pairs for players.
k=1;

                for a=1:1:n(1)
          for b=1:1:n(2)
        for c=1:1:m(1)
for d=1:1:m(2)
Pi=[pi1(a) pi2(b)];
```

```
Sigma=[sigma1(c) sigma2(d)];
[SP_1,SP_2]=StationaryPlay(M1_1,M1_2,M2_1,M2_2,Pi,Sigma,P_1,P_2,t0,T,c,mem_length,Type);
AP1(:,k)=mean(SP_1,2);
AP2(:,k)=mean(SP_2,2);
Strats(:,k)=[pi1(a);pi2(b);sigma1(c);sigma2(d)];%Storing strategy pair in column k in
k=k+1

end
    end
      end
        end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A.4 Function CyclicPlay

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function  [SP_1 SP_2]=...
CyclicPlayHistory(M1_1,M1_2,M2_1,M2_2,pi,sigma,P_1,P_2,t0,T,c,mem_length,Type)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%THE FUNCTION CALCULATES STAGE PAYOFFS FOR A GIVEN CYCLIC STARTEGY PAIR
%Input:
%M1_1,M1_2: rewards payoff matrices for player 1 in state 1 and 2 respectively.
%M2_1,M2_2: rewards payoff matrices for player 2 in state 1 and 2 respectively.
%P_1,P_2: Transtion probability matrices.
%t0: initial state.
%T: number of stages.
%pi: stratetegy for player 1.
%sigma: stratetegy for player 2.
%c: adjustments coefficients
%Type ('BS','PD' and 'SFW'): game type.

%Output:
%AvePay1,AvePay2: Average payoffs for players 1 and 2 repectively
%Strats: Strategy pairs for the simulations.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PREALLOCATIONG LOCAL VARIABLES
States=zeros(1,T); %To store states the games visits
Actions=zeros(2,T); %Stores which players choose at each stage
Action1_Freq=zeros(2,T); %Stores each instance action 1 is used
ActionPairs=zeros(T,8); %To store action pairs occurance
mem_length=[mem_length 2*T]; %Adding an entry for full memomry length
SP_1=zeros(size(mem_length,2),T); %stage payoffs for player 1
SP_2=zeros(size(mem_length,2),T); %stage payoffs for player 1


nextState=t0;

pi_length=size(pi,2);
sigma_length=size(sigma,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% GENERATING SEQUENCE OF ACTIONS AND STATES
for t=1:1:T

States(1,t)=nextState;
%Determining the strategy entry index
index_1=mod(t,pi_length);
index_2=mod(t,sigma_length);
if index_1==0
        index_1=pi_length;
end
if index_2==0
```

```
        index_2=sigma_length;
end

%GAME IS IN STATE 1
if States(1,t)==1

        P1_i=pi(2,index_1); %action for player 1 at stage t
        P2_i=sigma(2,index_2); %action for player 2 at stage t
        Actions(:,t)=[P1_i;P2_i];  %Storing actions for stage 1
        PrNextState=P_2(P1_i,P2_i); %probality of game moving to state 1

if P1_i==1&&P2_i==1 %Player 1 plays 1 and player 2 plays 1
        ActionPairs(t,:)=[1,0,0,0,0,0,0,0];
        Action1_Freq(:,t)=[1;1];
elseif P1_i==1&&P2_i==2 %Player 1 plays top and player 2 plays right.
        ActionPairs(t,:)=[0,1,0,0,0,0,0,0];
        Action1_Freq(:,t)=[1;0];
elseif P1_i==2&&P2_i==1 %Player 1 plays 2 and player 2 plays 2.
        ActionPairs(t,:)=[0,0,1,0,0,0,0,0];
        Action1_Freq(:,t)=[0;1];
else %Player 1 plays 2 and player 2 plays right.
        ActionPairs(t,:)=[0,0,0,1,0,0,0,0];
        Action1_Freq(:,t)=[0;0];
end

        %GAME IS IN STATE 2
else
        P1_i=pi(2,index_1); %action for player 1 at stage t
        P2_i=sigma(2,index_2); %action for player 2 at stage t
        Actions(:,t)=[P1_i;P2_i];  %Storing actions for stage 1
        PrNextState=P_2(P1_i,P2_i); %probality for the game to state 1

if P1_i==1&&P2_i==1 %Player 1 plays 1 and player 2 plays 1
        ActionPairs(t,:)=[0,0,0,0,1,0,0,0];
        Action1_Freq(:,t)=[1;1];
elseif P1_i==1&&P2_i==2 %Player 1 plays 1 and player 2 plays 2.
        ActionPairs(t,:)=[0,0,0,0,0,1,0,0];
        Action1_Freq(:,t)=[1;0];
elseif P1_i==2&&P2_i==1 %Player 1 plays 2 and player 2 plays 1.
        ActionPairs(t,:)=[0,0,0,0,0,0,1,0];
        Action1_Freq(:,t)=[0;1];
else %Player 1 plays 2 and player 2 plays right.
        ActionPairs(t,:)=[0,0,0,0,0,0,0,1];
        Action1_Freq(:,t)=[0;0];
end
end

%SETTING THE STATE FOR THE NEXT STAGE
StateDecider=rand;
if PrNextState==0 %Next state is 2.
        nextState=2;
elseif PrNextState==1 %Next state is 1.
        nextState=1;
else %next state is decided by comparing StateDecider to PrNextState.
        nextState=(StateDecider>=PrNextState)+1;
end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CALCULATING STAGE PAYOFFS
 for y=1:1:size(mem_length,2)
        A_S1=M1_1;
        A_S2=M1_2;
        B_S1=M2_1;
        B_S2=M2_2;

        P1_StagePayoffs=zeros(1,T);
        P2_StagePayoffs=zeros(1,T);
```

```

for t=1:1:T
   %Assiging the relevant payoff matrices using current state for stage t.
        if States(t)==1
                A=A_S1;  B=B_S1;
        else
                A=A_S2;  B=B_S2;
        end

        %Payoffs for stage t
        P1_StagePayoffs(1,t)=A(Actions(1,t),Actions(2,t));
        P2_StagePayoffs(1,t)=B(Actions(1,t),Actions(2,t));

%Calculating relative frequency for action 1/ pairs depending on the game
if strcmpi('BS',Type)==1
        if mem_length(1,y)>t
                RelFreq=sum(ActionPairs(1:t,:),1);
        else %Calculating the relative frequency for act pairs for t>mem_length
                g=t-mem_length(1,y)+1;
                RelFreq=sum(ActionPairs(g:t,:),1);
        end
else
        if mem_length(y)>t %Calculating the rel freq of act 1 for t=<mem_length.
                RelFreq=sum(Action1_Freq(:,1:t),2)/t;
        else %Calculating the relative frequency for action 1 for t>mem_length.
                g=t-mem_length(1,y)+1;
                RelFreq=sum(Action1_Freq(:,g:t),2)/mem_length(1,y);
        end
end

        %Determining payoff adjustmets for  stage  t+1
        [AH_SA,AL_SA,BH_SA,BL_SA] = PayoffAdjuster(RelFreq,c,Type);

%Adjusting the payoffs for stage t+1
        A_S1=M1_1+AH_SA;
        A_S2=M1_2+AL_SA;
        B_S1=M2_1+BH_SA;
        B_S2=M2_2+BL_SA;

end
%Storing the stage payoffs for for memory length mem_length(y).
SP_1(y,:)=P1_StagePayoffs;
SP_2(y,:)=P2_StagePayoffs;
end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A.5 Function CycSimulate

```
function [AvePay1,AvePay2,Strats]=...
CycSimulate(M1_1,M1_2,M2_1,M2_2,P_1,P_2,t0,T,mem_length,pi1,pi2,sigma1,sigma2,c,Type)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%THE FUNCTION CALCULATES FOR A GIVEN STARTEGY SETS AVERAGE PAYOFFS FOR EACH PLAYER.
%Input:
%M1_1,M1_2: rewards payoff matrices for player 1 in state 1 and 2 respectively.
%M2_1,M2_2: rewards payoff matrices for player 2 in state 1 and 2 respectively.
%P_1,P_2: Transtion probability matrices.
%t0: initial state.
%T: number of stages.
%pi1,pi2: stratetegy sets for player 1 in states 1 and 2 respectively.
%sigma1,sigma2: stratetegy sets for player 2 in states 1 and 2 respectively.
```

```
%c: adjustments coefficients
%Type ('BS','PD' and 'SFW'): game type.

%Output:
%AvePay1,AvePay2: Average payoffs for players 1 and 2 repectively
%Strats: Strategy pairs for the simulations.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n1=size(pi1,1); %number of strategies in state 1 for player 1
n2=size(pi2,1); %number of strategies in state 2 for player 1
m1=size(sigma1,1); %number of strategies in state 1 for player 2
m2=size(sigma2,1); %number of strategies in state 2 for player 2
rows=size(mem_length,2)+1; %number of average payoff sets
rounds=n1*n2*m1*m2; %number of all combinations of strategies
AvePay1=zeros(rows,rounds); %preallocating average payoffs for player 1
AvePay2=zeros(rows,rounds); %preallocating average payoffs for player 1
Strats=zeros(4,rounds); %storing strategy pairs for simulation k
k=1; %simulation index
                for a=1:1:n1
            for b=1:1:n2
        for c=1:1:m1
    for d=1:1:m2
Pi=[pi1(a,:); pi2(b,:)]; %Strategy for player 1
Sigma=[sigma2(c,:); sigma2(d,:)]; %Strategy for player 2
[SP_1,SP_2]=CyclicPlay(M1_1,M1_2,M2_1,M2_2,Pi,Sigma,P_1,P_2,t0,T,c,mem_length,Type);
AvePay1(:,k)=mean(SP_1,2);
AvePay2(:,k)=mean(SP_2,2);
Strats(:,k)=[a;b;c;d];
k=k+1
  end
        end
            end
                end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

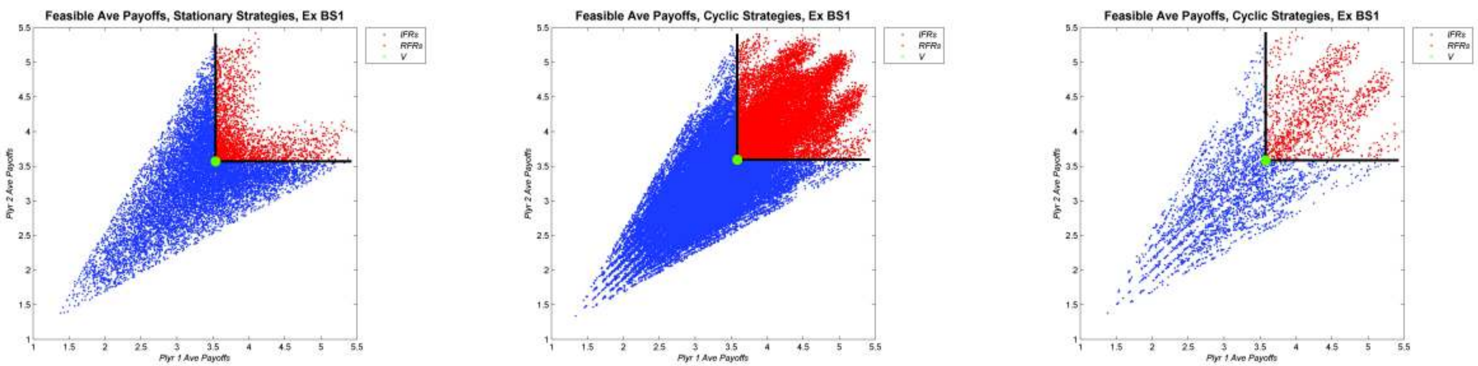
A.6 Function Threats

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [V_1,V_2,Pie_star,Sigma_star]=...
Threats(AvePayoffs_P1,AvePayoffs_P2,Strats)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Developed by Wellington Mahohoma
%The function detemines for a given sets of payoffs for each player and the
%associated strategy pairs the threat point and threat strategies.The
%strategy sets used to produce the payoffs should cover the entire range
%from 0 to 1.
%INPUT:
%AvePayoffs_P1: Average payoffs for player 1.
%AvePayoffs_P2: Average payoffs for player 2.
%Strats: strategy pairs that produced the average payoffs sequences.
%OUTPUT:
%               V_1,V_2: Threat point values for player 1 and 2 respectively.
%       Pie_star,Sigma_star: threat strategies
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Preallocating varibales for speed
V_1=zeros(1,size(AvePayoffs_P1,1)); %Threat point value vector for player 1
V_2=zeros(1,size(AvePayoffs_P1,1));       %Threat point value vector for player 2
Pie_star=zeros(4,size(AvePayoffs_P1,1)); %Threat strategy matrix for player 1
Sigma_star=zeros(4,size(AvePayoffs_P1,1)); %Threat strategy matrix for player 2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Threat point for player 2. The payoffs from each strategy from player 1 and the
%enetrie strategies of player 2 are grouped together and the maximum is stored.
```

```
%At each iteration if the maximum is less than the current threat value the threat
%         value is updated.
a=size(Strats,2);
b=a^0.25;
c=b^2;
for i=1:1:size(AvePayoffs_P1,1)
u=1;
TP=10000000; %Initialising local varible of the threat point to some big number.
e=1;
        for l=1:1:c
        maxS = max(AvePayoffs_P2(i,e:u*c),[],2);
        if TP>=maxS
                TP=maxS;
                V_2(1,i)=maxS;
                Pie_star(:,i)=Strats(:,e);
        end
        u=u+1;
        e=e+c;
        end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Threat point for player 1. The payoffs from each strategy from player 2 and the
%enetrie strategies of player 1 are grouped together and the maximum is stored.
%At each iteration if the maximum is less than the current threat value the threat
%         value is updated.
for i=1:1:size(AvePayoffs_P1,1)
        TP=1000000; %Initialising local varible of the threat point to some big number.
        for g=1:1:c
                E2=zeros(1,c);
                f=g;
                r=1;

                %Grouping the payoffs from staretgy i of P 2 and all startegies of P 1
        while f<=a
                E2(1,r)=AvePayoffs_P1(i,f);
                f=f+c;
                r=r+1;
        end
        maxT = max(E2,[],2);
        if TP>=maxT
                TP=maxT;
                V_1(1,i)=maxT;
                Sigma_star(:,i)=Strats(:,g);
        end

        end

end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

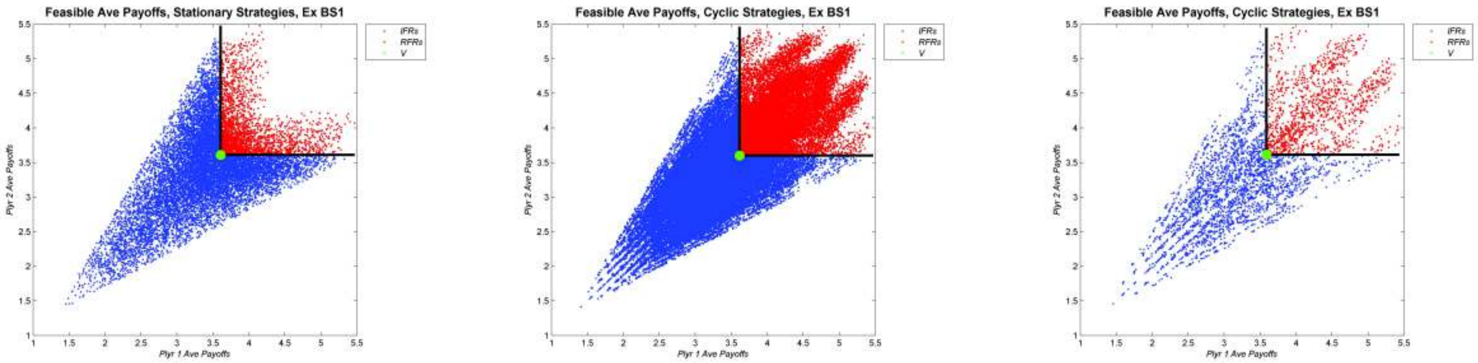
B. Appendix: Results

B.1 Battle of the Sexes



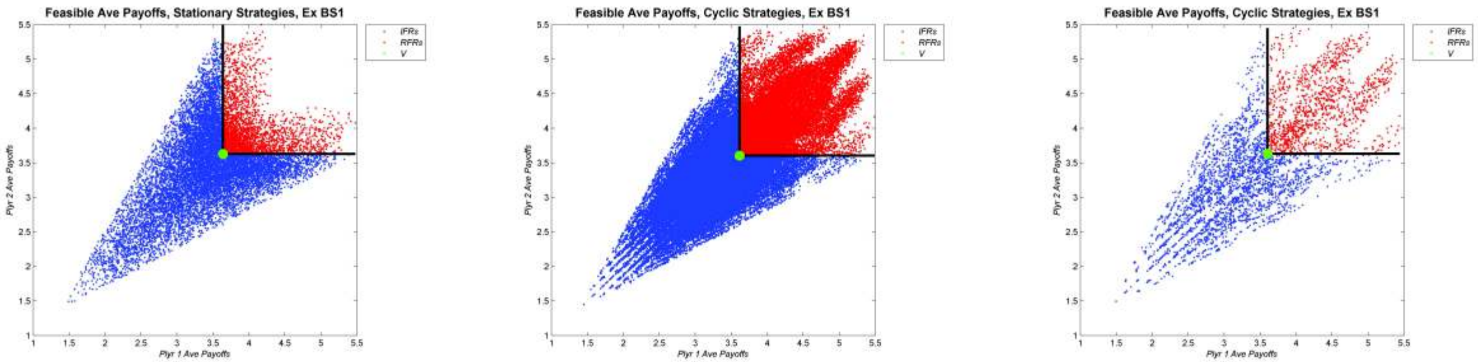
(a) (b) (c)

Figure B.1: Memory Length 5,(a) stationary strategies,(b) cyclic strategies of length 4 and (c) cyclic strategies of length 3.



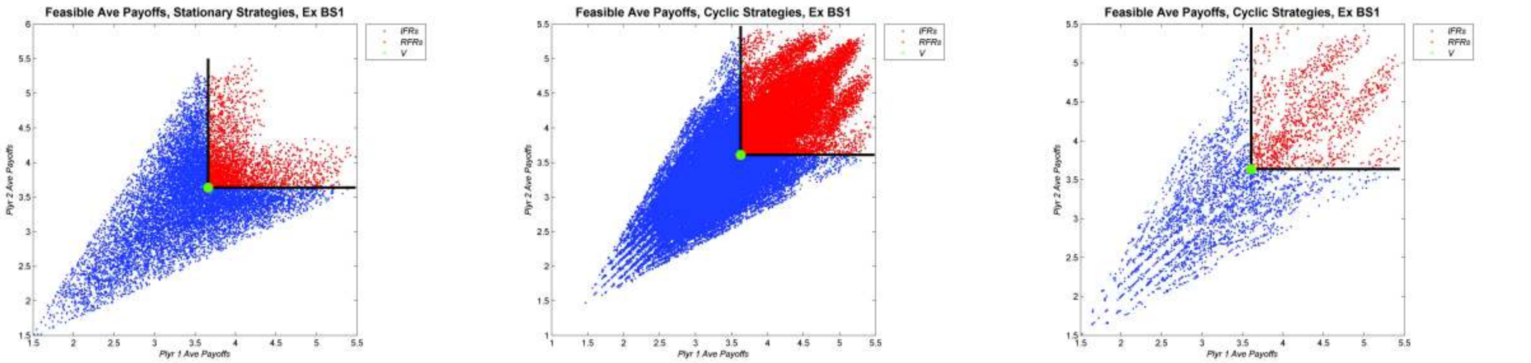
(a) (b) (c)

Figure B.2: Memory Length 10,(a) stationary strategies,(b) cyclic strategies of length 4 and (c) cyclic strategies of length 3.



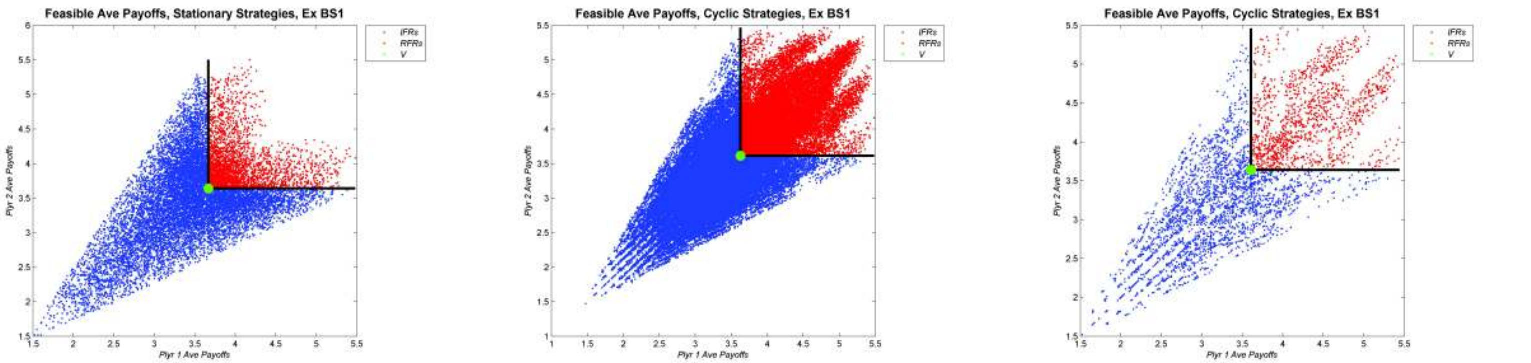
(a) (b) (c)

Figure B.3: Memory Length 20,() stationary strategies,(b) cyclic strategies of length 4 and (c) cyclic strategies of length 3.



(a) (b) (c)

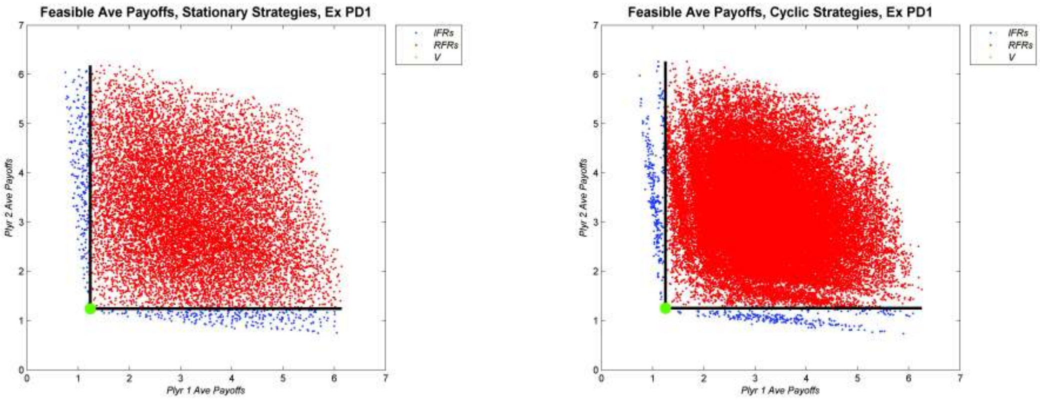
Figure B.4: Memory Length 50, (a) stationary strategies,(b) cyclic strategies of length 4 and (c) cyclic strategies of length 3.



(a) (b) (c)

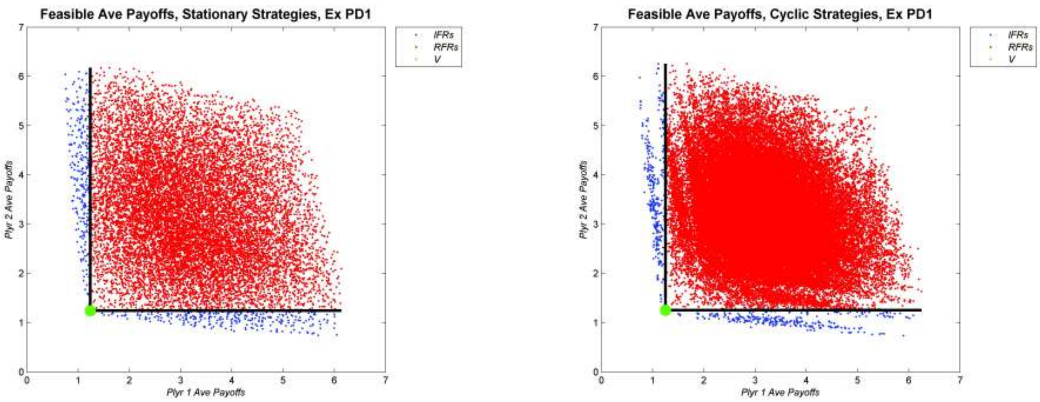
Figure B.5: Memory Length 75, (a) stationary strategies,(b) cyclic strategies of length 4 and (c) cyclic strategies of length 3.

B.2 Prisoners' Dilemma



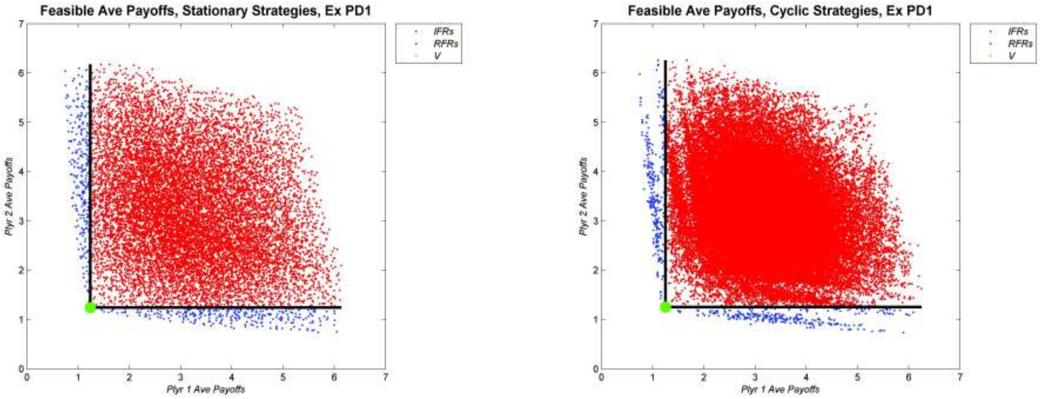
(a) (b)

Figure B.6: Memory Length 2, cyclic strategies are of length 4



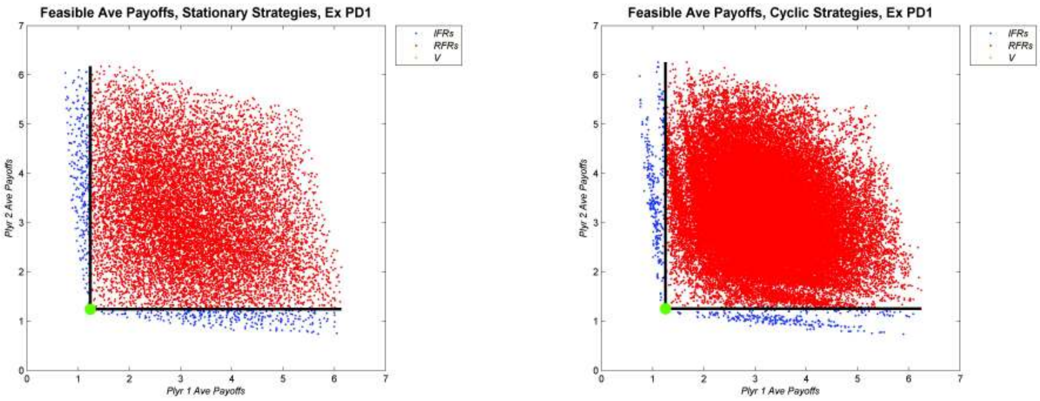
(a) (b)

Figure B.7: Memory Length 5,cyclic strategies are of length 4



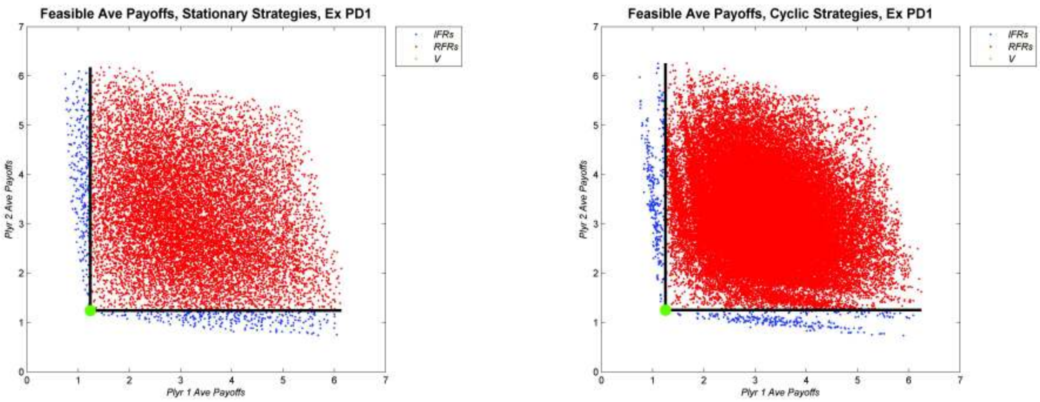
(a) (b)

Figure B.8: Memory Length 10,cyclic strategies are of length 4



(a) (b)

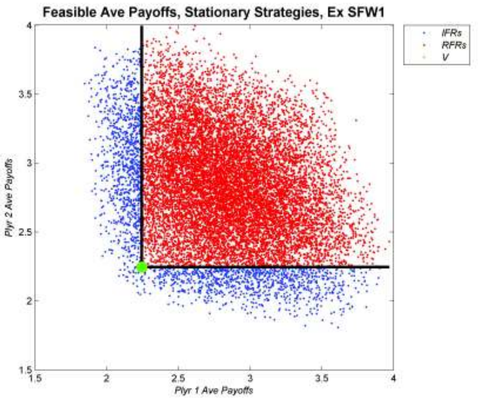
Figure B.9: Memory Length 20,cyclic strategies are of length 4



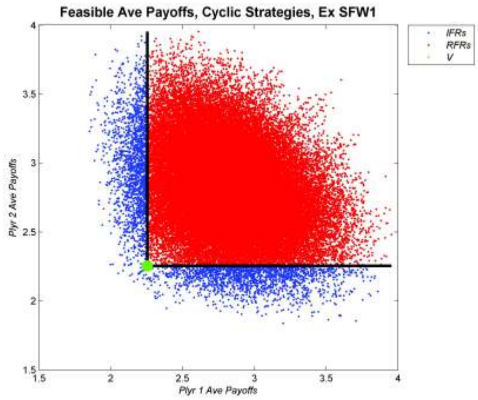
(a) (b)

Figure B.10: Memory Length 50,cyclic strategies are of length 4

B.3 Small Fish War

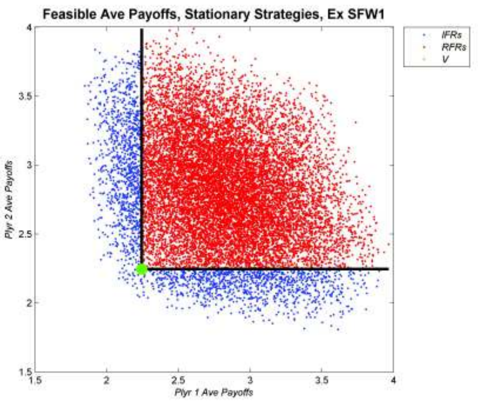


(a)

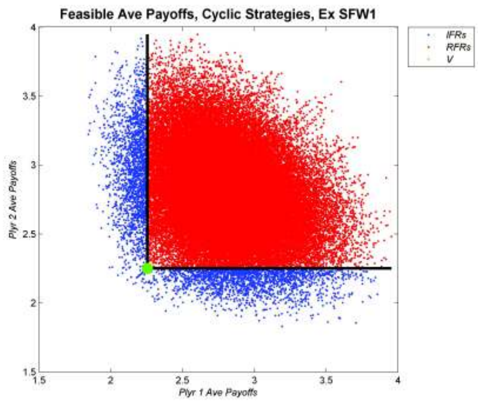


(b)

Figure B.11: Memory Length 2, cyclic strategies are of length 4

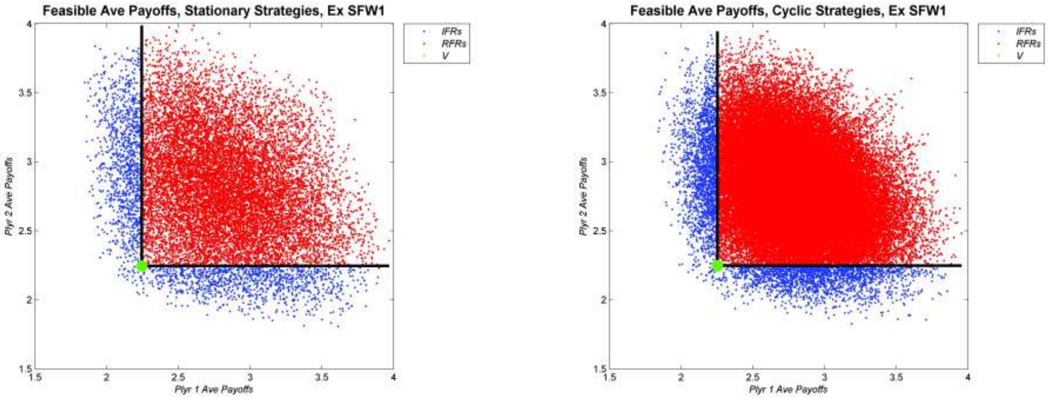


(a)



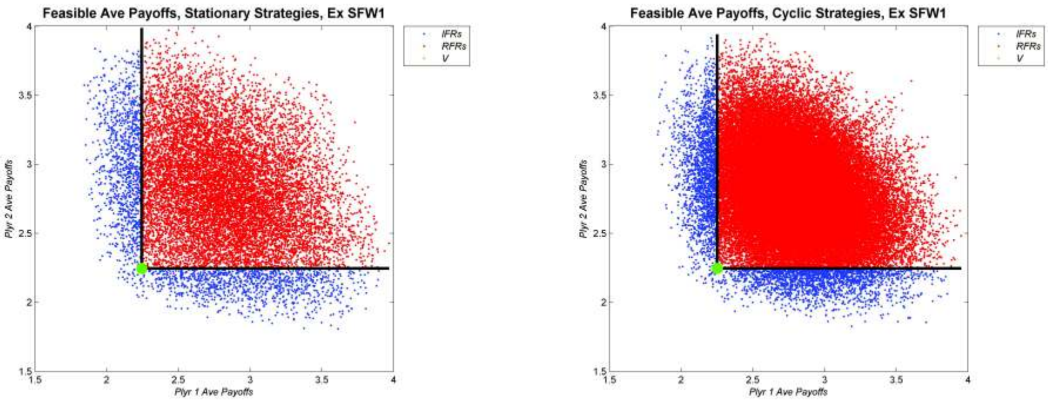
(b)

Figure B.12: Memory Length 5, cyclic strategies are of length 4.



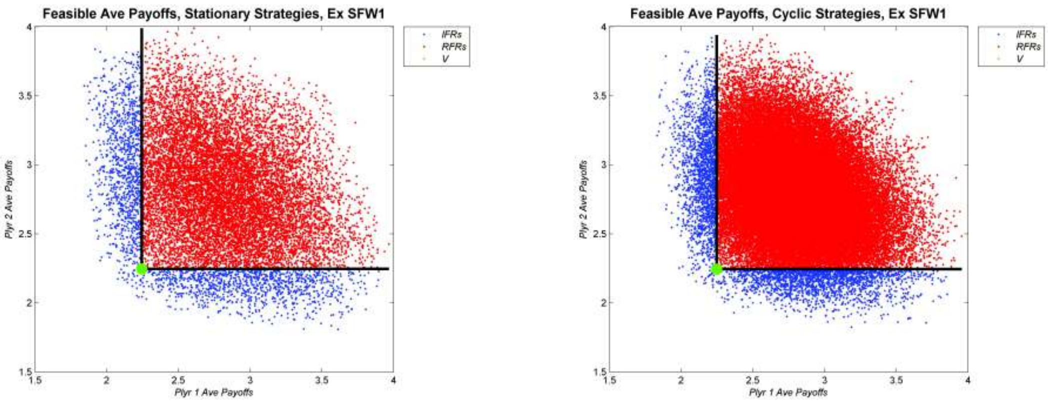
(a) (b)

Figure B.13: Memory Length 10, cyclic strategies are of length 4



(a) (b)

Figure B.14: Memory Length 20, cyclic strategies are of length 4



(a) (b)

Figure B.15: Memory Length 50, cyclic strategies are of length 4