Stability of defection, optimisation of strategies and the limits of memory in the Prisoner's Dilemma.

Nikoleta E. Glynatsi Vincent A. Knight

Abstract

In this manuscript we build upon a framework provided in 1989 to study best responses in the well known memory one strategies of the Iterated Prisoner's Dilemma. The aim of this work is to construct a compact way of identifying best responses of short memory strategies and to show their limitations in multi-opponent interactions. A number of theoretic results are presented.

1 Introduction

The Prisoner's Dilemma (PD) is a two player person game used in understanding the evolution of co-operative behaviour. Each player can choose between cooperation (C) and defection (D). The decisions are made simultaneously and independently. The normal form representation of the game is given by:

$$S_p = \begin{pmatrix} R & S \\ T & P \end{pmatrix} \quad S_q = \begin{pmatrix} R & T \\ S & P \end{pmatrix} \tag{1}$$

where S_p represents the utilities of the row player and S_q the utilities of the column player. The payoffs, (R, P, S, T) (the most common values used in the literature are $(3, 1, 0, 5)$ [4]), are constrained by equations (2) and (3). Constraint (2) ensures that defection dominates cooperation and constraint (3) ensures that there is a dilemma; the sum of the utilities for both players is better when both choose to cooperate.

$$T > R > P > S \tag{2}$$

$$2R > T + S \tag{3}$$

The PD is a one shot game, however it is commonly studied in a manner where the history of the interactions matter. The repeated form of the game is called the Iterated Prisoner's Dilemma (IPD) and in the 1980s following the work of [5, 6] it attracted the attention of the scientific community.

In [5] and [6], the first well known computer tournaments of the IPD were performed. A total of 13 and 63 strategies were submitted in computer code and competed against each other in a round robin tournament. All contestants competed against each other, a copy of themselves and a random strategy. The winner was decided on the average score a strategy achieved and not the total number of wins. How many turns of history that a strategy would use, the memory size, was left to the creator of the strategy to decide.

The winning strategy of both tournaments was a strategy called Tit for Tat. Tit for Tat is a strategy which starts by cooperating and then mimics the last move of it's opponent. This is a strategy which makes use of the previous move of the opponent only and a reactive strategy. In [20] a framework for studying such strategies was introduced. This was later used to introduce well known reactive strategies such as Generous Tit For Tat [21].

Reactive strategies are a subset of memory one strategies. Memory one strategies similarly are only concerned with the previous turn. However, they take into consideration both players' recent moves to decide on an action. Several successful memory one strategies are found in the literature, for example Pavlov [18].

A well known set of memory on strategies was introduced in [22], these were called zero determinant (ZD) strategies. The ZD strategies manage to force a linear relationship between the score of the strategy and the opponent. According to [22] the ZD strategies can dominate any evolutionary opponent in pairwise interactions by using a single slot of memory. Thus the usefulness of memory in the IPD was questioned.

The ZD strategies attracted a lot of attention. It was stated that "Press and Dyson have fundamentally changed the viewpoint on the Prisoner's Dilemma" [24]. In [24] a very similar tournament to Axelrod's tournament is run including ZD strategies and a new set of ZD strategies the Generous ZD. One specific advantage of memory one strategies is their mathematical tractability. As described in Section 2 they can be represented completely as an element of $\mathbb{R}^4$.

Even so, ZD and memory one strategies have also received criticism. In [17], the 'memory of a strategy does not matter' statement was questioned. A set of more complex strategies, strategies that take in account the entire history set of the game, were trained and proven to be more robust against multiple opponents.

The purpose of this work is to consider a given memory one strategy in a similar fashion to [22]. However whilst [22] found a way for a player to manipulate a given opponent, this work will consider a multidimensional optimisation approach to identify the best response to a group of opponents. In essence the aim is to produce a compact method of identifying the best memory one strategy against a given opponent.

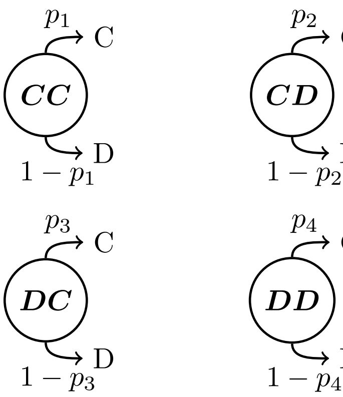
In the second part of this manuscript we explore the limitation of these best response strategies. This is achieved by comparing the performance of an optimal memory one strategy, for a given environment, with the performance of a more complex strategy that has a larger memory.

One particular benefit of this analysis is the identification of conditions for which defection is a best response. Thus, identifying environments for which cooperation can not occur.

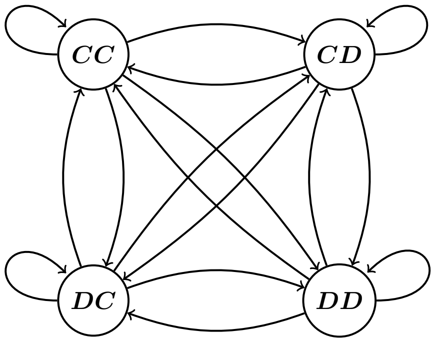
2 The utility against memory one players

In [22] a framework is described to sufficient study the interactions of memory one strategies modelled as a stochastic process. In this manuscript it is stated that if a strategy is concerned with only the outcome of a single turn then there are four possible 'states' the strategy could be in. These are CC, CD, DC, CC. A memory one strategy is denoted by the probabilities of cooperating after each of these states, $p = (p_1, p_2, p_3, p_4) \in \mathbb{R}_{[0,1]}^4$. A diagrammatic representation of such strategy is given in Figure 1a.

Moreover, if two players are moving from state to state following a general memory one strategy this can be modelled as a Markov process. A diagrammatic representation of the Markov chain is shown in Figure 1b. The corresponding transition matrix M given by:



(a) Diagrammatic representation of a memory one strategy.



(b) Markov chain on a PD game.

$$M = \begin{bmatrix} p_1q_1 & p_1\left(-q_1+1\right) & q_1\left(-p_1+1\right) & \left(-p_1+1\right)\left(-q_1+1\right) \\ p_2q_3 & p_2\left(-q_3+1\right) & q_3\left(-p_2+1\right) & \left(-p_2+1\right)\left(-q_3+1\right) \\ p_3q_2 & p_3\left(-q_2+1\right) & q_2\left(-p_3+1\right) & \left(-p_3+1\right)\left(-q_2+1\right) \\ p_4q_4 & p_4\left(-q_4+1\right) & q_4\left(-p_4+1\right) & \left(-p_4+1\right)\left(-q_4+1\right) \end{bmatrix}. \tag{4}$$

The long run steady state probability v is the solution to $\triangledown M = \triangledown$ (given in the Appendix). Combining the stationary vector v with the payoff matrices of equation (1) allow us to retrieve the expected payoffs for each player. Thus, the utility for player p against q, denoted as $u_q(p)$, is defined by,

$$u_q(p) = v \times (R, P, S, T). \tag{5}$$

Here we present the first theoretical result which concerns the form of $u_q(p)$. That is that $u_q(p)$ is given by a ratio of two quadratic forms [15], as presented by Theorem 1.

Theorem 1. *The expected utility of a memory one strategy $p \in \mathbb{R}^4_{[0,1]}$ against a memory one opponent strategy $q \in \mathbb{R}^4_{[0,1]}$, denoted as $u_q(p)$, can be written as a ratio of two quadratic forms:*

$$u_q(p) = \frac{\frac{1}{2}pQp^T + cp + a}{\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a}}, \tag{6}$$

where $Q, \bar{Q} \in \mathbb{R}^{4\times4}$ matrices defined by the transition probabilities of the opponent q_1, q_2, q_3, q_4 as follows:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{-\left(q_1-q_3\right)\left(q_2-5q_4-1\right)} & \boldsymbol{q_3\left(q_1-q_2\right)} & \boldsymbol{-5q_3\left(q_1-q_4\right)} \\ \boldsymbol{-\left(q_1-q_3\right)\left(q_2-5q_4-1\right)} & \boldsymbol{0} & \boldsymbol{\left(q_2-q_3\right)\left(q_1-3q_4-1\right)} & \boldsymbol{\left(q_3-q_4\right)\left(5q_1-3q_2-2\right)} \\ \boldsymbol{q_3\left(q_1-q_2\right)} & \boldsymbol{\left(q_2-q_3\right)\left(q_1-3q_4-1\right)} & \boldsymbol{0} & \boldsymbol{3q_3\left(q_2-q_4\right)} \\ \boldsymbol{-5q_3\left(q_1-q_4\right)} & \boldsymbol{\left(q_3-q_4\right)\left(5q_1-3q_2-2\right)} & \boldsymbol{3q_3\left(q_2-q_4\right)} & \boldsymbol{0} \end{bmatrix}, \tag{7}$$

$$\bar{Q} = \begin{bmatrix} 0 & -(q_1-q_3)(q_2-q_4-1) & (q_1-q_2)(q_3-q_4) & (q_1-q_4)(q_2-q_3-1) \\ -(q_1-q_3)(q_2-q_4-1) & 0 & (q_2-q_3)(q_1-q_4-1) & (q_1-q_2)(q_3-q_4) \\ (q_1-q_2)(q_3-q_4) & (q_2-q_3)(q_1-q_4-1) & 0 & -(q_2-q_4)(q_1-q_3-1) \\ (q_1-q_4)(q_2-q_3-1) & (q_1-q_2)(q_3-q_4) & -(q_2-q_4)(q_1-q_3-1) & 0 \end{bmatrix}. \quad (8)$$

c and $\bar{c} \in \mathbb{R}^{4\times 1}$ defined by:

$$c = \begin{bmatrix} q_1(q_2-5q_4-1) \\ -(q_3-1)(q_2-5q_4-1) \\ -q_1q_2+q_2q_3+3q_2q_4+q_2-q_3 \\ 5q_1q_4-3q_2q_4-5q_3q_4+5q_3-2q_4 \end{bmatrix}, \quad (9)$$

$$\bar{c} = \begin{bmatrix} q_1(q_2-q_4-1) \\ -(q_3-1)(q_2-q_4-1) \\ -q_1q_2+q_2q_3+q_2-q_3+q_4 \\ q_1q_4-q_2-q_3q_4+q_3-q_4+1 \end{bmatrix}. \quad (10)$$

and $a = -q_2 + 5q_4 + 1$ and $\bar{a} = -q_2 + q_4 + 1$.

Proof: The proof is given in Appendix.

Figure 2 indicates that the formulation of $u_q(p)$ as a quadratic ratio successfully captures the simulated behaviour. A data set offering further validation is available at.

The simulated utility, denoted as $U_q(p)$ has been calculated using [1]. It is an open research framework for the study of the IPD and is described in [16]. Note that when referring to $U_q(p)$ here onwards we mean the simulated utility calculated with [1].

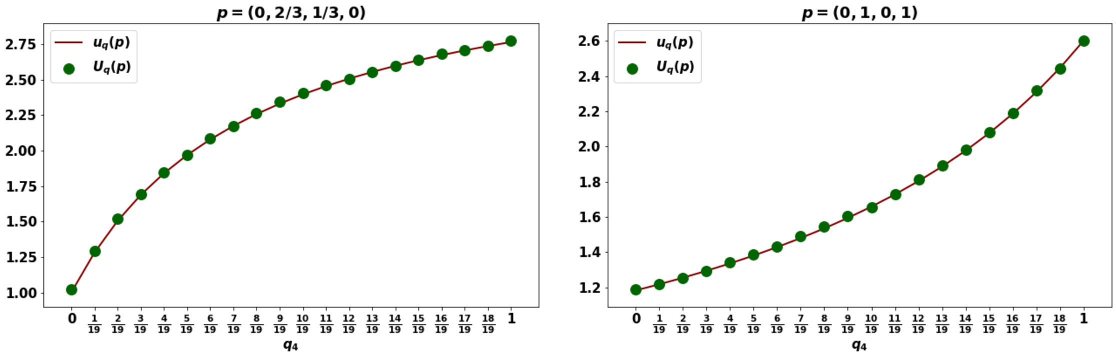


Figure 2: Differences between simulated and analytical results for $q = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, q_4)$.

Moreover Theorem 1 can be expanded to multi opponent interactions. The IPD is commonly studied in tournaments where a strategy interacts with a number of opponents. There the payoff of a player is given by the average payoffs the player achieved. More specifically the expected utility of a memory one strategy against a N number of opponents is given by Theorem 2.

Theorem 2. *The expected utility of a memory one strategy $p \in \mathbb{R}^4_{[0,1]}$ against a group of opponents $q^{(1)}, q^{(2)}, \dots, q^{(N)}$, denoted as $\frac{1}{N}\sum\limits_{i=1}^{N} {u_q}^{(i)}(p)$ is given by:*

$$\frac{1}{N}\sum_{i=1}^{N} {u_q}^{(i)}(p) = \frac{1}{N} \frac{\sum\limits_{i=1}^{N} (\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)}) \prod\limits_{\substack{j\ =\ 1 \\ j\ \neq\ i}}^{N} (\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)})}{\prod\limits_{i=1}^{N} (\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)})}. \tag{11}$$

To validate the formulation of Theorem 2 we calculate the simulated and the theoretical payoffs of several memory one strategies against a set of 10 opponents. The opponents used are the memory one strategies for the tournament conducted in [24]. The names and a small explanation of the strategic rules are given by Table 7, Appendix A. The values of $\frac{1}{10}\sum\limits_{i=1}^{10} u_{q^{(i)}}(p)$ and $\frac{1}{10}\sum\limits_{i=1}^{10} U_{q^{(i)}}(p)$ match (Table 7), thus we conclude that the formulation of Theorem 2 is correct.

	p_1	p_2	p_3	p_4	$\frac{1}{10}\sum\limits_{i=1}^{10} u_q^{(i)}(p)$	$\frac{1}{10}\sum\limits_{i=1}^{10} U_q^{(i)}(p)$
0	0.0	$\frac{1}{3}$	$\frac{1}{3}$	1.0	2.158	2.166
1	0.0	0.0	$\frac{1}{3}$	1.0	2.165	2.173
2	0.0	$\frac{1}{3}$	1.0	1.0	2.149	2.157
3	0.0	$\frac{1}{3}$	$\frac{2}{3}$	1.0	2.139	2.149
4	0.0	0.0	0.0	$\frac{2}{3}$	2.191	2.200
5	0.0	$\frac{1}{3}$	1.0	$\frac{2}{3}$	2.157	2.167
6	0.0	0.0	$\frac{2}{3}$	1.0	2.156	2.166
7	0.0	0.0	$\frac{2}{3}$	$\frac{2}{3}$	2.145	2.156
8	0.0	$\frac{1}{3}$	0.0	$\frac{2}{3}$	2.199	2.211
9	0.0	0.0	1.0	$\frac{1}{3}$	2.186	2.198

Table 1: Results of memory one strategies against the strategies in Table 7.

The analytical formulation of Theorem 2 will be used in the following sections to explore the best response to memory one strategies.

3 Best responses to memory one players

In the introduction a question was raised: which memory one strategy is the **best response** to a group of memory one strategies? This will be considered as an optimisation problem, where a memory one strategy p wants to optimise it's average utility $\frac{1}{N}\sum u_{q^{(i)}}(p)$ against a set opponents $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$.

The decision variable is the vector p and the solitary constraint is that $p \in \mathbb{R}^4_{[0,1]}$. The optimisation problem is given by (12). Note that we will considering the sum and not the average utility as their optimisation corresponds to the same thing.

$$\max_{p} : \sum_{i=1}^{N} {u_q}^{(i)}(p) \qquad (12)$$
$$\text{such that} : p \in \mathbb{R}_{[0,1]}$$

This work is concerned with an optimisation problem of a ratio of quadratic forms. It can be verified empirically for the case of a single opponent that there exist at least one point for which the definition of concavity does not hold.

There is some work on the optimisation on non concave ratios of of quadratic forms [7, 9]. In these both the numerator and the denominator of the fractional problem were concave which is not true for our case. These results are established in Theorem 3.

Theorem 3. *The utility of a player p against an opponent q, $u_q(p)$ given by (6), is not concave. Furthermore neither the numeration or the denominator of (6), are concave.*

Proof. A function $f(x)$ is said to be concave on an interval $[a, b]$ if, for any points x_1 and $x_2 \in [a, b]$, the function $-f(x)$ is convex on that interval.

A function $f(x)$ is convex on an interval $[a, b]$ if for any two points x_1 and x_2 in $[a, b]$ and any λ where $0 < \lambda < 1$,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \qquad (13)$$

Using numerical trials we have shown that there exists at least on point for which (6) is not concave. Let f be $u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}$ it can shown that for $x_1 = (\frac{1}{4}, \frac{1}{2}, \frac{1}{5}, \frac{1}{2}), x_2 = (\frac{8}{10}, \frac{1}{2}, \frac{9}{10}, \frac{7}{10})$ and $\lambda = \frac{1}{10}$ condition (13) does not hold as:

$$1.49 \geq 1.48$$

Moreover, any potential benefits from the structure of the numerator and the denominator are also investigated. In [3] it is stated that a quadratic form will be concave if and only if it's symmetric matrix is negative semi definite. A matrix$A$ is semi-negative definite if:

$$|A|_i \leq 0 \text{ for } i \text{ is odd and } |A|_i \geq 0 \text{ for } i \text{ is even.} \qquad (14)$$

For both Q and $\bar{Q}$ it is exhibited that for $i = 2$ (odd):

$$|Q|_2 = -\left(q_1 - q_3\right)^2 \left(q_2 - 5q_4 - 1\right)^2,$$
$$|\bar{Q}|_2 = -\left(q_1 - q_3\right)^2 \left(q_2 - q_4 - 1\right)^2$$

both determinants are negative, thus the concavity condition (14) fails for both quadratic forms. □

The non concavity of $u(p)$ indicates multiple local optimal points. Thus we are not searching a single optimal point but a set of candidate optimal points. The aim is to introduce a compact way of constructing the candidate set. Once the set is defined the point that maximises (11) corresponds to the best response strategy, thus any search over an infinitely sized continuous set can instead be replaced with a discrete set.

The problem considered is a bounded because $p \in \mathbb{R}^4_{[0,1]}$. It is known that the candidate solutions will exist either at the boundaries of the feasible solution space, or within that space. The method of Lagrange Multipliers [8] and Karush-Kuhn-Tucker conditions [11] are based on this. The Karush-Kuhn-Tucker conditions are used here because the constraints are inequalities.

Note that the best response can not be captured by optimising against the mean opponent.

$$\frac{1}{N}\sum_{i=1}^{N} {u_q}^{(i)}(p) \neq u_{\frac{1}{N}\sum\limits_{i=1}^{N} q^{(i)}}(p). \tag{15}$$

A number of numerical experiments have been performed for cases where $p = (p, p, p, p)$ and $p = (p_1, p_2, p_1, p_2)$. This was done in order to compare the right hand side of equation (15) to the left. The fact that equation (15) holds is evident by Figure 3.

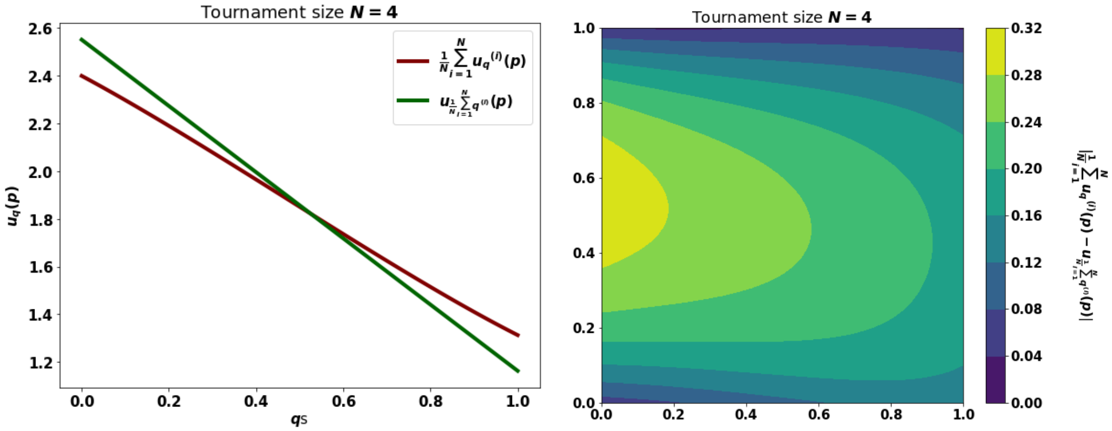


Figure 3: An example confirming equation (15) for $p = (p, p, p, p)$ and $p = (p_1, p_2, p_1, p_2)$.

The above discussion leads to Lemma 4 which presents the best memory one response to a group of opponents.

Lemma 4. *The optimal behaviour of a memory one strategy player* $p^* \in \mathbb{R}^4_{[0,1]}$ *against a set of* N *opponents* $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ *for* $q^{(i)} \in \mathbb{R}^4_{[0,1]}$ *is established by:*

$$p^* = \text{argmax}(\sum_{i=1}^{N} u_q(p)), \ \ p \in S_q,$$

where the set S_q *is defined as*

$$S_q = \{0, \bar{p_i}, 1\}^4 \textit{ for } i \in \mathbb{R},$$

where $\bar{p_i}$ *satisfies one of the following cases:*

$$\begin{cases} \frac{d\sum u}{dp_{|p=p^*}} = 0, & \textit{for } p^* \in \{\bar{p}_i\}^4, \\ \frac{d\sum u}{dp_{j_{|p_j=p_j^*}}} = 0, & \textit{for } p^* \in \{0, \bar{p}_i, 1\}^4 \textit{ where } p_j^* \in \{\bar{p}_i\} \textit{ for } j \in [1, 4]. \end{cases} \tag{16}$$

Equations (16) corresponds to the partial derivatives of the utility. For the partial derivatives to be zero, the numerator must fall to zero. The numerator of the partial derivatives is given by,

$$(\sum_{i=1}^{N} Q_N^{(i)'} \prod_{\substack{j=1\\ j\neq i}}^{N} Q_D^{(i)} + \sum_{i=1}^{N} Q_D^{(i)'} \sum_{\substack{j=1\\ j\neq i}}^{N} Q_N^{(i)} \prod_{\substack{j=1\\ j\neq\{i,j\}}}^{N} Q_D^{(i)}) \times \prod_{i=1}^{N} Q_D^{(i)} - (\sum_{i=1}^{N} Q_D^{(i)'} \prod_{\substack{j=1\\ j\neq i}}^{N} Q_D^{(i)}) \times (\sum_{i=1}^{N} Q_N^{(i)} \prod_{\substack{j=1\\ j\neq i}}^{N} Q_D^{(i)}) \tag{17}$$

and the denominator can not nullified, thus:

$$\prod_{i=1}^{N} Q_D^{(i)} \neq 0 \tag{18}$$

where,

$$\begin{aligned} Q_N^{(i)} &= \frac{1}{2} p Q^{(i)} p^T + c^{(i)} p + a^{(i)}, \\ Q_N^{(i)'} &= p Q^{(i)} + c^{(i)}, \\ Q_D^{(i)} &= \frac{1}{2} p \bar{Q}^{(i)} p^T + \bar{c}^{(i)} p + \bar{a}^{(i)}, \\ Q_D^{(i)'} &= p \bar{Q}^{(i)} + \bar{c}^{(i)}. \end{aligned}$$

Proof. The best response of a memory one strategy against a group of memory one strategies can captured by a candidate set of behaviours. This candidate set is constructed by considering behaviours where any or all of p_1, p_2, p_3, p_4 are $\in \{0, 1\}$ and the rest or all of p_1, p_2, p_3, p_4 are given by roots of the partial derivatives.

Note that for $p_i \in \{0, 1\}$ we consider the roots of the partial derivatives for $p_j \neq p_i$ for $i, j \in [1, 4]$.

The derivatives, $\frac{d\sum u}{dp}$, are given by,

$$\begin{aligned} &\frac{d}{dp} \sum_{i=1}^{N} {u_q}^{(i)}(p) = \\ &= \frac{(\sum\limits_{i=1}^{N} Q_N^{(i)'} \prod_{\substack{j=1\\ j\neq i}}^{N} Q_D^{(i)} + \sum\limits_{i=1}^{N} Q_D^{(i)'} \sum_{\substack{j=1\\ j\neq i}}^{N} Q_N^{(i)} \prod^{N}{}_{\substack{j=1\\ j\neq\{i,j\}}} Q_D^{(i)}) \times \prod\limits_{i=1}^{N} Q_D^{(i)} - (\sum\limits_{i=1}^{N} Q_D^{(i)'} \prod_{\substack{j=1\\ j\neq i}}^{N} Q_D^{(i)}) \times (\sum\limits_{i=1}^{N} Q_N^{(i)} \prod_{\substack{j=1\\ j\neq i}}^{N} Q_D^{(i)})}{(\prod\limits_{i=1}^{N} Q_D^{(i)})^2} \end{aligned} \tag{19}$$

For equation 19 to be zero, the numerator must fall to zero and the denominator can not nullified.

One the candidate set is constructed each point is evaluated using equation (11). The point with the maximum utility is selected. □

A special case of Lemma 4 is for $N = 1$, thus when a strategy plays against a single opponent. In this case the formulation of Theorem 1 is used and the best response is captured by Lemma 5.

Lemma 5. *The optimal behaviour of a memory one strategy player* $p^* \in \mathbb{R}^4_{[0,1]}$ *against a given opponent* $q \in \mathbb{R}^4_{[0,1]}$ *is given by:*

$$p^* = \text{argmax}(u_q(p)),\ p \in S_q,$$

where the set S_q *is defined as*

$$S_q = \{0, \bar{p}_i, 1\}^4 \textit{ for } i \in \mathbb{R},$$

where any $\bar{p}$ *satisfy condition (16). Note that now the numerators of the partial derivatives, (17), are given by*

$$(pQ + c)(\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a}) - (p\bar{Q} + \bar{c})(\frac{1}{2}pQp^T + cp + a) \tag{20}$$

and (18) is re-written as:

$$\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a} \neq 0 \tag{21}$$

Proof. The best response of a memory one strategy against another memory one strategy can captured by a candidate set of behaviours. This candidate set is constructed by considering behaviours where any or all of p_1, p_2, p_3, p_4 are $\in \{0, 1\}$ and the rest or all of p_1, p_2, p_3, p_4 are given by roots of the partial derivatives.

Note that for $p_i \in \{0, 1\}$ we consider the roots of the partial derivatives for $p_j \neq p_i$ for $i, j \in [1, 4]$.

The derivatives, $\frac{du}{dp}$, are given by,

$$\frac{du_q(p)}{dp} = \frac{(pQ + c)(\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a}) - (p\bar{Q} + \bar{c})(\frac{1}{2}pQp^T + cp + a)}{(\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a})^2} \tag{22}$$

For equation 19 to be zero, the numerator must fall to zero and the denominator can not nullified.

One the candidate set is constructed each point is evaluated using equation (11). The point with the maximum utility is selected. □

Equations (17) and (20) are systems of maximum 4 polynomials and the degree of the polynomials is gradually increasing every time an extra opponent is taken into account. Solving system of polynomials corresponds to the calculation of a resultant and for large systems these quickly become intractable. Because of that no further analytical consideration is given to problems described here.

Lemma 4 and Theorem 2 will now be used to give a list of particular best response results.

3.1 Reactive Strategies

The first constrained case considered here is that of the reactive strategies. Reactive strategies are a set of memory one strategies where they only take into account the opponent's previous moves. As described in Section 1 Tit for Tat is a reactive strategy. The optimisation problem of (12) now has an extra constraint and is re written as,

$$\begin{aligned} \max_p : & \sum_{i=1}^{N} u_q(p) \\ \text{such that} : & \; p_1 = p_3 \text{ and } p_2 = p_4 \\ & \; p_1, p_2 \in \mathbb{R}_{[0,1]}. \end{aligned} \tag{23}$$

Reactive strategies allow us to study u_p as a function of two variables p_1, p_2 and the best reactive response against a group of opponents is captured by Lemma 6.

Lemma 6. *The optimal behaviour of a reactive player $p^* \in \mathbb{R}^2_{[0,1]}$ against a set of N opponents $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ for $q^{(i)} \in \mathbb{R}^4_{[0,1]}$ is given by:*

$$p^* = \text{argmax}(\sum_{i=1}^{N} u_q(p)), \; p \in S_q,$$

where the set S_q is defined as

$$S_q = \{0, \bar{p_i}, 1\}^2 \textit{ for } i \in \mathbb{R},$$

where $\bar{p_i}$ satisfies one of the following cases:

$$\begin{cases} \frac{\sum u}{dp_{|p=p^*}} = 0, & \textit{for } p^* \in \{\bar{p_i}\}^4, \\ \frac{\sum u}{{dp_1}_{|p_1=p_1^*}} = 0, & \textit{for } p_1^* \in \{\bar{p_i}\} \textit{ while } p_2^* \in \{0,1\}, \\ \frac{\sum u}{{dp_2}_{|p_2=p_2^*}} = 0, & \textit{for } p_2^* \in \{\bar{p_i}\} \textit{ while } p_1^* \in \{0,1\}. \end{cases} \tag{24}$$

For cases (24) to be true the numerator of the partial derivatives, given by (17), must equal to zero and condition (18) must hold.

Similarly to the memory one approach a special case for $N = 1$, against a single memory one strategy, is applied to Lemma 6. The best response against a single opponent is captured by Lemma 7.

Lemma 7. *The optimal behaviour of a reactive player $p^* \in \mathbb{R}^2_{[0,1]}$ against a given opponent $q \in \mathbb{R}^4_{[0,1]}$ is given*

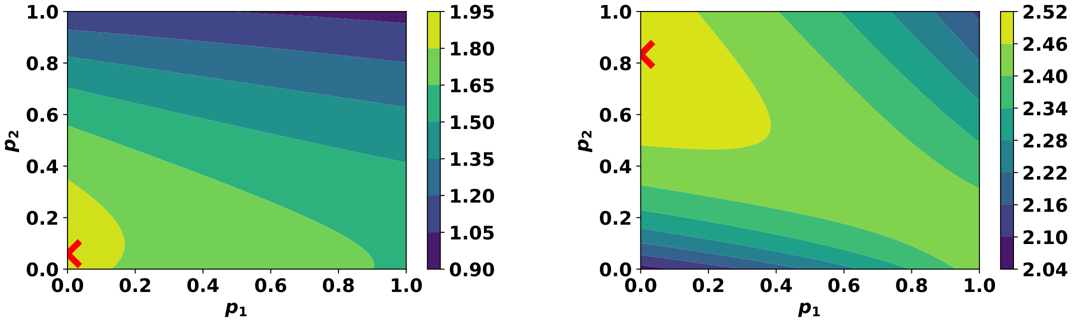


Figure 4: Numerical experiments for Algorithm 1 for $N = 1$.

by:

$$p^* = \text{argmax}(u_q(p)),\ p \in S_q,$$

where the set S_q is defined as

$$S_q = \{0, \bar{p}_i, 1\}^2 \textit{ for } i \in \mathbb{R},$$

where $\bar{p}_i$ satisfies any of (24). Partial derivatives are now given by (20) and (21) must be true.

Note that now equation (17) and (20) correspond to systems of maximum 2 polynomials over 2 variables. Each polynomial is equivalent to a partial derivative over p_1 and p_2. Several methods exist that allow for the 2 polynomial systems to be solved analytically. In this work the results theory is used to extract the roots from the partial derivatives.

Note that for pairwise interactions the maximum degree of the polynomials is equal to $2N$, however the degree increases as opponents are introduced.

The resultant is a symmetric function of the roots of the polynomials of a system and it can be expressed as a polynomial in the coefficients of the polynomials. The resultant will equal zero if and only if the system has at least one common root. Thus, the resultant becomes very useful in identifying whether common roots exist.

In this work the Sylvester's resultant [2] denoted as (R_S) is considered. The Sylvester's resultant is used to solve system of a single variable. However, for a system of two variables we solve over one variable and the second is kept as a coefficient. Thus we can find the roots of the equations and that is why the resultant is often refereed to as the eliminator.

In Appendix Algorithm 1 describes the process and based on the Algorithm several examples are performed. Illustrated by Figure 4 are the results of these examples. The results suggest that the best response behaviour is captured by our algorithm.

3.2 Purely random

The next constrained problem to be explored is that of the purely random strategies. Purely random strategies are a set of memory one strategies where the transition probabilities of each state are the same. The optimisation problem of (12) now has an extra constraint and is re written as,

$$\begin{aligned} \max_{p} : \frac{1}{N} \sum_{i=1}^{N} {u_q}^{(i)}(p) \\ \text{such that} : p_1 = p_2 = p_3 = p_4 = p \\ p \in \mathbb{R}_{[0,1]} \end{aligned} \tag{25}$$

and the exact optimal behaviour of purely random strategies is described in Lemma 8.

Lemma 8. *The optimal behaviour of a* ***purely random*** *player* $p^* \in \mathbb{R}_{[0,1]}$ *in an* $N-$*memory one player tournament,* $\{q_{(1)}, q_{(2)} \dots, q_{(N)}\}$ *for* $q_{(i)} \in \mathbb{R}_{[0,1]}^4$ *is given by:*

$$p^* = \text{argmax}(\sum_{i=1}^{N} {u_q}^{(i)}(p)), \ p \in S_{q(i)},$$

where the set S_q *is defined as:*

$$S_q = \{0, \lambda_i, 1\}, \ \textit{for } i \in [1, 2N]$$

where λ_i *are the eigenvalues of the companion matrix corresponding to the numerator of*

$$\frac{d}{dp} \sum_{i=1}^{N} {u_q}^{(i)}(p)$$

for which

$$\frac{d}{dp} \sum_{i=1}^{N} {u_q}^{(i)}(\lambda_j) = 0, \ \textit{for } j \in [1, 2N].$$

Proof. The best behaviour of a purely random strategy against a set of opponents is captured by a set of potential best behaviours. For constructing this a set a similar as the ones described in previous sections is used.

It is know that p^* will either be $\in \{0, 1\}$ or p^* will because given by the roots of $\frac{d}{dp} \sum\limits_{i=1}^{N} {u_q}^{(i)}(p)$. The roots of $\frac{d}{dp} \sum\limits_{i=1}^{N} {u_q}^{(i)}(p)$ are the roots only of the numerator, as the denominator can not be nullified.

Studying equation (6) as a function of a single variable p it can be verified that the degree of the numerator is equal to$2N$. Thus, the size of roots of the numerator is equal to $2N$.

The roots on the polynomial in this work will be calculated using a companion matrix method [10]. This method allows the roots of the polynomial to be computed by calculating the eigenvalues of the corresponding

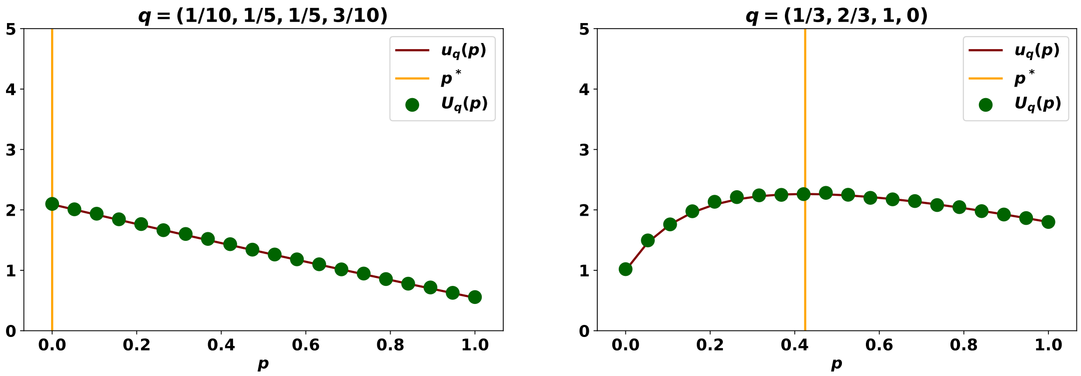


Figure 5: Numerical experiments for Algorithm 2 for $N = 1$.

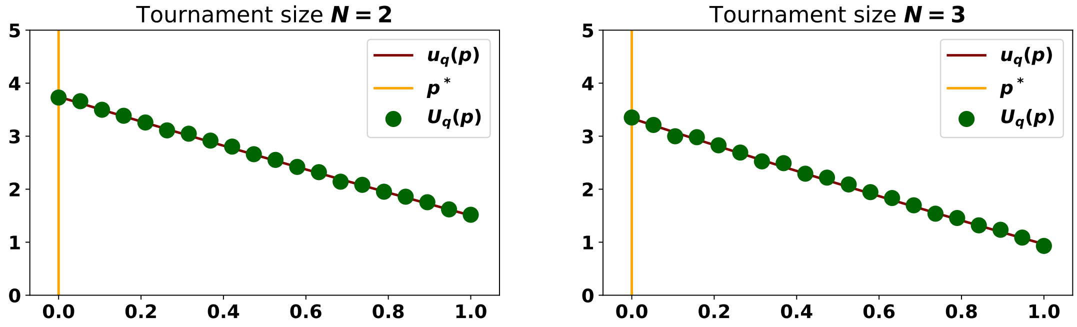


Figure 6: Numerical experiments for Algorithm 2 for $N > 1$.

companion matrix.

The eigenvalues and the bounds compose the candidate set of solutions. The best response of a purely random player corresponds to the point that maximises (6). □

Lemma 8 can be further expanded to the case where $N = 1$. Algorithm 2 describes the process of best purely random responses and it is used for a number of empirical trials. The results of these trials are illustrated in Figures 5 and 6. Figure 5 demonstrates Lemma 8 for $N = 1$ and Figure 6 cases were $N > 1$. It is evident that the optimal behaviour has been captured by our search algorithm.

Furthermore, for the case of the purely random players two more theoretical results are discussed. These are the cases where the opponent has manage to make a random player indifferent and the case where a purely random player is better of playing a pure strategy.

There is importance in both results. Initially, being indifferent refers to our actions no having any effects on the match. Thus there is not optimal behaviour for player p.

Secondly, by a pure strategy we are referring to the $p = 0$ and $p = 1$. In this case it is know that p^* is $\in 0, 1$ without testing the roots of the derivative. The optimisation problem crumbles to a binary problem.

The results are given equivalently by Lemmas 9 and 10 and they are respective to the actions of the opponent. Figure 7 illustrates examples for both lemmas.

Lemma 9. *A given memory one player,* (q_1, q_2, q_3, q_4)*, makes a* ***purely random*** *player,* (p, p, p, p)*, indifferent if and only if,* $-q_1+q_2+2q_3-2q_4 = 0$ *and* $(q_2-q_4-1)(q_1-2q_2-5q_3+7q_4+1)-(q_2-5q_4-1)(q_1-q_2-q_3+q_4) = 0$*.*

Lemma 10. *Against a memory one player,* (q_1, q_2, q_3, q_4)*, a* ***purely random*** *player would always play a pure strategy if and only if* $(q_1q_4 - q_2q_3 + q_3 - q_4)(4q_1 - 3q_2 - 4q_3 + 3q_4 - 1) = 0$*.*

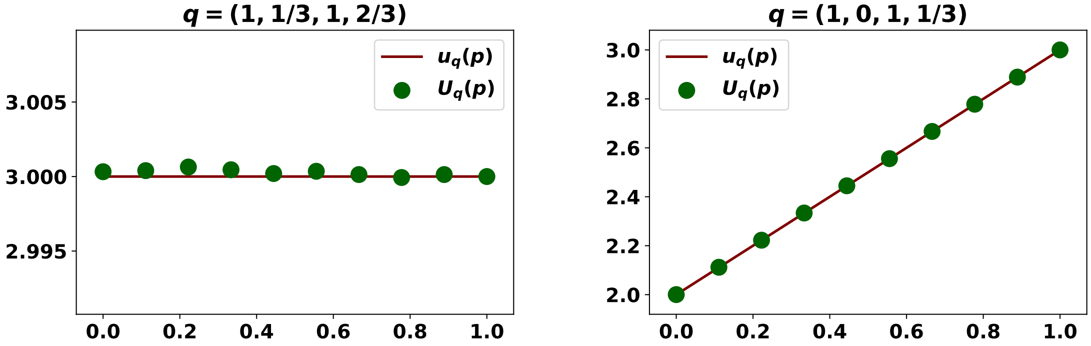


Figure 7: Proof of concept for Lemmas 9, 10.

4 Stability of defection

In this section the stability of defection is explored. Defection is known to be the dominant action in the PD and it can be proven to be the dominant strategy for the IPD for given environments. Even so, several works have proven that cooperation emerges in the IPD many studies around the game focus on the emergence of cooperation. In this manuscript we try to provide a condition for when defection is the best response in the IPD, thus when it is known that cooperation can not occur.

Initially, let's consider equation (22) and let equation (22) for $p = (0, 0, 0, 0)$,

$$\frac{du}{dp_{|p=(0,0,0,0)}} = \frac{c\bar{a} - \bar{c}a}{\bar{a}^2}. \tag{26}$$

The numerator $\bar{c}a - c\bar{a}$ is given by,

$$\left[\begin{matrix}0\\0\\q_{4} \left(4 q_{1} q_{2} - 3 q_{2}^{2} - 4 q_{2} q_{3} + 3 q_{2} q_{4} + 4 q_{3} - 5 q_{4} - 1\right)\\- \left(q_{2} - 1\right) \left(4 q_{1} q_{4} - 3 q_{2} q_{4} + q_{2} - 4 q_{3} q_{4} + 4 q_{3} + 3 q_{4}^{2} - 6 q_{4} - 1\right)\end{matrix}\right]$$

and the denominator $\bar{a}^2 = (-q_2 + q_4 + 1)^2$, which is always positive. In order for defection to be the best response the derivative must have a negative sign at the point $p = (0, 0, 0, 0)$. That means that the utility is only decreasing after $p = (0, 0, 0, 0)$.

Because $\bar{a}^2$ is always positive the sign of the derivative is given by $\bar{c}a - c\bar{a}$. More specifically from equations,

$$q_4\left(4q_1q_2 - 3q_2^2 - 4q_2q_3 + 3q_2q_4 + 4q_3 - 5q_4 - 1\right) \tag{27}$$

$$-\left(q_2 - 1\right)\left(4q_1q_4 - 3q_2q_4 + q_2 - 4q_3q_4 + 4q_3 + 3q_4^2 - 6q_4 - 1\right) \tag{28}$$

Both signs of the partial derivatives must be negative in order for the overall function to be decreasing, thus defection being the best response. The signs of equations (27) and (28) vary. There are cases that they have the same sign and cases that they do not, this is shown by numerical example summarized in Table 2.

					equation(27)	equation(28)
1	$q_1 = \frac{3}{10}$,	$q_2 = \frac{3}{20}$,	$q_3 = \frac{13}{20}$,	$q_4 = \frac{7}{100}$	+	+
2	$q_1 = \frac{11}{25}$,	$q_2 = \frac{3}{10}$,	$q_3 = \frac{9}{10}$,	$q_4 = \frac{1}{2}$	-	-
3	$q_1 = \frac{17}{20}$,	$q_2 = \frac{3}{4}$,	$q_3 = \frac{2}{5}$,	$q_4 = \frac{1}{4}$	-	+
4	$q_1 = \frac{13}{88}$,	$q_2 = \frac{21}{92}$,	$q_3 = \frac{21}{26}$,	$q_4 = \frac{20}{67}$	+	-

Table 2: Numerical examples of the derivative's sign.

For a tournament setting we substitute $p = (0, 0, 0, 0)$ in equation (19) which laid out:

$$\sum_{i=1}^{N}(c^{(i)T}\bar{a}^{(i)} - \bar{c}^{(i)T}a^{(i)}) \prod_{\substack{j\,=\,1 \\ j\,\neq\,i}}^{N} (\bar{a}^{(i)})^2 \tag{29}$$

The product term $\prod_{\substack{j\,=\,1 \\ j\,\neq\,i}}^{N} (\bar{a}^{(i)})^2$ is known to always be positive. However the sign of the sum term $\sum_{i=1}^{N}(c^{(i)T}\bar{a}^{(i)} - \bar{c}^{(i)T}a^{(i)})$ can vary based on the transition probabilities of the opponents, as discussed above. A condition that must hold in order for defection to be stable in a tournament is that the sum term must be negative. The results are exhibited in Lemma 11.

Lemma 11. *In a tournament of* N *players where* $q^{(i)} = (q_1^{(i)}, q_2^{(i)}, q_3^{(i)}, q_4^{(i)})$ *defection is known to be a best response if the transition probabilities of the opponents satisfy the condition:*

$$\sum_{i=1}^{N}(c^{(i)T}\bar{a}^{(i)} - \bar{c}^{(i)T}a^{(i)}) <= 0. \tag{30}$$

Moreover lets us consider a constrained version of the problem once again. Lets us assume that in an pairwise interaction the opponent is a reactive player $q = (q_1, q_2, q_1, q_2)$. By substituting $q_3 = q_1$ and $q_4 = q_2$ equations (27) and (28) are now re written as follow,

$$\begin{bmatrix} -q_2\left(4q_1 - 5q_2 - 1\right) \\ \left(q_2 - 1\right)\left(4q_1 - 5q_2 - 1\right) \end{bmatrix}$$

The sign of both equations is now based on the same term,$(4q_1 - 5q_2 - 1)$, which is a term that can have both negative and positive values. This is shown by Figure 8. Following this the following result is retrieved,

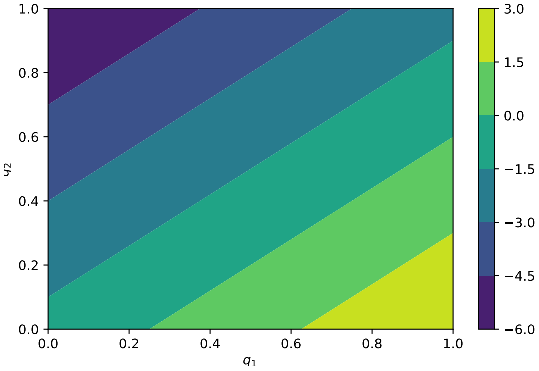


Figure 8: Sign of $(4q_1 - 5q_2 - 1)$.

Lemma 12. *Defection is the best responses of a memory one player p against a reactive player q if the transition probabilities of the opponent satisfy the condition:*

$$4q_1 - 5q_2 - 1 > 0 \tag{31}$$

5 Optimisation against memory one strategies

The main aim of this work is to capture best response memory one strategies. Though an analytical approach can not be applied further than discussed in Section 3 an empirical approach is considered here. More specially, the optimisation problem of (12) is maximized using bayesian optimisation. Bayesian optimisation is a global optimisation algorithm, introduced in [19], which has proven to outperform many other popular algorithms [14].

As many other algorithms Bayesian optimisation is used to find the maximum of a function f on some bounded set. Bayesian constructs a probabilistic model for f and then exploits this model to make decisions about where in the bounded set to next evaluate the function. It relays on the previous of f and not simply rely on local gradient and Hessian approximations. This allows the algorithm to optimise a non concave function with relatively few evaluations, at the cost of performing more computation to determine the next point to try [23].

The open source package [13] offers an implementation of bayesian optimisation and is used in this paper to compute a large number of best memory one responses against sets of random memory one strategies. The implementation of Bayesian in [13] allows us to perform the algorithm for a different combination of parameters. The parameters explored here are:

- Number of calls, maximum number of calls to the objective function.
- Number of random starts, number of evaluations of the objective function with random points before approximating it.

The different parameters' combinations and their respective values are laid out in Table 3.

	number of calls	number of random starts
1	20	10
2	30	20
3	40	20
4	45	20
5	50	20

Table 3: Bayesian optimisation sets of parameters' values.

Note that another global optimization algorithm called differential evolution [25] was also reviewed for the purpose of this paper. Bayesian optimization was chosen over differential evolution due to a high computational cost.

The combination that was chosen to carry out the empirical trials is that of 50 number of calls and 20 number of random starts. This set of parameters was determined to be the most efficient using best reactive responses as an experimental case.

A total of 9900 different memory one opponents were randomly generated and bayesian optimisation was used to find the optimal reactive strategy. The results were compared to that of Lemma 7 which we know exhibits the best reactive response. The particular parameters' set was the set with the smallest difference between the two captured best behaviours. The results of this comparison are presented by Table 4.

Difference	Labels
0.026991	calls: 20, random starts: 10
0.018916	calls: 30, random starts: 20
0.007298	calls: 40, random starts: 20
0.005552	calls: 45, random starts: 20
0.005114	calls: 50, random starts: 20

Table 4: Difference of $u_q(p)$ for $p \in \mathbb{R}^2_{[0,1]}$. The difference was calculated as exact $u_q(p^*)$ minus Bayesian $u_q(\tilde{p}^*)$.

Thus it can be confirmed that bayesian optimisation manages to capture the optimal behaviour of reactive strategies. The very same set of parameters was used to optimise memory one strategies against single opponents and against sets of $N = 2$ opponents. $N = 2$ was chosen because is the smallest N for which there is a multi opponent interaction. For each $N = 1$ and $N = 2$ opponents a total of 1022 best responses of memory one strategies have been captured. This data has been archived in.

6 Limitation of memory

The third and final part of this paper focuses on proving that short memory strategies have limitations. Though it has been proven [22] that there exists a set of memory one strategies that can outperform any opponent, this was done only for the case of $N = 1$. In this section we introduce several empirical results that show that more complex strategies can indeed perform better in cases of $N = 2$. This is achieved by comparing the performance of an optimised memory one strategy to that of a trained long memory one.

The long memory strategies are trained using bayesian algorithm as the one described in Section 5. The trained strategy used is a strategy called Gambler, introduced and discussed in [12], and the objective function of a Gambler is the average performance in a tournament of 200 turns and 50 repetitions.

6.1 Gambler

Several means of representing strategies have been used over the years for IPD strategies. In [12] several of those 'archetypes' are presented and used to train different successful strategies. One of the archetypes firstly introduced in that paper is Gambler. Gambler is based on a lookup table and encodes a probability of cooperating based on the opponent's first n_1 moves, the opponent's last m_1 moves, and the players last m_2 moves.

Several variants of Gambler have been trained for this work, a summary is given by Table 5. In essence Gambler can represent any generic strategy and that is why it has been chosen.

	n_1	m_1	m_2
1	1	1	2
2	2	2	0
3	2	2	1
4	2	2	2
5	4	4	4

Table 5: Variants of Gambler used.

6.2 Empirical Results

The performance of the two strategies are compared for cases of $N = 1$ and $N = 2$. The following steps are taken:

1. An N number of random opponents are generated.
2. Using (12) p^* for that given environment is captured.
3. A Gambler type (each variant of Table 5) is trained for the same environment.
4. Both utilities of the optimised and the trained strategy are reordered.

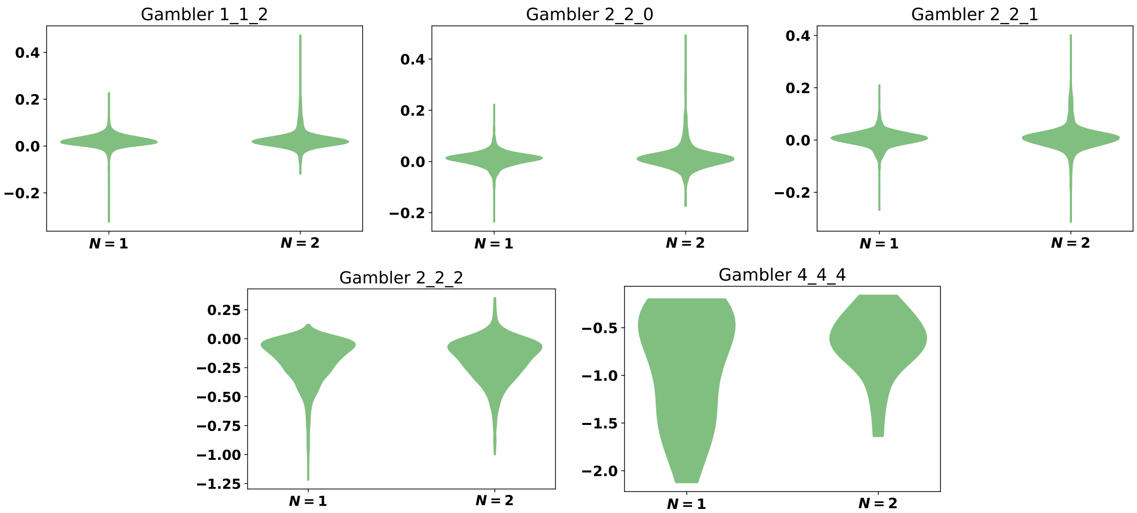


Figure 9: Difference between $\frac{1}{N}\sum\limits_{i=1}^{N} {u_q}^{(i)}(p^*)$ and $\frac{1}{N}\sum\limits_{i=1}^{N} {U_q}^{(i)}G$ for $N = 1$ and $N = 2$.

A large data set containing the opponents as well as the optimised/trained behaviours can be found in. The number of experimental cases for each Gambler are displayed in Table 6. Note that a number of 1022 trials corresponds to 1022 trials for $N = 1$ and 1022 trials for $N = 2$.

	Gamblers	Number of trials
0	Gambler 1_1_2	1022
1	Gambler 2_2_0	1043
2	Gambler 2_2_1	1074
3	Gambler 2_2_2	821
4	Gambler 4_4_4	22

Table 6: Number of trials, for $N = 1$ and $N = 2$, for each Gambler instance.

The results are explored by studying the difference between $\frac{1}{N}\sum\limits_{i=1}^{N} {u_q}^{(i)}(p^*)$ and $\frac{1}{N}\sum\limits_{i=1}^{N} {U_q}^{(i)}G$, where UG represents the utility of a Gambler. The results are illustrated in Figure 9. It is exhibited that for each Gambler instance memory one strategies perform better, which was anticipated.

However, the results indicate that for $N = 2$ a significant difference in the perform can occur. For example, cases of Gambler $n_1 = 1, m_1 = 1, m_2 = 2$, $n_1 = 2, m_1 = 2, m_2 = 0$ and $n_1 = 2, m_1 = 2, m_2 = 1$ there is a difference in performance of nearly 0.5.

For the rest of the Gambler's types they perform worst of the same. This could also be a result of the number of calls, which would need to be higher so such complex strategies.

7 Discussion

In this framework a specific set of strategies for the well known game the IPD have been discussed. This set of strategies are the memory one strategies, which are a set of strategies that utilize a single slot of memory to define their next action.

The analytical approach for retrieving the payoffs of memory one strategies against memory one strategies was manipulated here. Though several works have done similar works, this manuscript is the first to show that the utility has a compact form and exploit it as an objective function to a non convex optimisation problem.

In essence, we used analytical approaches to establish best responses in the IPD. Initially, for reactive and purely random strategies it was proven, using algebraic approaches such as companion matrices and resultants, that best responses can be captured analytically.

Secondly, the stability of defection was investigate. It was proven that environments for which cooperation will never emerge can be recognised immediately by the transitions of the opponents.

Moreover, a large date set of bests memory one responses was generated for $N = 1$ and $N = 2$. The limitations of memory were tried to be shown by comparing the performance of best memory one strategies to that of more complex strategies. Though there are indications that complex strategies indeed perform better, the significant of the difference is in question. More experimental trials and exploration is hope to be carried out.

A Appendix Tables

The memory one strategies used in the computer tournament described in [24] are given by Table 7.

	Name	Memory one representation	Explanation
1	Cooperator	$(1, 1, 1, 1)$	Always chooses C.
2	Defector	$(0, 0, 0, 0)$	Always chooses D.
3	Random	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$	Randomly chooses between C and D with a probability of 0.5.
4	Tit for Tat	$(1, 0, 1, 0)$	Start with a C and then mimics the opponent's last move.
5	Grudger	$(1, 0, 0, 0)$	Starts by cooperating however will defect if at any point the opponent has defected.
6	Generous Tit for Tat	$(1, \frac{1}{3}, 1, \frac{1}{3})$	A more generous version of Tit for Tat.
7	Win Stay Lose Shift	$(1, 0, 0, 1)$	Starts with a C and then repeats it's previous move only if it was awarded with a payoff of R or T.
8	ZDGTFT2	$(1, \frac{1}{8}, 1, \frac{1}{4})$	A generous zero determinant strategy introduced in [24]
9	ZDExtort2	$(\frac{8}{9}, \frac{1}{2}, \frac{1}{3}, 0)$	An extortionate zero determinant strategy introduced in [24]
10	Hard Joss	$(\frac{9}{10}, 0, \frac{9}{10}, 0)$	Cooperates with probability $\frac{9}{10}$ when the opponent cooperates, otherwise emulates Tit for Tat.

Table 7: The memory one strategies from [24].

Algorithm 1 Best response algorithm for reactive strategies

```
procedure REACTIVE SEARCH
    $N \leftarrow$ number of opponets
    $S_q \leftarrow \{0,1\}^2$
    $u' \leftarrow \frac{d\sum\limits_{i=1}^{N} u}{d\bar{p}}$
    $\frac{u_N}{u_D} \leftarrow u'$
    $S(u_N, p_2) \leftarrow$ Sylvester's matrix for$p_2$.Coefficients are polynomials of $p_1$
    $R_S(S) \leftarrow det(S)$
    roots$_{p_1} \leftarrow p_1$ for $det(M)_{p_1} = 0$
loop root in roots$_{p_1}$:
    system$(p_2) \leftarrow u_N(root)$
    root$_{p_2} \cup p_2$ for system$(p_2) = 0$
    if $u_D((\text{root}, \text{root}_{p_2})) \neq 0$ then
        $S_q \cup \{(\text{root}, \text{root}_{p_2})\}$
    goto loop.
    close;
    $p^* \leftarrow \text{argmax}(\sum\limits_{i=1}^{N} u_{q^{(i)}}(p)), p \in S_q$.
```

Algorithm 2 Best response algorithm for purely random strategies

```
procedure PURELY RANDOM SEARCH
    $N \leftarrow$ number of opponets
    $S_q \leftarrow \{0,1\}$
    $u' \leftarrow \frac{d\sum\limits_{i=1}^{N} u}{d\bar{p}}$
    $\frac{u_N}{u_D} \leftarrow u'$
    $C(u_N) \leftarrow$ companion matrix of $u_N$
loop $i = 1$ to $2N$:
    $\lambda_i \leftarrow$ eigenvalue of $C(u_N)$
    if $u_D(\lambda_i) \neq 0$ then
        $S_q \cup \lambda_i$.
    goto loop.
    close;
    $p^* \leftarrow \text{argmax}(\sum\limits_{i=1}^{N} u_{q^{(i)}}(p)), p \in S_q$.
```

B Appendix Algorithms

References

[1] The Axelrod project developers . Axelrod: ¡release title¿, April 2016.

[2] Alkiviadis G. Akritas. *Sylvester's form of the Resultant and the Matrix-Triangularization Subresultant PRS Method*, pages 5–11. Springer New York, New York, NY, 1991.

[3] Howard Anton and Chris Rorres. *Elementary Linear Algebra: Applications Version.* Wiley, eleventh edition, 2014.

[4] R Axelrod and WD Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[5] Robert Axelrod. Effective choice in the prisoner's dilemma. *The Journal of Conflict Resolution*, 24(1):3–25, 1980.

[6] Robert Axelrod. More effective choice in the prisoner's dilemma. *The Journal of Conflict Resolution*, 24(3):379–403, 1980.

[7] Amir Beck and Marc Teboulle. A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid. *Mathematical Programming*, 118(1):13–35, 2009.

[8] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods.* Academic press, 2014.

[9] Hongyan Cai, Yanfei Wang, and Tao Yi. An approach for minimizing a quadratically constrained fractional quadratic problem with application to the communications over wireless channels. *Optimization Methods and Software*, 29(2):310–320, 2014.

[10] Alan Edelman and H Murakami. Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation*, 64(210):763–776, 1995.

[11] Giorgio Giorgi, Bienvenido Jiménez, and Vicente Novo. Approximate karush—kuhn—tucker condition in multiobjective optimization. *J. Optim. Theory Appl.*, 171(1):70–89, October 2016.

[12] Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E. Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoners dilemma. *PLOS ONE*, 12(12):1–33, 12 2017.

[13] Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Z Vincius, cmmalone, Christopher Schrder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan, rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak, SoManyUsernamesTaken, Fred Callaway, Loc Estve, Lilian Besson, Mehdi Cherti, Karlson Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander Fabisch. scikit-optimize/scikit-optimize: v0.5.2, March 2018.

[14] Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[15] Jeremy Kepner and John Gilbert. *Graph algorithms in the language of linear algebra.* SIAM, 2011.

[16] Vincent Knight, Owen Campbell, Marc Harper, Karol Langner, James Campbell, Thomas Campbell, Alex Carney, Martin Chorley, Cameron Davidson-Pilon, Kristian Glass, Tomáš Ehrlich, Martin Jones, Georgios Koutsovoulos, Holly Tibble, Müller Jochen, Geraint Palmer, Paul Slavin, Timothy Standen, Luis Visintini, and Karl Molden. An open reproducible framework for the study of the iterated prisoner's dilemma. 1(1), 2016.

[17] Christopher Lee, Marc Harper, and Dashiell Fryer. The art of war: Beyond memory-one strategies in population games. *PLOS ONE*, 10(3):1–16, 03 2015.

[18] Frederick A Matsen and Martin A Nowak. Win–stay, lose–shift in language learning from peers. *Proceedings of the National Academy of Sciences*, 101(52):18053–18057, 2004.

[19] J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[20] Martin Nowak and Karl Sigmund. Game-dynamical aspects of the prisoner's dilemma. *Applied Mathematics and Computation*, 30(3):191–213, 1989.

[21] Martin Nowak and Karl Sigmund. The evolution of stochastic strategies in the prisoner's dilemma. *Acta Applicandae Mathematicae*, 20(3):247–265, 1990.

[22] William H. Press and Freeman J. Dyson. Iterated prisoners dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.

[23] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[24] Alexander J. Stewart and Joshua B. Plotkin. Extortion and cooperation in the prisoners dilemma. *Proceedings of the National Academy of Sciences*, 109(26):10134–10135, 2012.

[25] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.