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

Nikoleta E. Glynatsi Vincent Knight

Abstract

Memory-one strategies are a set of Iterated Prisoner's Dilemma strategies that have been praised for their mathematical tractability and performance against single opponents. This manuscript investigates *best response* memory-one strategies as a multidimensional optimisation problem. Though extortionate memory-one strategies have gained much attention, we demonstrate that best response memory-one strategies do not behave in an extortionate way, and moreover, for memory one strategies to be evolutionary robust they need to be able to behave in a forgiving way. We also provide evidence that memory-one strategies suffer from their limited memory in multi agent interactions and can be out performed by longer memory strategies.

1 Introduction

The Prisoner's Dilemma (PD) is a two player game used in understanding the evolution of co-operative behaviour, formally introduced in [10]. Each player has two options, to cooperate (C) or to defect (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), 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. The most common values used in the literature are $(R, P, S, T) = (3, 1, 0, 5)$ [6].

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

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

The PD is a one shot game, however it is commonly studied in a manner where the history of the interactions matters. The repeated form of the game is called the Iterated Prisoner's Dilemma (IPD) and in the 1980s, following the work of [4, 5] it attracted the attention of the scientific community. In [4] and [5], the first well known computer tournaments of the IPD were performed. A total of 13 and 62 strategies were submitted respectively in the form of computer code. The contestants competed against each other, a copy of themselves and a random strategy, and the winner was then decided on the average score achieved (not the total number of wins). The contestants were given access to the entire history of a match, however, how many turns of history a strategy would incorporate, refereed to as the *memory size* of a strategy, was a result of the particular strategic decisions made by the author. The winning strategy of both tournaments was the strategy called Tit for Tat and it's success, in both tournaments, came as a surprise. Tit for Tat was a simple, forgiving strategy that opened each interaction by cooperation, but it had managed to defeat far more complicated opponents. Tit for Tat provided evidence that being nice can be advantageous and became the major paradigm for reciprocal altruism.

Another trait of Tit for Tat is that it considers only the previous move of the opponent. These type of strategies are called *reactive* [25] and are a subset of so called *memory-one* strategies, which incorporate both players' latests moves. Memory-one strategies have been studied thoroughly in the literature [26, 27], however, they have gained most of their attention when a certain subset of memory-one strategies was introduced in [28], the zero-determinants. In [29] it was stated that "Press and Dyson have fundamentally changed the viewpoint on the Prisoner's Dilemma".

Zero-determinants are a special case of memory-one and extortionate strategies. They chose their actions so that a linear relationship is forced between the players' score ensuring that they will always receive at least as much as their opponents. Zero-determinant strategies are indeed mathematically unique and are proven to be robust in pairwise interactions, however, their true effectiveness in tournaments and evolutionary dynamics has been questioned [2, 20, 22].

In a similar fashion to [28] the purpose of this work is to consider a given memory-one strategy; however, whilst [28] 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 given group of opponents. In particular, this work presents a compact method of identifying the best response memory-one strategy against a given set of opponents and evaluate whether it behaves extortionately, similar to zero-determinants. Further theoretical and empirical results of this work include:

1. The factors that make a best response memory-one strategy evolutionary robust.
2. A well designed framework that allows the comparison of an optimal memory one strategy, and a more complex strategy that has a larger memory and was obtained through contemporary reinforcement learning techniques [13].
3. An identification of conditions for which defection is known to be a best response; thus identifying environments where cooperation will not occur.

2 The utility

One specific advantage of memory-one strategies is their mathematical tractability. They can be represented completely as an element of $\mathbb{R}^4_{[0,1]}$. This originates from [25] where 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; CC, CD, DC, CC. Therefore, a memory-one strategy can be denoted by the probability vector of cooperating after each of these states; $p = (p_1, p_2, p_3, p_4) \in \mathbb{R}^4_{[0,1]}$.

In [25] it was shown that it is not necessary to simulate the play of a strategy p against a memory-one

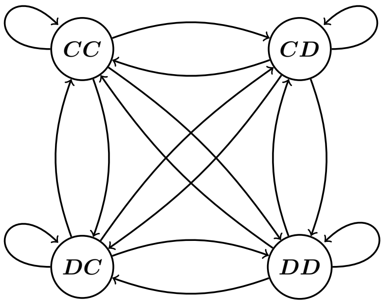


Figure 1: Markov Chain

opponent q. Rather this exact behaviour can be modeled as a stochastic process, and more specifically as a Markov chain (Figure 1) whose corresponding transition matrix M is given by (4).

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

The long run steady state probability vector v is the solution to $vM = v$. The stationary vector v combined with the payoff matrices of (1) give the expected payoffs for each player. More specifically, the utility for a memory-one strategy p against an opponent q, denoted as $u_q(p)$, is defined by,

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

To the authors knowledge there has been no previous work which explored the form of $u_q(p)$ any further. This manuscript proves that $u_q(p)$ is given by a ratio of two quadratic forms [18], as presented in Theorem 1.

Theorem 1. *The expected utility of a memory-one strategy $p \in \mathbb{R}^4_{[0,1]}$ against a memory-one opponent $q \in \mathbb{R}^4_{[0,1]}$, 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 $Q, \bar{Q} \in \mathbb{R}^{4\times 4}$ are square matrices whose diagonal elements are all equal to zero, and are defined by the transition probabilities of the opponent q_1, q_2, q_3, q_4 as follows:

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

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

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

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

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

and the constant terms $a, \bar{a}$ are defined as $a = -q_2 + 5q_4 + 1$ and $\bar{a} = -q_2 + q_4 + 1$.

The proof of Theorem 1 is given in Appendix A.1.

Numerical simulations have been carried out to validate the formulation of $u_q(p)$ as a quadratic ratio. The simulated utility, which is denoted as $U_q(p)$, has been calculated using [1] an open source research framework for the study of the IPD [1]. For smoothing the simulated results the simulated utility has been estimated in a tournament of 500 turns and 200 repetitions. Figure 2 shows that the formulation of Theorem 1 successfully captures the simulated behaviour.

The source code used in this manuscript has been written in a sustainable manner. It is open source (`https://github.com/Nikoleta-v3/Memory-size-in-the-prisoners-dilemma`) and tested which ensures the validity of the results. It has also been archived and can be found at.

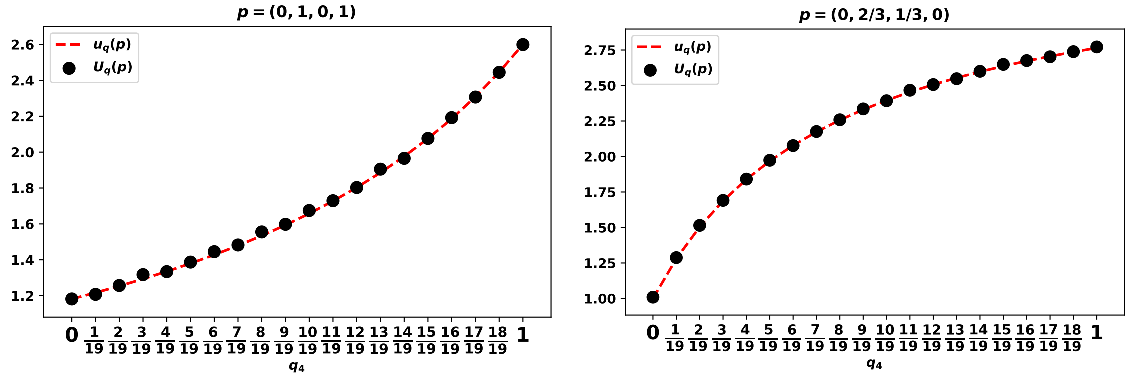


Figure 2: Simulated and analytically calculated utility for $p = (0, 1, 0, 1)$ and $p = (0, \frac{2}{3}, \frac{1}{3}, 0)$ against $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, q_4)$ for $q_4 \in \{0, \frac{1}{19}, \frac{2}{19}, \dots, \frac{18}{19}, 1\}$.

Theorem 1 can be extended to consider multiple opponents. The IPD is commonly studied in tournaments and/or Moran Processes where a strategy interacts with a number of opponents. The payoff of a player in

[1] The project is described in [19].

such interactions is given by the average payoff the player received against each opponent. 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:*

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

The proof of Theorem 2 is a straightforward algebraic manipulation.

Similar to the previous result, the formulation of Theorem 2 is validated using numerical simulations where the 10 memory-one strategies described in [29] have been used as the opponents. Figure 3 shows that the simulated behaviour has been captured successfully.

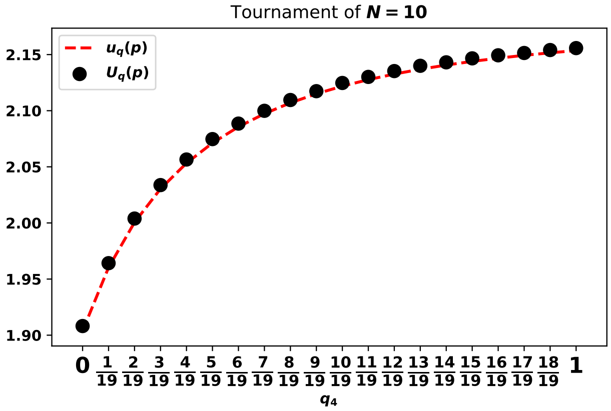


Figure 3: The utilities of memory-one strategies $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, p_4)$ for $p_4 \in \{0, \frac{1}{19}, \frac{2}{19}, \dots, \frac{18}{19}, 1\}$ against the 10 memory-one strategies described in [29].

The list of strategies from [29] was also used to check whether the utility against a group of strategies could be captured by the utility against the mean opponent. Thus whether condition (12) holds. However condition (12) fails, as shown in Figure 4.

$$\frac{1}{N}\sum_{i=1}^{N} {u_q}^{(i)}(p) = u_{\frac{1}{N}\sum\limits_{i=1}^{N} q^{(i)}}(p), \tag{12}$$

In this section two theoretical results were presented. The formulation of Theorem 2 which allows for the utility of a memory-one strategy against any number of opponents to be estimated without simulating the interactions is the main result used in this manuscript. In Section 3 it is used to define best response memory-one strategies and explore the conditions under which defection dominates cooperation.

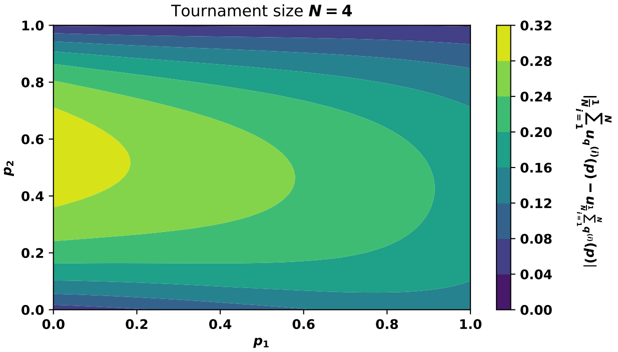


Figure 4: The difference between the average utility against the opponents from [29] and the utility against the average player of the strategies in [29]. A positive difference indicates that condition (12) does not hold.

3 Best responses to memory-one players

This section focused on best responses and more specifically *memory-one best response* strategies. A *best response* is the strategy which corresponds to the most favorable outcome [31], thus a memory-one best response corresponds to a strategy p^* for which (11) is maximised. This is considered as a multi dimensional optimisation problem given by:

$$\begin{aligned} \max_{p} : & \sum_{i=1}^{N} {u_q}^{(i)}(p) \\ \text{such that} : & \ p \in \mathbb{R}_{[0,1]} \end{aligned} \tag{13}$$

Optimising this particular ratio of quadratic forms is not trivial. 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. Some results are known for non concave ratios of quadratic forms [7, 9], however, in these works it's assumed that either both the numerator and the denominator of the fractional problem are concave or that the denominator is greater than zero. Both assumptions fail here as stated 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 numerator or the denominator of (6), are concave or greater than zero.*

Proof is given in Appendix A.2.

The non concavity of $u(p)$ indicates multiple local optimal points. The approach taken here is to introduce a compact way of constructing the candidate set of all local optimal points. Once the set is defined the point that maximises (11) corresponds to the best response strategy. The problem considered is bounded because $p \in \mathbb{R}^4_{[0,1]}$. Therefore, the candidate solutions will exist either at the boundaries of the feasible solution space, or within that space (the methods of Lagrange Multipliers [8] and Karush-Kuhn-Tucker conditions [11] are based on this). This approach allow us to define the best response memory-one strategy to a group of opponents in the following Lemma:

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:*

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

The set S_q is defined as all the possible combinations of:

$$S_q = \left\{ p \in \mathbb{R}^4 \left| \begin{array}{l} \bullet \quad p_j \in \{0,1\} \quad \textit{and} \quad \dfrac{d}{dp_k}\displaystyle\sum_{i=1}^{N} u_q^{(i)}(p) = 0 \quad \textit{forall} \quad j \in J \quad \& \quad k \in K \quad \textit{forall} \quad J, K \\ \qquad\qquad\qquad\qquad \textit{where} \quad J \cap K = \emptyset \quad \textit{and} \quad J \cup K = \{1,2,3,4\}. \\ \bullet \quad p \in \{0,1\}^4 \end{array} \right. \right\}. \tag{14}$$

The proof is given in the Appendix A.3.

Note that there is no immediate way to find the zeros of $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q(p)$;

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

$$= \sum_{i=1}^{N} \frac{\left(pQ^{(i)} + c^{(i)}\right)\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right) - \left(p\bar{Q}^{(i)} + \bar{c}^{(i)}\right)\left(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)}\right)}{\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right)^2} \tag{15}$$

For $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q(p)$ to equal zero then:

$$\sum_{i=1}^{N}\left(\left(pQ^{(i)} + c^{(i)}\right)\left(\frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)}\right) - \left(p\bar{Q}^{(i)} + \bar{c}^{(i)}\right)\left(\frac{1}{2}pQ^{(i)}p^T + c^{(i)}p + a^{(i)}\right)\right) = 0, \quad \textit{while} \tag{16}$$

$$\sum_{i=1}^{N} \frac{1}{2}p\bar{Q}^{(i)}p^T + \bar{c}^{(i)}p + \bar{a}^{(i)} \neq 0. \tag{17}$$

Finding best response memory-one strategies, more specifically constructing the subset S_q, can done analytically. The points for any or all of $p_i \in \{0,1\}$ for $i \in \{1,2,3,4\}$ are trivial, and finding the roots of the partial derivatives which are a set of polynomials of equations (16) is feasible using resultant theory [17]; however, for large systems building the resultant quickly becomes intractable. As a result, a numerical method taking advantage of the structure will be used for finding best response memory-one strategies. This will be described in Section 4. The rest of the section focuses on an immediate theoretical result from Lemma 4.

3.1 Stability of defection

An immediate result from Lemma 4 can be obtained by evaluating the sign of the derivative (15) at $p = (0,0,0,0)$. If at that point the derivative is negative, then a player maximises their utility by playing as a defector, and their utility would only decrease if they were to change their behaviour. Thus, defection is the best response.

Lemma 5. *In a tournament of N players $\{q^{(1)}, q^{(2)}, \dots, q^{(N)}\}$ for $q^{(i)} \in \mathbb{R}^4_{[0,1]}$ defection is a best response if the transition probabilities of the opponents satisfy conditions (18) and (19).*

$$\sum_{i=1}^{N} (c^{(i)T} \bar{a}^{(i)} - \bar{c}^{(i)T} a^{(i)}) \leq 0 \tag{18}$$

while,

$$\sum_{i=1}^{N} \bar{a}^{(i)} \neq 0 \tag{19}$$

Proof. For defection to be a best response the derivative of the utility at the point $p = (0, 0, 0, 0)$ must be negative. This would indicate that the utility function is only declining from that point onwards.

Substituting $p = (0, 0, 0, 0)$ in equation (15) gives:

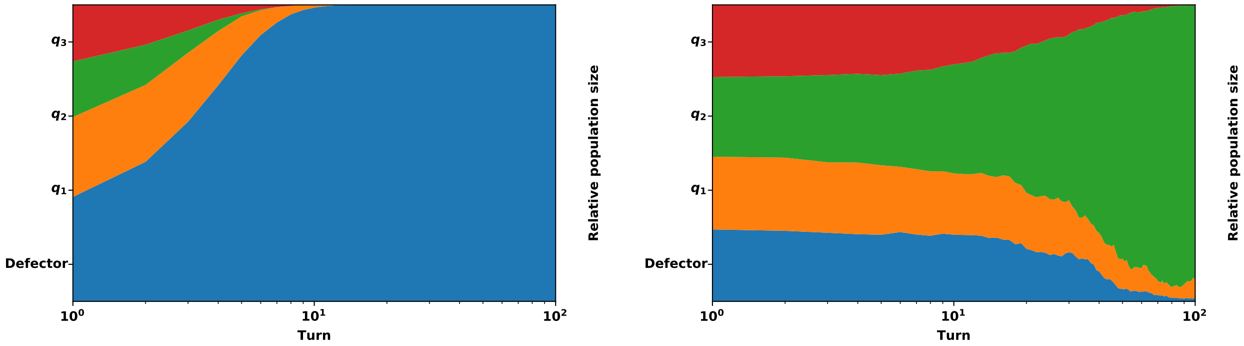
$$\sum_{i=1}^{N} \frac{(c^{(i)T} \bar{a}^{(i)} - \bar{c}^{(i)T} a^{(i)})}{(\bar{a}^{(i)})^2} \tag{20}$$

The sign of the numerator $\sum_{i=1}^{N} (c^{(i)T} \bar{a}^{(i)} - \bar{c}^{(i)T} a^{(i)})$ can vary based on the transition probabilities of the opponents. The denominator can not be negative, and otherwise is always positive. Thus the sign of the derivative is negative if and only if $\sum_{i=1}^{N} (c^{(i)T} \bar{a}^{(i)} - \bar{c}^{(i)T} a^{(i)}) \leq 0$. □

In an environment where defection is the best response the average payoff of a defector is always higher than any other strategy can achieve. If a setting where in each the prevalence of each type of strategy was determined by that strategy's success in the previous round is considered, then in a population such that (18) and (19) hold, defection would prevail; thus cooperation would never occur. This is demonstrated in Figures 5a and 5b. Lemma 5 is the last theoretical result presented in this manuscript. The following section focuses on numerical experiments.

4 Numerical experiments

The results of this section rely on estimating memory-one best responses, but as stated in Section 3, estimating best responses analytically can quickly become an intractable problem. As a result, best responses will be estimated heuristically using Bayesian optimisation [24]. Bayesian optimisation is a global optimisation algorithm that has proven to outperform many other popular algorithms [16]. The algorithm builds a bayesian understanding of the objective function which it is well suited to the multiple local optimas in the described search area of this work. Differential evolution [30] was also considered, however it was not selected due to Bayesian being computationally more efficient.



(a) For opponents $q_1 = (\frac{371}{1250}, \frac{4693}{25000}, \frac{4037}{50000}, \frac{18461}{25000})$, $q_2 = (\frac{48841}{100000}, \frac{30587}{50000}, \frac{76591}{100000}, \frac{25921}{50000})$ and $q_3 = (\frac{22199}{100000}, \frac{87073}{100000}, \frac{646}{3125}, \frac{91861}{100000})$ conditions (18) and (19) hold and Defector takes over the population.

(b) For opponents $q_1 = (\frac{69773}{100000}, \frac{21609}{100000}, \frac{97627}{100000}, \frac{623}{100000})$, $q_2 = (\frac{12649}{50000}, \frac{43479}{100000}, \frac{38969}{50000}, \frac{19769}{100000})$ and $q_3 = (\frac{96703}{100000}, \frac{54723}{100000}, \frac{24317}{25000}, \frac{35741}{50000})$ (18) fails and (19) holds and Defector does not take over the population.

As an example of the algorithm's usage let's consider the optimisation problem of (13). Figure 6 illustrates the change of the utility function over iterations of the algorithm. The default number of iterations that has been used in this work is 60. After 60 calls the convergence of the utility is checked. If the optimised utility has changed in the last 10% iterations then a further 20 iterations are considered.

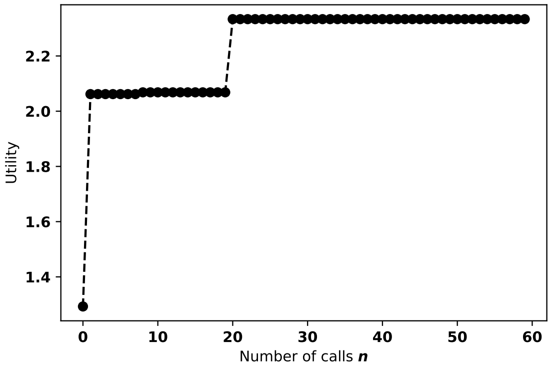


Figure 6: Utility over time of calls using Bayesian optimisation. The opponents are $q^{(1)} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $q^{(2)} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The best response obtained is $p^* = (0, \frac{11}{50}, 0, 0)$

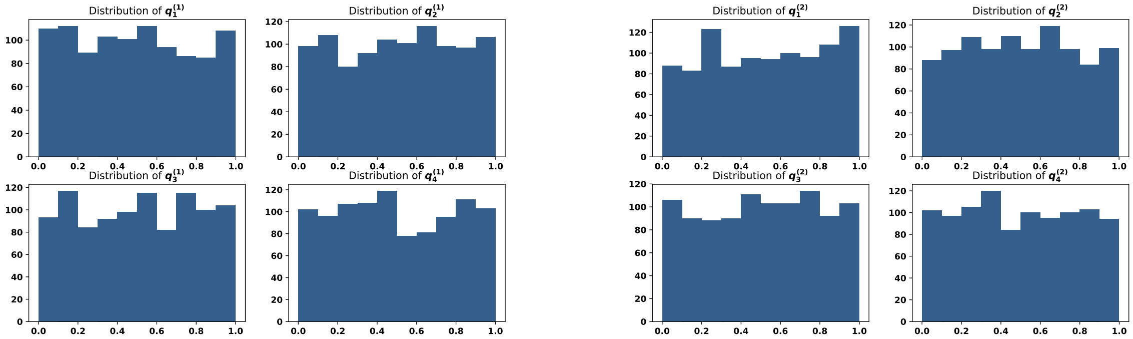
The rest of the section is structured as follows. In Section 4.1, Bayesian optimisation is used to generate a data set containing memory-one best responses against a number of random opponents. The extortionate behaviour of these best responses is then evaluated using a method introduced in [21]. In Section 4.2, a similar data set and approach is discussed but this time the best responses are memory-one best responses to a number of opponents and a copy of themselves. Finally, Section 4.3 compares the performances of memory-one and longer-memory best responses against a number of opponents.

4.1 Best response memory-one strategies for $N = 2$

As briefly discussed in Section 1, zero-determinants have been praised for their robustness against a single opponent. Zero-determinants' extortionate behaviour reassures that the strategies will never lose a game. Though extortion works in pairwise interactions, this paper argues that in multi opponent interactions, where the payoffs matter, strategies trying to exploit their opponents suffer.

Compared to zero-determinants, best response memory-one strategies, that have a theory of mind of their opponents, utilise their behaviour in order to gain the most from their interactions. The question that arises then is whether best response strategies are the optimal because they behave in an extortionate way. To estimate a strategy's extortionate behaviour the SSE method as described in [21] is used. SSE is defined as how far a strategy is from behaving extortionate, thus a high SSE implies a non extortionate behaviour.

A data set of best response memory-one strategies when $N = 2$ has been generated which is available at [12]. The data set contains a total of 1000 trials corresponding to 1000 different instances of a best response strategy. For each trial a set of 2 opponents is randomly generated and the memory-one best response against them is estimated. Though the probabilities q_i of the opponents are randomly generated, Figures 7a and 7b, show that they are uniformly distributed over the trials. Thus, the full space of possible opponents has been covered.



(a) Distributions of first opponents' probabilities.

(b) Distributions of second opponents' probabilities.

The SSE method has been applied to the data set. It's distribution is given in Figure 10 and a statistics summary in Table 2. The distribution of SSE is skewed to the left, indicating that the best response does exhibit extortionate behaviour, however, the best response is not uniformly extortionate. A positive measure of skewness and kurtosis indicates a heavy tail to the right. Therefore, in several cases the strategy is not trying to extortionate the opponents.

So though the best response strategy can exhibit extortionate behaviour, it's performance is maximised by behaving in a more adaptable way than zero-determinant strategies. This analysis is extended to an evolutionary setting.

4.2 Memory-one best responses in evolutionary dynamics

As mentioned in Section 2, the IPD is commonly studied in Moran processes, and generally, in evolutionary processes. In these processes self interactions are key. This section extends the formulation of best responses to evolutionary settings, more specifically, the optimisation problem of (13) is extended to include self interactions.

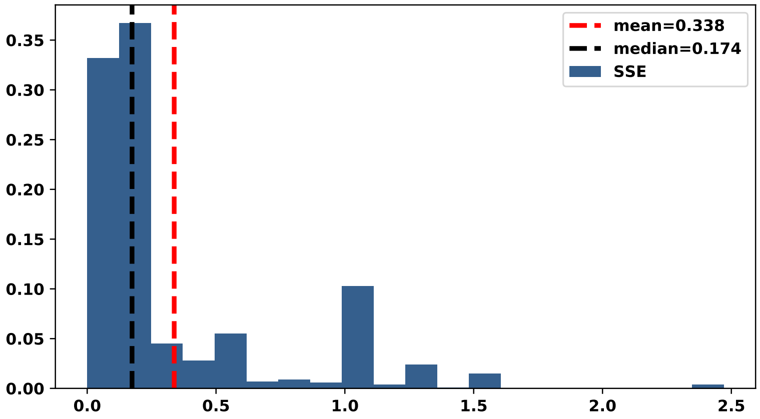


Figure 8: Distribution of SSE for memory-one best responses, when $N = 2$.

	SSE
count	1000.00000
mean	0.33762
std	0.39667
min	0.00000
5%	0.02078
25%	0.07597
50%	0.17407
95%	1.05943
max	2.47059
median	0.17407
skew	1.87231
kurt	3.60029

Table 1: Summary statistics SSE of best response memory one strategies included tournaments of $N = 2$.

Self interactions can be incorporated in the formulation that has been used so far. The utility is given by,

$$\frac{1}{N} \sum_{i=1}^{N} {u_q}^{(i)}(p) + u_p(p). \tag{21}$$

and the optimisation problem of (13) is re written as,

$$\begin{aligned} \max_p : & \ \frac{1}{N} \sum_{i=1}^{N} {u_q}^{(i)}(p) + u_p(p) \\ \text{such that} : & \ p \in \mathbb{R}_{[0,1]} \end{aligned} \tag{22}$$

For determining the memory-one best response in an evolutionary setting, an algorithmic approach is considered, called *best response dynamics*. Best response dynamics are commonly used in evolutionary game theory. They represent a class of strategy updating rules, where players in the next round are determined by their best responses to some subset of the population. The best response dynamics approach used in this manuscript is given by Algorithm 1.

Algorithm 1: Best response dynamics Algorithm

```
$p^{(t)} \leftarrow (1, 1, 1, 1)$;
while $p^{(t)} \neq p^{(t-1)}$ do
    $p^{(t+1)} = \text{argmax} \frac{1}{N} \sum_{i=1}^{N} {u_q}^{(i)}(p^{(t+1)}) + u_p^{(t)}(p^{(t+1)})$;
end
```

The best response dynamics algorithm starts by setting an initial solution $p^{(1)} = (1, 1, 1, 1)$, and repeatedly finds a strategy that maximises (22) using Bayesian optimisation. The algorithm stops once a cycle (a sequence of iterated evaluated points) is detected. A numerical example of the algorithm is given in Figure 9.

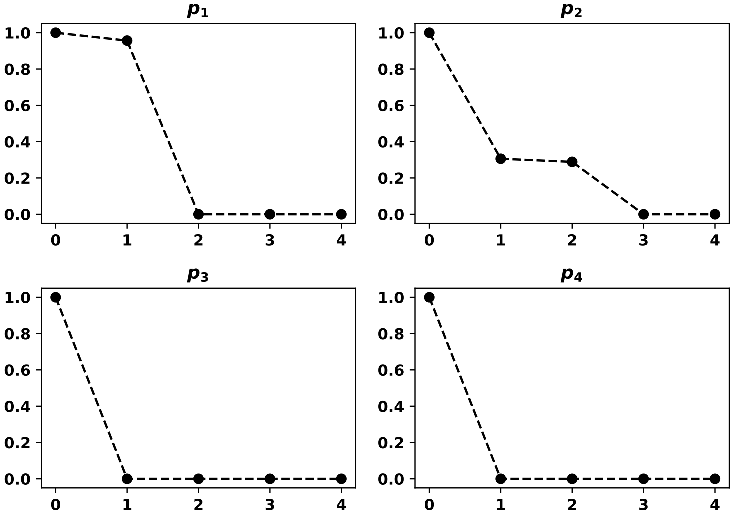


Figure 9: Best response dynamics with $N = 2$. More specifically, for $q^{(1)} = (\frac{59}{250}, \frac{1031}{10000}, \frac{99}{250}, \frac{1549}{10000})$ and $q^{(2)} = (\frac{133}{2000}, \frac{803}{2000}, \frac{9179}{10000}, \frac{2001}{2500})$.

The algorithm has been used to estimate the best response in an evolutionary setting for each of the 1000 pairs of opponents described in Section 4.1. These are also included in the data set [12], and moreover, the SSE method has also been applied. The distribution of SSE is given by Figure 10 and a statistical summary by Table 2.

Similarly to the results of Section 4.1, the evolutionary best response strategy does not behave uniformly extortionate. A larger value of both the kurtosis and the skewness of the SSE distribution indicates that in evolutionary settings a memory-one best response is even more adaptable.

The difference between best responses in tournaments and in evolutionary settings are further explored by Figure 11. Though, Table 3 details that no statistically significant differences has been found, from Figure 11, it seems that evolutionary best response has a higher p_2 median. Thus, they more likely to forgive after being tricked.

Best Response Median in:	Tournament	Evolutionary Settings	p-values
Distribution p_1	0.0	0.00000	0.0
Distribution p_2	0.0	0.19847	0.0
Distribution p_3	0.0	0.00000	0.0
Distribution p_4	0.0	0.00000	0.0

Table 3: A non parametric test, Wilcoxon Rank Sum, has been performed to tests the difference in the medians. A non parametric test is used because is evident that the data are skewed.

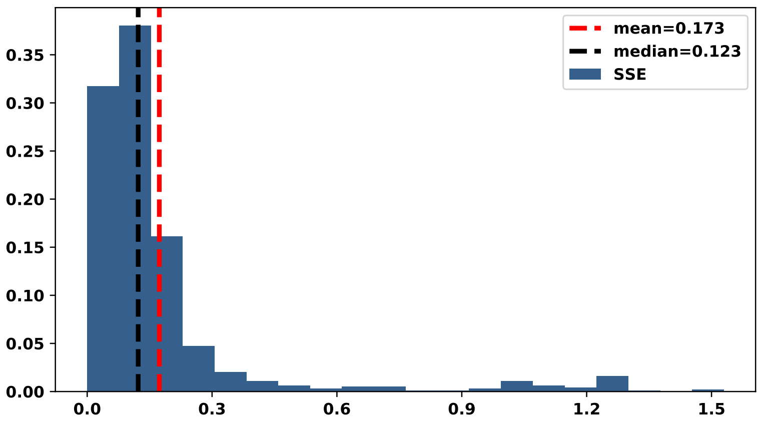


Figure 10: Distribution of SSE of best response memory-one strategies in evolutionary settings, when when $N = 2$.

	SSE
count	1000.00000
mean	0.17326
std	0.23489
min	0.00001
5%	0.01497
25%	0.05882
50%	0.12253
95%	0.67429
max	1.52941
median	0.12253
skew	3.41839
kurt	11.92339

Table 2: Summary statistics SSE of best response memory-one strategies in evolutionary settings, when when $N = 2$.

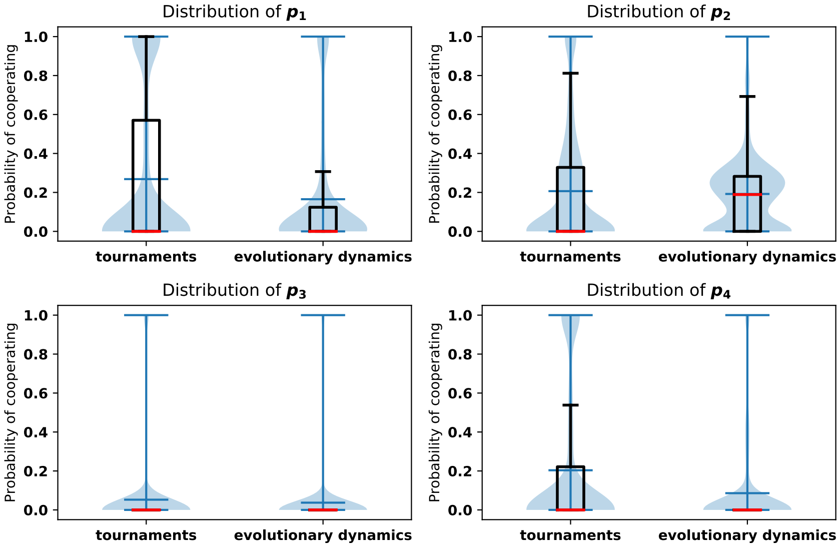


Figure 11: Distributions of p^* for both best response and evo memory-one strategies.

4.3 Longer memory best response

This section focuses on the memory size of strategies. The effectiveness of memory in the IPD has been previously explored in the literature, as discussed in Section 1, however, none of the previous works has compared the performance of longer-memory strategies to memory-one best responses.

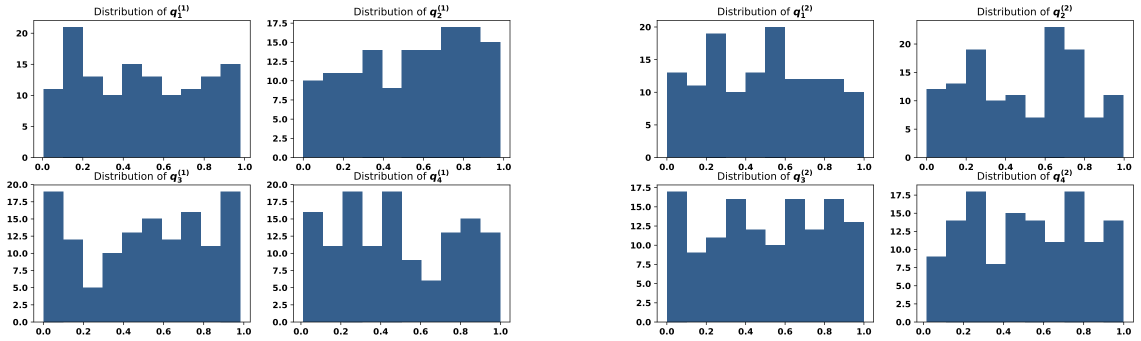
In [13], a strategy called *Gambler* which makes probabilistic decisions based on the opponent's n_1 first moves, the opponent's m_1 last moves and the player's m_2 last moves was introduced. In this manuscript Gambler $n_1 = 2, m_1 = 1$ and $m_2 = 1$ is used as the longer-memory strategy.

By considering the opponent's first two moves, the opponents last move and the player's last move, there are only 16 ($4 \times 2 \times 2$) possible outcomes that can occur, furthermore, Gambler also makes a probabilistic decision of cooperating in the opening move. Thus, Gambler is a function $f : \{\text{C, D}\}^{16 \cup 1} \to (0, 1)_{\mathbb{R}}$. This can be hard coded as an element of $[0, 1]_{\mathbb{R}}^{16+1}$, one probability for each outcome plus the opening move. Hence, compared to (13), finding an optimal Gambler is a 17 dimensional problem given by:

$$\begin{aligned} \max_{p} : & \sum_{i=1}^{N} {U_q}^{(i)}(f) \\ \text{such that} : & \ f \in \mathbb{R}_{[0,1]}^{17} \end{aligned} \tag{23}$$

Note that (11) can not be used here for the utility of Gambler, and actual simulated players are used. This is done using [1] with 500 turns and 200 repetitions, moreover, (23) is solved numerically using Bayesian optimisation.

Similarly to previous sections, a large data set has been generated with instances of an optimal Gambler and a memory-one best response, available at [12]. For each trial two random opponents have been selected. The distributions of their transition probabilities are given in Figures 12a and 12a. The results of this section are base on a total of 130 trials.



(a) Distributions of first opponents' probabilities for longer memory experiment.

(b) Distributions of second opponents' probabilities for longer memory experiment.

The utilities of both strategies are plotted against each other in Figure 13. Though Gambler has an infinite memory (in order to remember the opening moves of the opponent) the information the strategy considers is not significantly larger than memory-one strategies. Even so, it is evident from Figure 13 that Gambler will always performs the same or better. This seems to be at odd with the result of [28] that against a memory-one opponent having a longer memory will not give a strategy any advantage. However, against two memory-one opponents Gambler performs better than the optimal memory-one strategy, thus having a shorter memory in this setting is limiting.

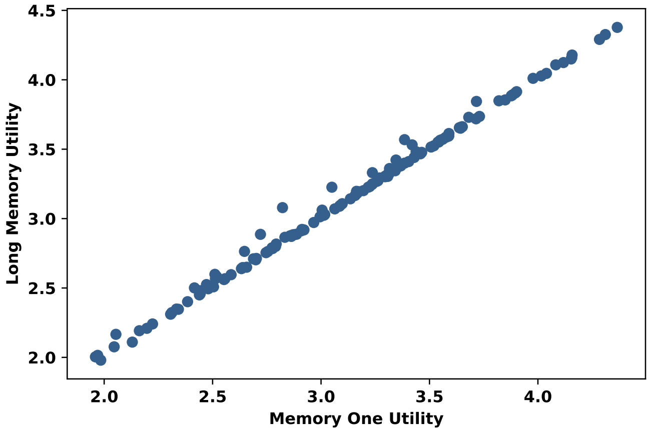


Figure 13: Utilities of Gambler and best response memory-one strategies for 120 different pair of opponents.

5 Conclusion

This manuscript considers *best response* strategies in the IPD game, and more specifically, *memory-one best responses*. It has proven that there is a compact way of identifying a memory-one best response to a group of opponents, and moreover, that there exists a condition for which in an environment of memory-one opponents defection is the dominant and stable choice. The later parts of this paper focused on a series of empirical results where it was shown that the performance and the evolutionary stability of memory-one strategies rely not on extortion but on adaptability. Finally, it was shown that memory-one strategies' performance is limited by their memory in cases where they interact with multiple opponents.

Following the work described in [25], where it was shown that the utility between two memory-one strategies can be estimated by a Markov stationary state, we proved that the utilities can be written as a ration of two quadratic forms in R^4, Theorem 1. This was extended to include multiple opponents as the IPD is commonly studied in such situations, Theorem 2. The formulation of Theorem 2 and the result that the utility has a form allowed us to introduce an approach for identifying memory-one best responses to any number of opponents, Lemma 4. This does not only have game theoretic novelty, but also a mathematical novelty of solving quadratic ratio optimisation problem where the quadratics are non concave. The results of Lemma 4 were also used to define a condition for which defection is known to be a best response.

This manuscript also presented empirical results. These results were mainly to investigate the behaviour of memory-one strategies and their limitations. In Sections 4.1 and 4.2, a large data set which contained best responses in tournaments and in evolutionary settings for $N = 2$ was generated. This allowed us to investigate their respective behaviours and whether it was extortionate acts that made them most favorable strategies. However, it was shown that it was not extortion but adaptability that allowed the strategies to gain the most from their interactions. In evolutionary settings it was specifically shown that being adaptable and being able to forgive after being tricked were key factors. In Section 4.3, the performance of memory-one strategies was put against the performance of a longer memory strategy called Gambler. There were several cases where Gambler would outperform the memory-one strategy, however a memory-one strategy never managed to outperform a Gambler. This result occurred whilst considering a Gambler with a sufficiently larger memory but not a sufficiently larger amount of information regarding the game.

All the empirical results presented in this manuscript have been for the case of $N = 2$. In future work we would consider larger values of N, however, we believe that for larger values of N the results that have been presented here would only be more evident.

6 Acknowledgements

A variety of software libraries have been used in this work:

- The Axelrod library for IPD simulations [1].
- The Scikit-optimize library for an implementation of Bayesian optimisation [14].
- The Matplotlib library for visualisation [15].
- The SymPy library for symbolic mathematics [23].
- The Numpy library for data manipulation [32].

References

[1] The Axelrod project developers . Axelrod: 4.4.0, April 2016.

[2] Christoph Adami and Arend Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4:2193, 2013.

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

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

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

[6] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[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] Merrill M. Flood. Some experimental games. *Management Science*, 5(1):5–26, 1958.

[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] Nikoleta E. Glynatsi. Raw data for: "Stability of defection, optimisation of strategies and the limits of memory in the Prisoner's Dilemma.", September 2019.

[13] 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.

[14] 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.

[15] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

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

[17] Gubjorn Jonsson and Stephen Vavasis. Accurate solution of polynomial equations using macaulay resultant matrices. *Mathematics of computation*, 74(249):221–262, 2005.

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

[19] 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.

[20] Vincent Knight, Marc Harper, Nikoleta E. Glynatsi, and Owen Campbell. Evolution reinforces cooperation with the emergence of self-recognition mechanisms: An empirical study of strategies in the moran process for the iterated prisoners dilemma. *PLOS ONE*, 13(10):1–33, 10 2018.

[21] Vincent A. Knight, Marc Harper, Nikoleta E. Glynatsi, and Jonathan Gillard. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR*, abs/1904.00973, 2019.

[22] 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.

[23] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, 2017.

[24] 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.

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

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

[27] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56, 1993.

[28] 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.

[29] 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.

[30] 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.

[31] Steve Tadelis. *Game theory: an introduction.* Princeton University Press, 2013.

[32] S. Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Appendices

A Proofs of the Theorems

A.1 Proof of Theorem 1

Proof. The utility of a memory one player p against an opponent q, $u_q(p)$, can be written as a ratio of two quadratic forms on R^4.

In Section 2, it was discussed that $u_q(p)$ its the product of the steady states v and the PD payoffs,

$$u_q(p) = v \cdot (R, S, T, P).$$

More specifically,

$$u_q(p) = \left(\frac{\begin{gathered} p_1p_2(q_1q_2 - 5q_1q_4 - q_1 - q_2q_3 + 5q_3q_4 + q_3) + p_1p_3(-q_1q_3 + q_2q_3) + p_1p_4(5q_1q_3 - 5q_3q_4) + p_3p_4(-3q_2q_3 + 3q_3q_4)+ \\ p_2p_3(-q_1q_2 + q_1q_3 + 3q_2q_4 + q_2 - 3q_3q_4 - q_3) + p_2p_4(-5q_1q_3 + 5q_1q_4 + 3q_2q_3 - 3q_2q_4 + 2q_3 - 2q_4)+ \\ p_1(-q_1q_2 + 5q_1q_4 + q_1) + p_2(q_2q_3 - q_2 - 5q_3q_4 - q_3 + 5q_4 + 1) + p_3(q_1q_2 - q_2q_3 - 3q_2q_4 - q_2 + q_3)+ \\ p_4(-5q_1q_4 + 3q_2q_4 + 5q_3q_4 - 5q_3 + 2q_4) + q_2 - 5q_4 - 1 \end{gathered}}{\begin{gathered} p_1p_2(q_1q_2 - q_1q_4 - q_1 - q_2q_3 + q_3q_4 + q_3) + p_1p_3(-q_1q_3 + q_1q_4 + q_2q_3 - q_2q_4) + p_1p_4(-q_1q_2 + q_1q_3 + q_1 + q_2q_4 - q_3q_4 - q_4)+ \\ p_2p_3(-q_1q_2 + q_1q_3 + q_2q_4 + q_2 - q_3q_4 - q_3) + p_2p_4(-q_1q_3 + q_1q_4 + q_2q_3 - q_2q_4) + p_3p_4(q_1q_2 - q_1q_4 - q_2q_3 - q_2 + q_3q_4 + q_4)+ \\ p_1(-q_1q_2 + q_1q_4 + q_1) + p_2(q_2q_3 - q_2 - q_3q_4 - q_3 + q_4 + 1) + p_3(q_1q_2 - q_2q_3 - q_2 + q_3 - q_4) + p_4(-q_1q_4 + q_2 + q_3q_4 - q_3 + q_4 - 1)+ \\ q_2 - q_4 - 1 \end{gathered}} \right) \tag{24}$$

Let's consider the numerator of the $u_q(p)$. The cross product terms, p_ip_j, are given by

$$\begin{gathered} p_1p_2(q_1q_2 - 5q_1q_4 - q_1 - q_2q_3 + 5q_3q_4 + q_3) + p_1p_3(-q_1q_3 + q_2q_3) + p_1p_4(5q_1q_3 - 5q_3q_4) + p_3p_4(-3q_2q_3 + 3q_3q_4)+ \\ p_2p_3(-q_1q_2 + q_1q_3 + 3q_2q_4 + q_2 - 3q_3q_4 - q_3) + p_2p_4(-5q_1q_3 + 5q_1q_4 + 3q_2q_3 - 3q_2q_4 + 2q_3 - 2q_4). \end{gathered}$$

The cross products' expression can be re written in a matrix format given by (25).

$$(p_1, p_2, p_3, p_4)\frac{1}{2}\begin{bmatrix} 0 & -(q_1 - q_3)(q_2 - 5q_4 - 1) & q_3(q_1 - q_2) & -5q_3(q_1 - q_4) \\ -(q_1 - q_3)(q_2 - 5q_4 - 1) & 0 & (q_2 - q_3)(q_1 - 3q_4 - 1) & (q_3 - q_4)(5q_1 - 3q_2 - 2) \\ q_3(q_1 - q_2) & (q_2 - q_3)(q_1 - 3q_4 - 1) & 0 & 3q_3(q_2 - q_4) \\ -5q_3(q_1 - q_4) & (q_3 - q_4)(5q_1 - 3q_2 - 2) & 3q_3(q_2 - q_4) & 0 \end{bmatrix}\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \tag{25}$$

Note that the coefficients are multiplied by $\frac{1}{2}$ because they are added twice.

Similarly, the linear terms,

$$p_1(-q_1q_2 + 5q_1q_4 + q_1) + p_2(q_2q_3 - q_2 - 5q_3q_4 - q_3 + 5q_4 + 1) + p_3(q_1q_2 - q_2q_3 - 3q_2q_4 - q_2 + q_3) + \\ p_4(-5q_1q_4 + 3q_2q_4 + 5q_3q_4 - 5q_3 + 2q_4).$$

can be written using a matrix format, (26).

$$(p_1, p_2, p_3, p_4)\begin{bmatrix} q_1(q_2 - 5q_4 - 1) \\ -(q_3 - 1)(q_2 - 5q_4 - 1) \\ -q_1q_2 + q_2q_3 + 3q_2q_4 + q_2 - q_3 \\ 5q_1q_4 - 3q_2q_4 - 5q_3q_4 + 5q_3 - 2q_4 \end{bmatrix} \tag{26}$$

Finally the constant term of the numerator, which is obtained by substituting $p = (0, 0, 0, 0)$ is given by (27).

$$q_2 - 5q_4 - 1 \tag{27}$$

Equations (25), (26) and (27) are combined and the numerator of can be written as,

$$\frac{1}{2}p\begin{bmatrix} 0 & -(q_1 - q_3)(q_2 - 5q_4 - 1) & q_3(q_1 - q_2) & -5q_3(q_1 - q_4) \\ -(q_1 - q_3)(q_2 - 5q_4 - 1) & 0 & (q_2 - q_3)(q_1 - 3q_4 - 1) & (q_3 - q_4)(5q_1 - 3q_2 - 2) \\ q_3(q_1 - q_2) & (q_2 - q_3)(q_1 - 3q_4 - 1) & 0 & 3q_3(q_2 - q_4) \\ -5q_3(q_1 - q_4) & (q_3 - q_4)(5q_1 - 3q_2 - 2) & 3q_3(q_2 - q_4) & 0 \end{bmatrix}p^T + \\ \begin{bmatrix} 0 & -(q_1 - q_3)(q_2 - 5q_4 - 1) & q_3(q_1 - q_2) & -5q_3(q_1 - q_4) \\ -(q_1 - q_3)(q_2 - 5q_4 - 1) & 0 & (q_2 - q_3)(q_1 - 3q_4 - 1) & (q_3 - q_4)(5q_1 - 3q_2 - 2) \\ q_3(q_1 - q_2) & (q_2 - q_3)(q_1 - 3q_4 - 1) & 0 & 3q_3(q_2 - q_4) \\ -5q_3(q_1 - q_4) & (q_3 - q_4)(5q_1 - 3q_2 - 2) & 3q_3(q_2 - q_4) & 0 \end{bmatrix}p + q_2 - 5q_4 - 1$$

or as,

$$\frac{1}{2}pQp^T + cp + a$$

where $Q \in \mathbb{R}^{4\times 4}$ ia a square matrix whose diagonal elements are all equal to zero, and is defined by the transition probabilities of the opponent q_1, q_2, q_3, q_4 as follows:

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

$c \in \mathbb{R}^{4\times 1}$ is similarly defined by:

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

and $a = -q_2 + 5q_4 + 1$.

The same process is done for the denominator. □

A.2 Proof of Theorem 3

Proof. Utility $u_q(p)$ is non concave and neither are it's numerator or denominator.

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

$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda)f(x_2). \tag{28}$$

Let f be $u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}$. 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 $\lambda = 0.1$, direct substitution in (28) gives:

$$\begin{gathered} u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}\left(0.1\left(\frac{1}{4},\frac{1}{2},\frac{1}{5},\frac{1}{2}\right) + 0.9\left(\frac{8}{10},\frac{1}{2},\frac{9}{10},\frac{7}{10}\right)\right) \geq 0.1 \times u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}\left(\left(\frac{1}{4},\frac{1}{2},\frac{1}{5},\frac{1}{2}\right)\right) + 0.9 \times u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}\left(\left(\frac{8}{10},\frac{1}{2},\frac{9}{10},\frac{7}{10}\right)\right) \Rightarrow \\ 1.485 \geq 0.1 \times 1.790 + 0.9 \times 1.457 \Rightarrow \\ 1.485 \geq 1.490 \end{gathered}$$

which can not hold. Thus $u_{(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})}$ is not concave. Because the concavity condition fails for at least one point of $u_q(p)$, $u_q(p)$ is not concave.

Utility $u_q(p)$ is given by (6). As stated in [3] 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:

$$|A|_i \leq 0 \text{ for } i \text{ is odd and } |A|_i \geq 0 \text{ for } i \text{ is even.} \tag{29}$$

For (6), neither $\frac{1}{2}pQp^T + cp + a$ or $\frac{1}{2}p\bar{Q}p^T + \bar{c}p + \bar{a}$ are concave because:

$$\begin{gathered} |Q|_2 = -\left(q_1 - q_3\right)^2\left(q_2 - 5q_4 - 1\right)^2 \text{ and} \\ |\bar{Q}|_2 = -\left(q_1 - q_3\right)^2\left(q_2 - q_4 - 1\right)^2 \end{gathered}$$

are negative. □

A.3 Proof of Lemma 4

Proof. 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:

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

where S_q is given by (14).

The optimisation problem of (13) can be written as:

$$\begin{aligned} \max_p :\ & \sum_{i=1}^{N} {u_q}^{(i)}(p) \\ \text{such that} : p_i & \leq 1 \text{ for } \ \in \{1,2,3,4\} \\ -p_i & \leq 0 \text{ for } \ \in \{1,2,3,4\} \end{aligned} \tag{30}$$

The optimisation problem has two inequality constraints and regarding the optimality this means that:

- either the optimum is away from the boundary of the optimization domain, and so the constraints plays no role;
- or the optimum is on the constraint boundary.

Thus, the following three cases must be considered:

Case 1: The solution is on the boundary and any of the possible combinations for $p_i \in \{0,1\}$ for $i \in \{1,2,3,4\}$ are candidate optimal solutions.

Case 2: The optimum is away from the boundary of the optimization domain and the interior solution p^* necessarily satisfies the condition $\frac{d}{dp}\sum\limits_{i=1}^{N} u_q(p^*) = 0$.

Case 3: The optimum is away from the boundary of the optimization domain but some constraints are satisfied. The candidate solutions in this case are any combinations of $p_j \in \{0,1\}$ and $\frac{d}{dp_k}\sum\limits_{i=1}^{N} u_q^{(i)}(p) = 0$ forall $j \in J$ & $k \in K$ forall J, K where $J \cap K = \emptyset$ and $J \cup K = \{1,2,3,4\}$.

Combining cases 1-3 a set of candidate solution is constructed as:

$$S_q = \left\{ p \in \mathbb{R}^4 \;\middle|\; \begin{array}{l} \bullet \quad p_j \in \{0,1\} \quad \text{and} \quad \dfrac{d}{dp_k}\displaystyle\sum_{i=1}^{N} u_q^{(i)}(p) = 0 \quad \text{forall} \quad j \in J \quad \& \quad k \in K \quad \text{forall} \quad J, K \\ \qquad\qquad\qquad\qquad \text{where} \quad J \cap K = \emptyset \quad \text{and} \quad J \cup K = \{1,2,3,4\}. \\ \\ \bullet \quad p \in \{0,1\}^4 \end{array} \right\}.$$

This set is denoted as S_q and the optimal solution to (13) is the point from S_q for which the utility is maximised.

□