An Empirical Study of Invasion and Resistance for Iterated Prisoner's Dilemma Strategies

Vincent Knight Marc Harper Nikoleta E. Glynatsi Owen Campbell

Abstract

The Iterated Prisoner's Dilemma is a well established framework for the study of emergent behaviour. In this paper an extensive numerical study of the evolutionary dynamics of this framework are presented. Fixation probabilities for Moran processes are obtained for 164 different strategies. It is found that players with long memories and sophisticated behaviours outperform many strategies that perform well in a two player setting.

1 Introduction

The Prisoner's Dilemma (PD) [8] is a fundamental two player game used to model a large variety of strategic interactions. Each player can choose between cooperation (C) or defection (D). The decisions are made simultaneously and independently. The payoffs of the game are defined by the matrix $\begin{pmatrix} R & S \\ T & P \end{pmatrix}$, where $T > R > P > S$ and $2R > T + S$. The PD is a one round game, but is commonly studied in a manner where the prior outcomes matter. This extended form is called the Iterated Prisoner's Dilemma (IPD). As described in [5, 12, 20] a number of strategies have been developed to take advantage of the history of play. Recently, some strategies referred to as Zero Determinant strategies [20] even manipulate players through extortion.

The Moran Process [18] is a model of evolutionary population dynamics that has been used to gain insights about the evolutionary stability in a number of settings (more details given in Section 1.1). Several earlier works have studied iterated games in the context of the prisoner's dilemma [19, 24], however these often make simplifying assumptions and/or do not consider sophisticated behaviour: only considering strategies that either cooperate or defect.

This manuscript provides a detailed numerical analysis of **164** complex and adaptive strategies for the IPD. This is made possible by the Axelrod library [25], an effort to provide software for reproducible research for the IPD. The library now contains over 186 parameterized strategies including classics like TitForTat and WinStayLoseShift, as well as recent variants such as OmegaTFT, Zero determinant and other memory one strategies, strategies based on finite state machines, lookup tables, neural networks, and other machine learning based strategies, and a collection of novel strategies. Not all strategies have been considered for this study: same use knowledge of the game such as the number of turns and others have a high computational run time. The large number of strategies are available thanks to the open source nature of the project with over 40 contributions made by different programmers and researchers [12]. Three of the considered strategies are finite state machines trained specifically for Moran processes (described further in Section 1.2.

The library can conduct matches, tournaments and population dynamics with variations including noise and spatial structure. The strategies and simulation frameworks are automatically tested to an extraordinarily high degree of coverage in accordance with best research software practices.

Fixation probabilities for all pairs of strategies are presented, identifying those that are effective invaders and those resistant to invasion, for population sizes $N = 2$ to $N = 14$.

In particular the following questions are addressed:

1. What strategies are good invaders?
2. What strategies are good at resisting invasion?
3. How does the population size affect these findings?

While the results agree with some of the published literature, it is found that:

1. Zero determinant strategies are not particularly effective for $N > 2$
2. Complex strategies can be effective, and in fact can naturally evolve through evolutionary processes to outperform intelligently designed strategies.

3. Strong resistors specifically evolve or have a handshake mechanism. This has the potential to offer insight in to organisms such as antibacterial resistant bacteria [7].

1.1 The Moran Process

Figure 1 shows a diagrammatic representation of the Moran process, a stochastic birth death process on a finite population in which the population size stays constant over time. Individuals are **selected** according to a given fitness landscape. Once selected, the individual is reproduced and similarly another individual is chosen to be removed from the population. In some settings mutation is also considered but without mutation (the case considered in this work) this process will arrive at an absorbing state where the population is entirely made up of players of one strategy. The probability with which a given strategy is the survivor is called the *fixation probability*. A more detailed analytic description of this is given in Section 2.

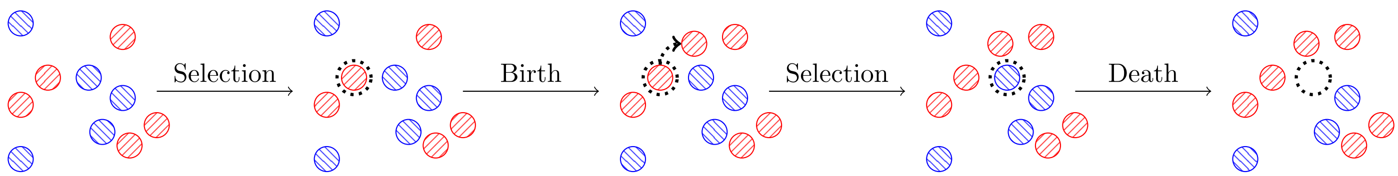


Figure 1: A diagrammatic representation of a Moran process

The Moran process was initially introduced in [18]. It has since been used in a variety of settings including the understanding of the spread of cooperative and non-cooperative behaviour such as cancer [27] and the emergence of cooperative behaviour in spatial topologies [3]. However these works mainly consider non-sophisticated strategies. Some work has looked at evolutionary stability of strategies within the Prisoner's Dilemma [15] but this is not done in the more widely used setting of the Moran process, rather in terms of infinite population stability. In [6] Moran processes are studied in a theoretical framework for a small subset of strategies. The subset included memory one strategies, strategies that recall the events of the previous round only.

Of particular interest are the Zero determinant strategies introduced in [20] and praised in [24] it was argued that generous ZD strategies are robust against invading strategies. However, in [13] a strategy using machine learning techniques was capable of resisting invasion and also able to invade any memory one strategy. Recent work [9] has investigated the effect of memory length on strategy performance and the emergence of cooperation but this is not done in Moran process context and only considers specific cases of memory 2 strategies. In [1] it was recognised that many Zero determinant strategies do not fare well against themselves. This is a disadvantage for the Moran process where the best strategies cooperate well with other players using the same strategy.

1.2 Strategies considered

To carry out this large numerical experiment, 164 strategies, listed in Appendix A are used library. There are 43 stochastic and 121 deterministic strategies. Their memory depth, defined by the number of rounds of history used by the strategy each round, is shown in Table 1. The memory depth is infinite if the strategy uses the entire history of play (whatever its length). For example, a strategy that utilizes a handshaking mechanism where the opponents actions on the first few rounds of play determines the strategies subsequent behavior would have infinite memory depth.

A number of these strategies have been trained with reinforcement learning algorithms.

- Evolved ANN: a neural network based strategy;
- Evolved LookerUp: a lookup table based strategy;
- PSO Gambler: a stochastic version of the lookup table based strategy;
- Evolved HMM: a hidden Markov model based strategy.

A part from the PSO Gambler strategy, which was trained using a particle swarm optimisation algorithm, all these strategies are trained with an evolutionary algorithm that perturbs strategy parameters and optimizes the mean total score against all other opponents [2]. Variation is introduced via mutation and crossover of parameters, and the best performing strategies are carried to the next generation along with new variants. Similar methods appear in the literature [4].

More information about each player can be obtained in the documentation for [25] and a detailed description of the performance of these strategies in IPD tournaments will be described in upcoming manuscript.

All of the training code is available in the Axelrod repository with documentation to train new strategies easily. Training typically takes less than 200 iterations and can be completed within several hours on commodity hardware.

There are three further strategies trained specifically for this study; Trained FSM 1, 2, and 3 (TF1 - TF3). These are based on finite state machines of 16, 16, and 8 states respectively (see Figures 2, 3 and 4).

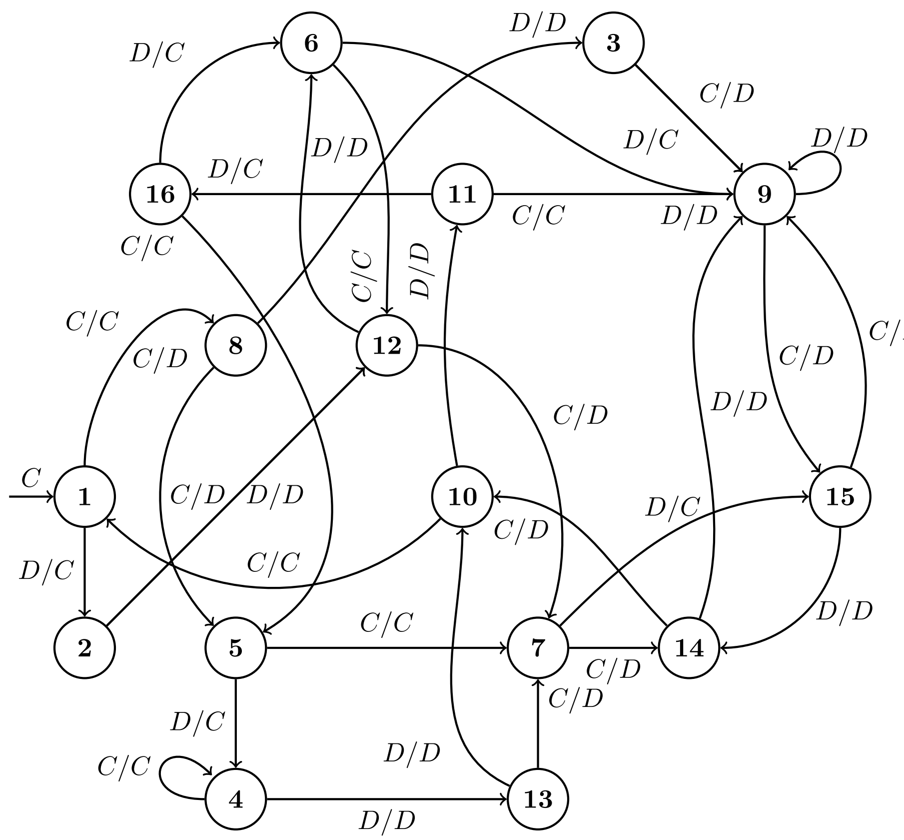


Figure 2: TF1: a 16 state finite state machine.

As opposed to the previously described strategies, these strategies were trained with the objective function of **mean fixation probabilities for Moran processes** starting at initial population states consisting of $N/2$ individuals of the training candidates and $N/2$ individuals of an opponent strategy, taken from a selection of 150 opponents from the axelrod library:

- TF1 $N = 12$, 0% noise, 10000 repetitions per match
- TF2 $N = 10$, 0% noise, 10000 repetitions per match
- TF3 $N = 8$, 1% noise, 100 repetitions per match

Each matchup of players was run to fixation for the specified number of repetitions to estimate the absorption probabilities. The trained algorithms were run for less than 50 generations. Training data for this is available at [11].

TF3 cooperates and defects with various cycles depending on the opponent's actions. TF3 will mutually cooperate with any strategy and only tolerates a few defections before defecting for the rest of match. It is similar to but not exactly the same as Fool Me Once, a strategy that cooperates until the opponent has defected twice (not necessarily consecutively), and defects indefinitely thereafter. Though a product of training with a Moran objective. Interestingly it has no handshake mechanism, this could be due to the fact that it is trained in a noisy environment.

TF2 always starts with CD and will defect against opponents that start with DD. It plays CDD against itself and then cooperates thereafter. There is a longer complex handshake which eventually results in mutual cooperation with Firm but Fair, Fortress3, Fortress4, and Grofman (always) and Evolved HMM 5 and GTFT (depending on the random seed).

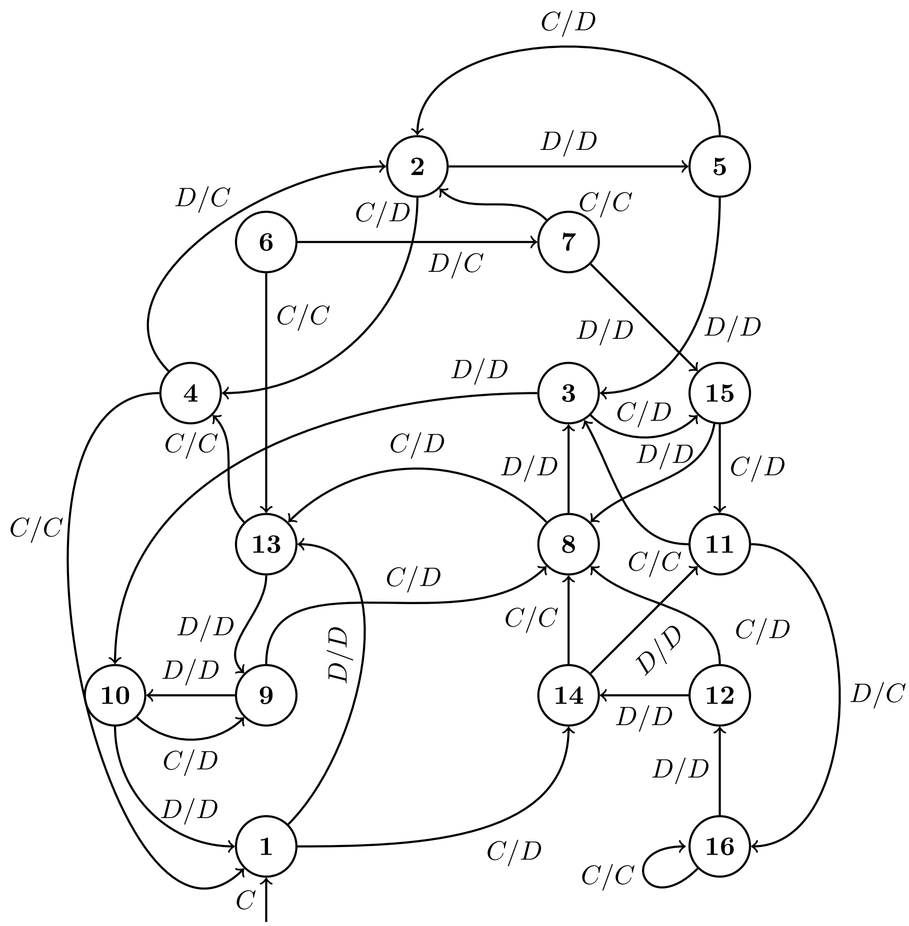


Figure 3: TF2: a 16 state finite state machine.

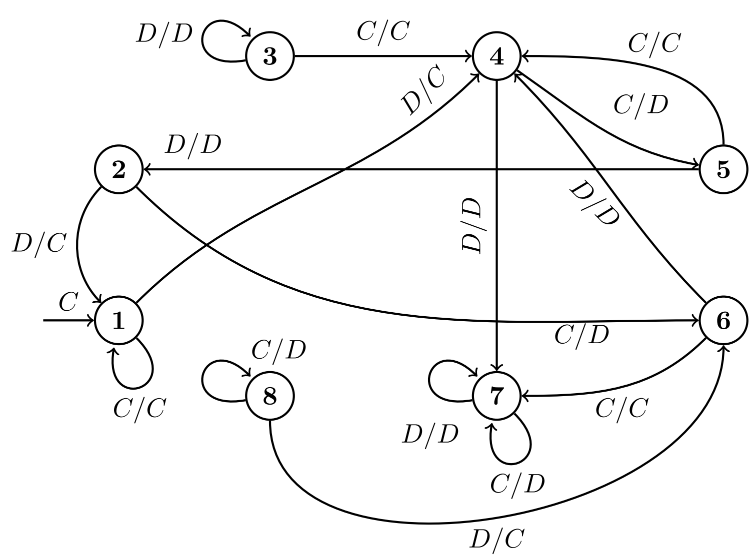


Figure 4: TF3: an 8 state finite state machine.

TF1 has an initial handshake of CCD and cooperates if the opponent matches. However if the opponent later defects, TF1 will respond in kind, so the handshake is not permanent. Only one player (Prober 4 [16]) manages to achieve cooperation with TF1 after about 20 rounds of play.

For both TF1 and TF2 a handshake mechanism naturally emerges from the structure of the underlying finite state machine. This behavior is an outcome of the evolutionary process and is in no way hard-coded or included via an additional mechanism.

Memory Depth	0	1	2	3	4	5	6	9	10	11	12	16	20	40	200	∞
Count	3	28	12	8	2	6	1	1	5	1	1	3	2	2	1	88

Table 1: Memory depth

1.3 Data collection

Each strategy pair is run for 1000 repetitions of the Moran process to fixation with starting population distributions of $(1, N-1)$, $(N/2, N/2)$ and $(N-1, 1)$, for N from 2 through 14. The fixation probability is then empirically computed for each combination of starting distribution and value of N. The axelrod library can carry out exact simulations of the Moran process. Since some of the strategies have a high computational cost or are stochastic, samples are taken from a large number of match outcomes for the pairs of players for use in computing fitnesses in the Moran process. This approach was verified to agree with unsampled calculations to a high degree of accuracy in specific cases. This is described in Algorithms 1 and 2..

Algorithm 1 Data Collection

```
for player one in players list do
  for player two in (players list - player one) do
    pair ← (player one, player two)
    for starting population distributions in [(1, N−1), (N/2, N/2), (N−1, 1)] do
      while repetitions ≤ 1000 do
        simulate moran process*(pair, starting distribution)
      end while
      return fixation probabilities
    end for
  end for
end for
```

Algorithm 2 Moran process

```
initial population ← (pair, starting distribution)
population ← initial population
while repetitions ≤ max repetitions do
  for player in population do
    for opponent in (population - player) do
      match ← (player, opponent)
      results ← cache (match)
    end for
  end for
  population ← sorted(results)
  parent ← first in population
  child ← parent
  kill off ← random player from population
  population ← child replaces kill off
end while
```

Section 2 will further validate the methodology by comparing simulated results to analytical results in some cases. The main results of this manuscript are presented in Section 3 which will present a detailed analysis of all the data generated. Finally, Section 4 will conclude and offer future avenues for the work presented here.

2 Validation

As described in [19] consider the payoff matrix:

$$M = \begin{pmatrix} a, b \\ c, d \end{pmatrix} \tag{1}$$

The expected payoffs of i players of the first type in a population with $N - i$ players of the second type are given by:

$$F_i = \frac{a(i-1) + b(N-i)}{N-1} \tag{2}$$

$$G_i = \frac{ci + d(N-i-1)}{N-1} \tag{3}$$

With an intensity of selection ω the fitness of both strategies is given by:

$$f_i = 1 - \omega + \omega F_i \tag{4}$$

$$g_i = 1 - \omega + \omega G_i \tag{5}$$

The transitions within the birth death process that underpins the Moran process are then given by:

$$p_{i,i+1} = \frac{if_i}{if_i + (N-i)g_i} \frac{N-i}{N} \tag{6}$$

$$p_{i,i-1} = \frac{(N-i)g_i}{if_i + (N-i)g_i} \frac{i}{N} \tag{7}$$

$$p_{ii} = 1 - p_{i,i+1} - p_{i,i-1} \tag{8}$$

Using this it is a known result [3] that the fixation probability of the first strategy in a population of i individuals of the first type (and $N - i$ individuals of the second::

$$x_i = \frac{1 + \sum_{j=1}^{i-1} \prod_{k=1}^{j} \gamma_j}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^{j} \gamma_j} \tag{9}$$

where:

$$\gamma_j = \frac{p_{j,j-1}}{p_{j,j+1}}$$

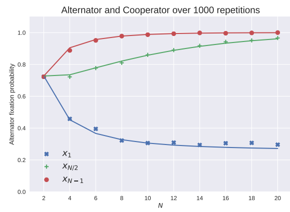
A neutral strategy will have fixation probability $x_i = i/N$.

Comparisons of $x_1, x_{N/2}, x_{N-1}$ are shown in Figure 5. The points represent the simulated values and the line shows the theoretical value. Note that these are all deterministic strategies and show a perfect match up between the expected value of (9) and the actual Moran process for all strategies pairs.

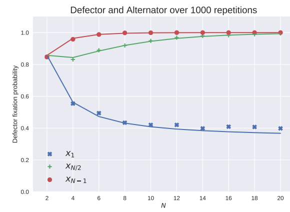
Figure 6 shows the fixation probabilities for stochastic strategies. These are no longer a good match which highlights the weakness of the analytical formulae that relies on the average payoffs (1).

All data generated for this validation exercise can be found at [11].

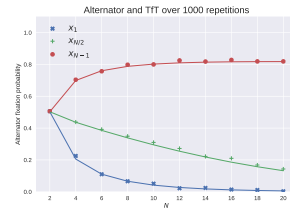
A detailed analysis of the 164 strategies considered, using direct Moran processes will be shown in the next Section.



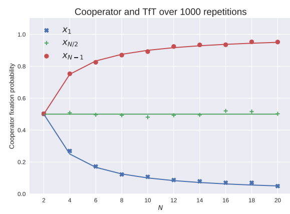
(a) Alternator and Cooperator



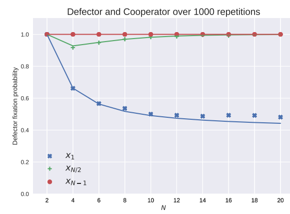
(b) Defector and Alternator



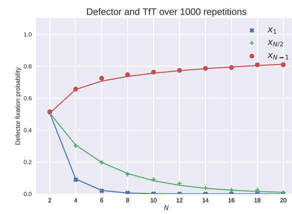
(c) Alternator and Tit For Tat



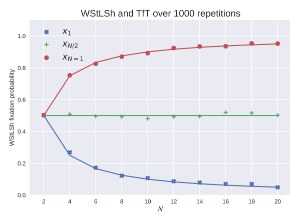
(d) Cooperator and Tit For Tat



(e) Defector and Cooperator



(f) Defector and Tit For Tat



(g) Win Stay Lose Shift and Tit For Tat

(h) Alternator and Win Stay Lose Shift

(i) Defector and Win Stay Lose Shift

Figure 5: Comparison of theoretic and actual Moran Process fixation probabilities for **deterministic** strategies

3 Empirical results

This section outlines the data analysis carried out:

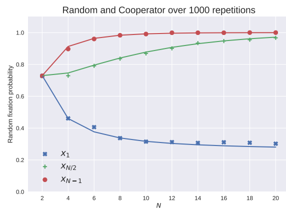
- Section 3.1 considers the specific case of $N = 2$.
- Section 3.2 investigates the effect of population size on the ability of a strategy to invade another population. This will highlight how complex strategies with long memories outperform simpler strategies.
- Section 3.3 similarly investigates the ability to defend against an invasion.
- Section 3.4 investigates the relationship between performance for differing population sizes as well as taking a close look at Zero determinant strategies [20].

3.1 The special case of $N = 2$

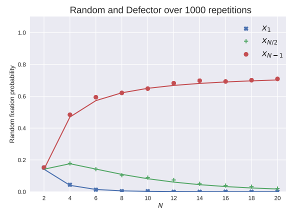
When $N = 2$ the Moran process is effectively a measure of the relative mean payoffs over all possible matches between two players. The strategy that scores higher than the other more often will fixate more often.

For $N = 2$ the two cases of x_1 and x_{N-1} coincide, but will be considered separately for larger N in sections 3.2 and 3.3. Figure 7 shows all fixation probabilities for the strategies considered. This is summarised in Table 2.

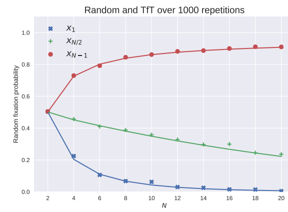
1. The top strategy is the Collective Strategy (CS) which has a simple handshake mechanism (a cooperation followed by a defection on the first move). As long as the opponent plays the same handshake and does not defect in the future it cooperates. Otherwise it defects for all rounds. This strategy was specifically designed for evolutionary processes [14].
2. The Defector: it always defects. As it has little potential interaction with itself (recall that $N = 2$), its aggressiveness is rewarded.



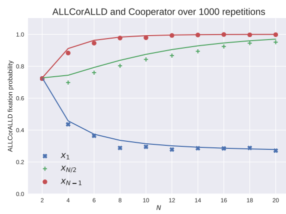
(a) Random and Cooperator



(b) Random and Defector

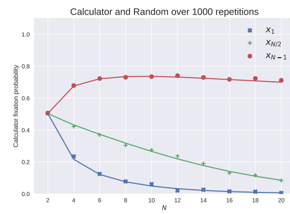


(c) Random and Tit For Tat

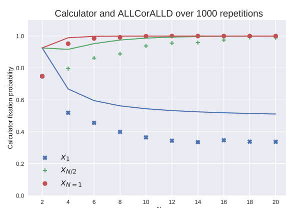


(d) All C or all D and Cooperator

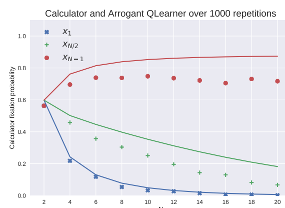
(e) All C or all D and Defector



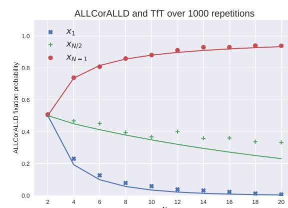
(f) Calculator and Random



(g) Calculator and All C or all D



(h) Calculator and Arrogant Q learner



(i) All C or all D and Tit For Tat

Figure 6: Comparison of theoretic and actual Moran Process fixation probabilities for **stochastic** strategies

3. The Aggravater strategy which plays like Grudger (responding to any defections with unconditional defections throughout) however starts by playing 3 defections.

4. Predator, a finite state machine described in [4].

5. Handshake: a slightly less aggressive version of the Collective strategy [22]. As long as the initial sequence is played then it cooperates. Thus it will do well in a population consisting of many members of itself: just as the Collective strategy does. However it is not aggressive enough to invade other populations (as can be seen in Section 3.2.

Player	Mean p_1	Memory Depth	Stochastic
CS	0.665141	∞	False
Defector	0.649638	0	False
Aggravater	0.632773	∞	False
Predator	0.630129	9	False
Handshake	0.623982	∞	False

Table 2: Summary of top five strategies for $N = 2$

As will be demonstrated in Section 3.4 the results for $N = 2$ differ from those of larger N. Hence these results do not concur with the literature which suggests that Zero Determinant strategies should be effective for larger population sizes, but these analysis consider stationary behaviour, while this work runs for a fixed number of rounds. Note that the stationarity assumptions allows for the analysis to take place leading to the conclusions about zero determinant strategies however it is dependent on a specific structure of strategies. The analysis carried out here makes no assumptions about the structure of the strategies by using actual simulation interactions.

In the next sections close attention to strategies who are strong invaders/resistors is given.

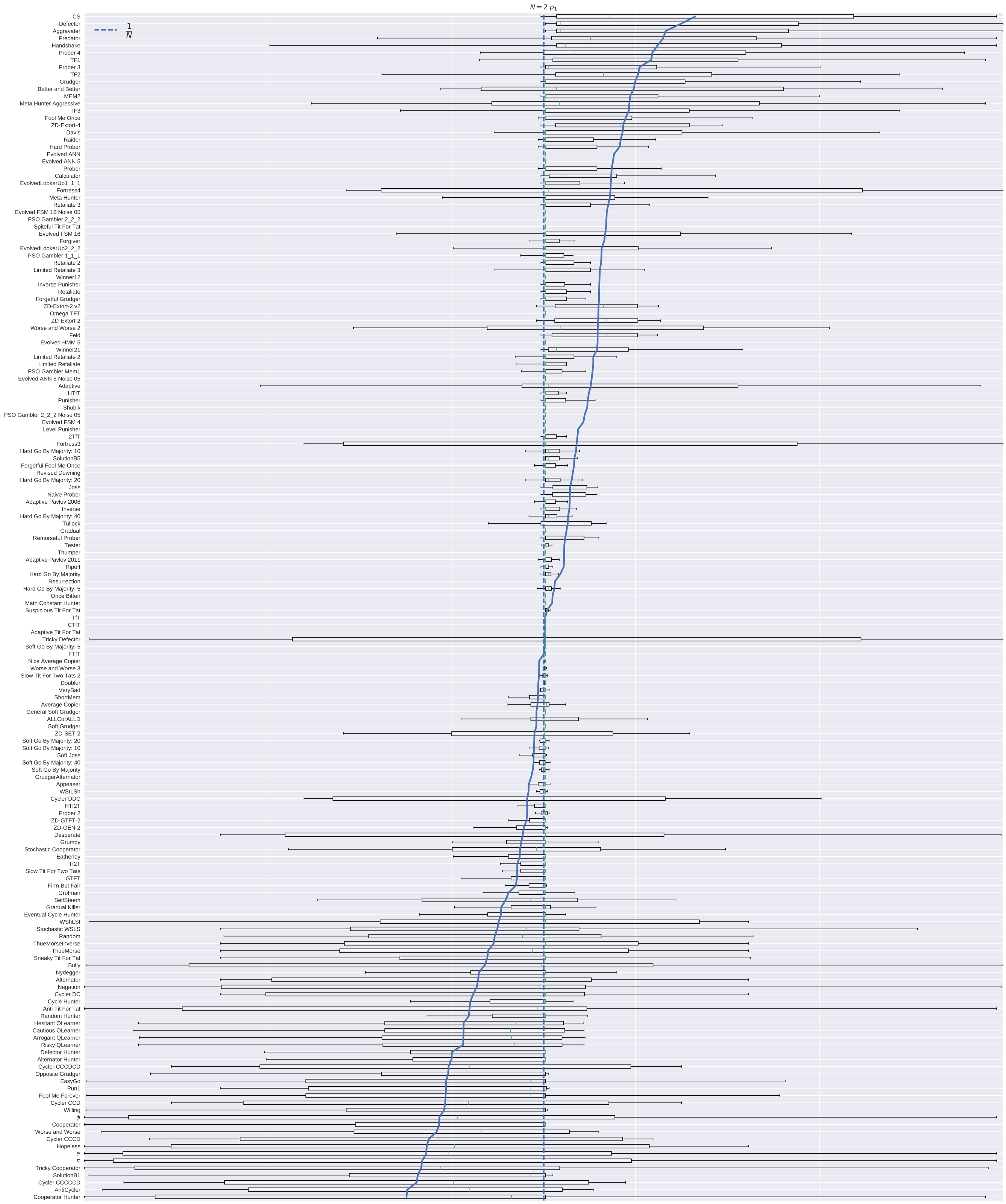


Figure 7: The fixation probabilities for $N = 2$

3.2 Strong invaders

In this section the focus is on the ability of a mutant strategy to invade: the probability of 1 individual of a given type successfully becoming fixated in a population of $N-1$ other individuals, denoted by x_1. The fixation probabilities are shown in Figures 8, 9 and 10 for $N \in \{3, 7, 14\}$ showing the mean fixation as well as the neutral fixation for each given scenario.

The top five strategies are given in Tables 3.

Player	Mean p_1	Memory Depth	Stochastic
CS	0.447761	∞	False
Grudger	0.431264	∞	False
MEM2	0.427804	∞	False
TF3	0.426736	16	False
Prober 4	0.424215	∞	False

(a) $N = 3$

Player	Mean p_1	Memory Depth	Stochastic
Evolved FSM 16	0.252282	16	False
PSO Gambler 2_2_2	0.246742	∞	True
Fool Me Once	0.245871	∞	False
Evolved ANN 5	0.244982	∞	False
Evolved ANN	0.244933	∞	False

(b) $N = 7$

Player	Mean p_1	Memory Depth	Stochastic
Evolved FSM 16	0.209564	16	False
PSO Gambler 2_2_2	0.204215	∞	True
EvolvedLookerUp2_2_2	0.201411	∞	False
Evolved ANN	0.201387	∞	False
Evolved ANN 5	0.200387	∞	False

(c) $N = 14$

Table 3: Properties of top five invaders

It can be seen that apart from CS, none of the strategies of Table 2 perform well for $N \in \{3, 7, 14\}$. The new high performing strategies are:

- Grudger (which only performs well for $N = 3$), starts by cooperating but will defect if at any point the opponent has defected.
- MEM2, an infinite memory strategy that switches between TfT, Tf2T, and Defector [15].
- TF3, the finite state machine trained specifically for Moran processes described in Section 1.
- Prober 4, complex strategy with an initial 20 move sequence of cooperations and defections [16]. This initial sequence serves as approximate handshake.
- PSO Gambler and Evolved Lookerup 2 2 2: are strategies that make use of a lookup table mapping the first 2 moves of the opponent as well as the last 2 moves of both players to an action. The PSO gambler is a stochastic version which maps those states to probabilities of cooperating. The lookerup was described in [12].
- The evolved ANN strategies are neural networks that map a number of attributes (first move, number of cooperations, last move etc...) to an action. Both of these have been trained using an evolutionary algorithm and the ANN 5 was trained to perform well in a noisy tournament.
- The Evolved FSM 16 is a 16 state finite state machine trained to perform well in tournaments.

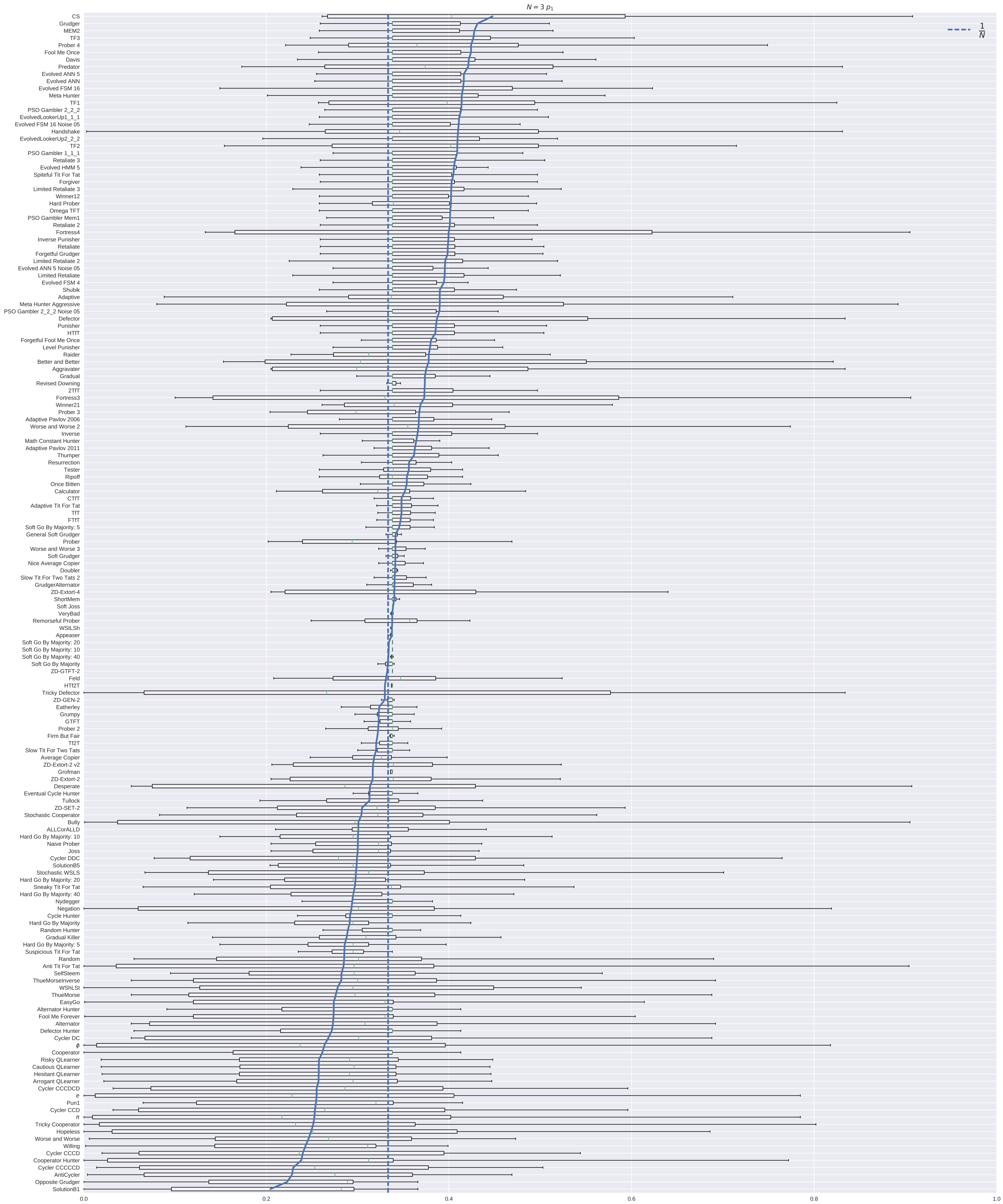


Figure 8: The fixation probabilities x_1 for $N = 3$

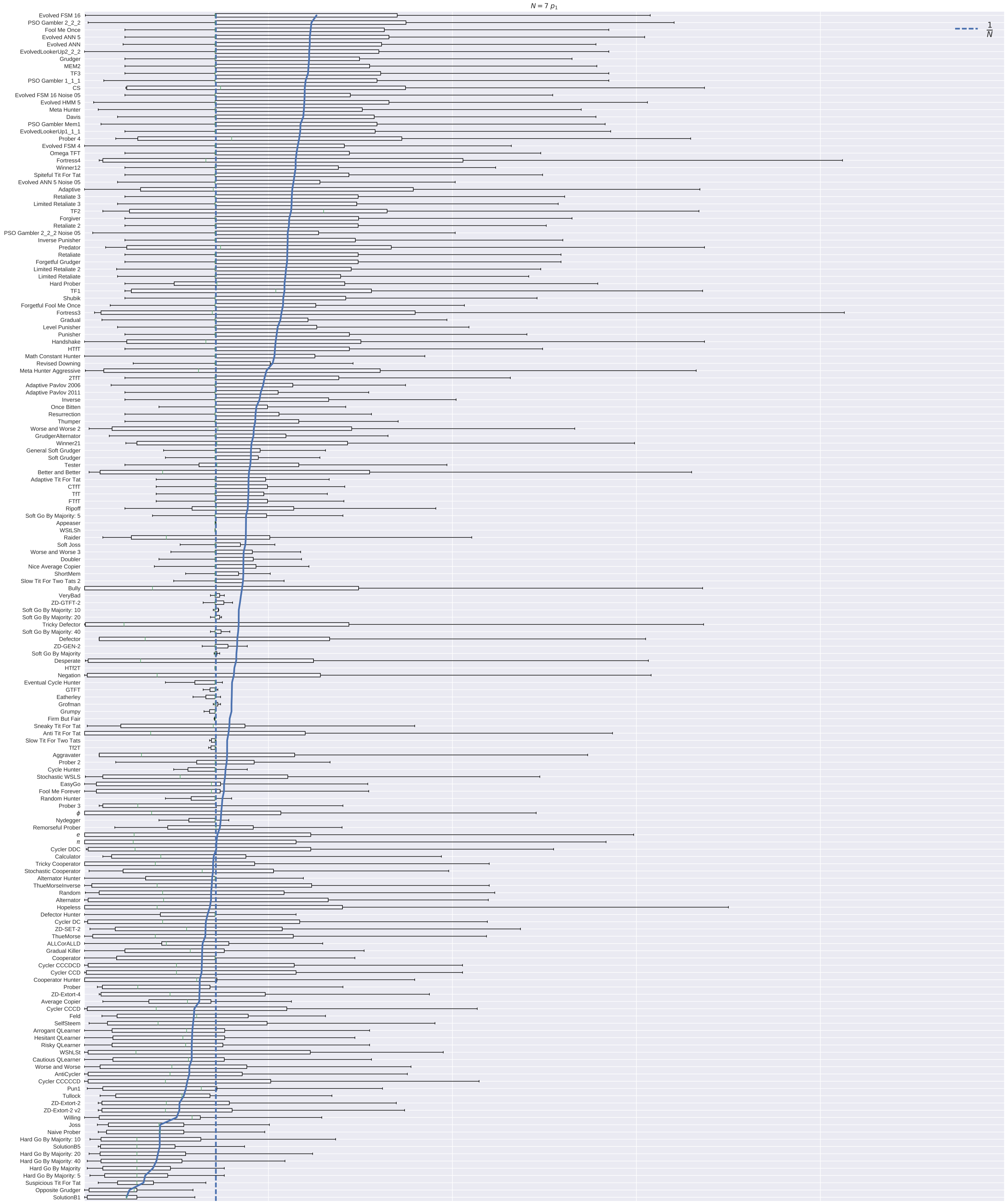


Figure 9: The fixation probabilities x_1 for $N = 7$

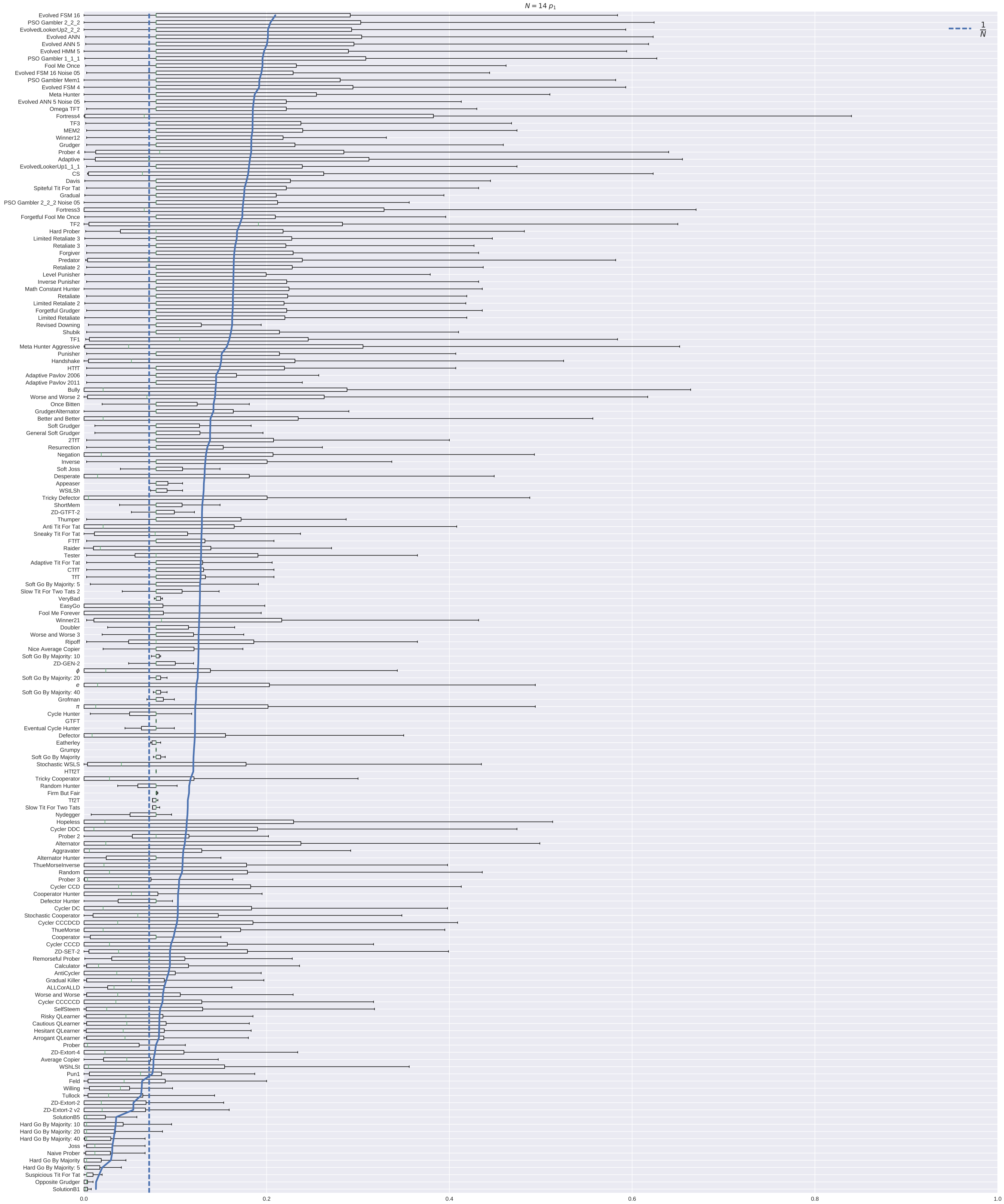


Figure 10: The fixation probabilities x_1 for $N = 14$

As well as noting that the memory length and complexity of these strategies are much greater than one, it is interesting to note that none of them are akin to memory one strategies. Only one is stochastic although close inspection of the source code of PSO Gambler shows that it makes stochastic decisions with extreme probabilities: almost always acting as expected. Apart from TF3 in $N = 3$, the finite state machines trained specifically for Moran processes do not appear in the top 5: whereas strategies trained for tournament do. This is due to the nature of invasion: most of the opponents will initially be different strategies. The next section will consider the converse situation.

3.3 Strong resistors

In addition to identifying good invaders, strategies resistant to invasion by other strategies are identified by examining the distribution of x_{N-1} for each strategy. Note that this is equivalent to looking at x_1 for all opponents.

The fixation probabilities are shown in Figures 11, 9 and 13 for $N \in \{3, 7, 14\}$ showing the mean fixation as well as the neutral fixation for each given scenario.

Table 4 shows the top five strategies when ranked according to x_{N-1} for $N \in \{3, 7, 14\}$. Once again none of the short memory strategies from Section 3.1 perform well for high N.

Player	Mean p_{N-1}	Memory Depth	Stochastic
CS	0.835859	∞	False
Predator	0.812129	9	False
TF1	0.808736	∞	False
Handshake	0.801356	∞	False
TF2	0.795736	∞	False

(a) $N = 3$

Player	Mean p_{N-1}	Memory Depth	Stochastic
CS	0.976491	∞	False
TF1	0.971405	∞	False
TF2	0.967712	∞	False
Predator	0.967687	9	False
Handshake	0.954650	∞	False

(b) $N = 7$

Player	Mean p_{N-1}	Memory Depth	Stochastic
CS	0.998442	∞	False
TF1	0.997319	∞	False
TF2	0.994865	∞	False
Predator	0.994074	9	False
Prober 4	0.986301	∞	False

(c) $N = 14$

Table 4: Properties of top five resistors

Interestingly none of these strategies are stochastic: this is explained by the need of strategies to have a steady hand when interacting with their own kind. In essence: acting stochastically increase the chance of friendly fire. However it is possible to design a strategy with a “stochastic handshake” [13].

There are are only two new strategies that appear in the top ranks for x_{N-1}: TF1 and TF2. These two strategies are with CS the strongest resistors. They all have handshakes, and whilst the handshake that CS and Handshake (which ranks highly for the smaller values of N) have was programmed, the handshakes of TF1 and TF2 evolved through an evolutionary process.

As described in Section 3.2 the strategies trained with the payoff maximizing objective are among the best invaders in the library however they are not as resistant to invasion as the strategies trained using a Moran objective function. These strategies include trained finite state machine strategies, but they do not appear to have handshaking mechanisms. Therefore it is reasonable to conclude that the objective function is the cause of the emergence of handshaking mechanisms.

The payoff maximizing strategies typically will not defect before the opponent’s first defection, possibly because the training strategy collection contains a significant portion of strategies such as Grudger and Fool Me Once that retaliate

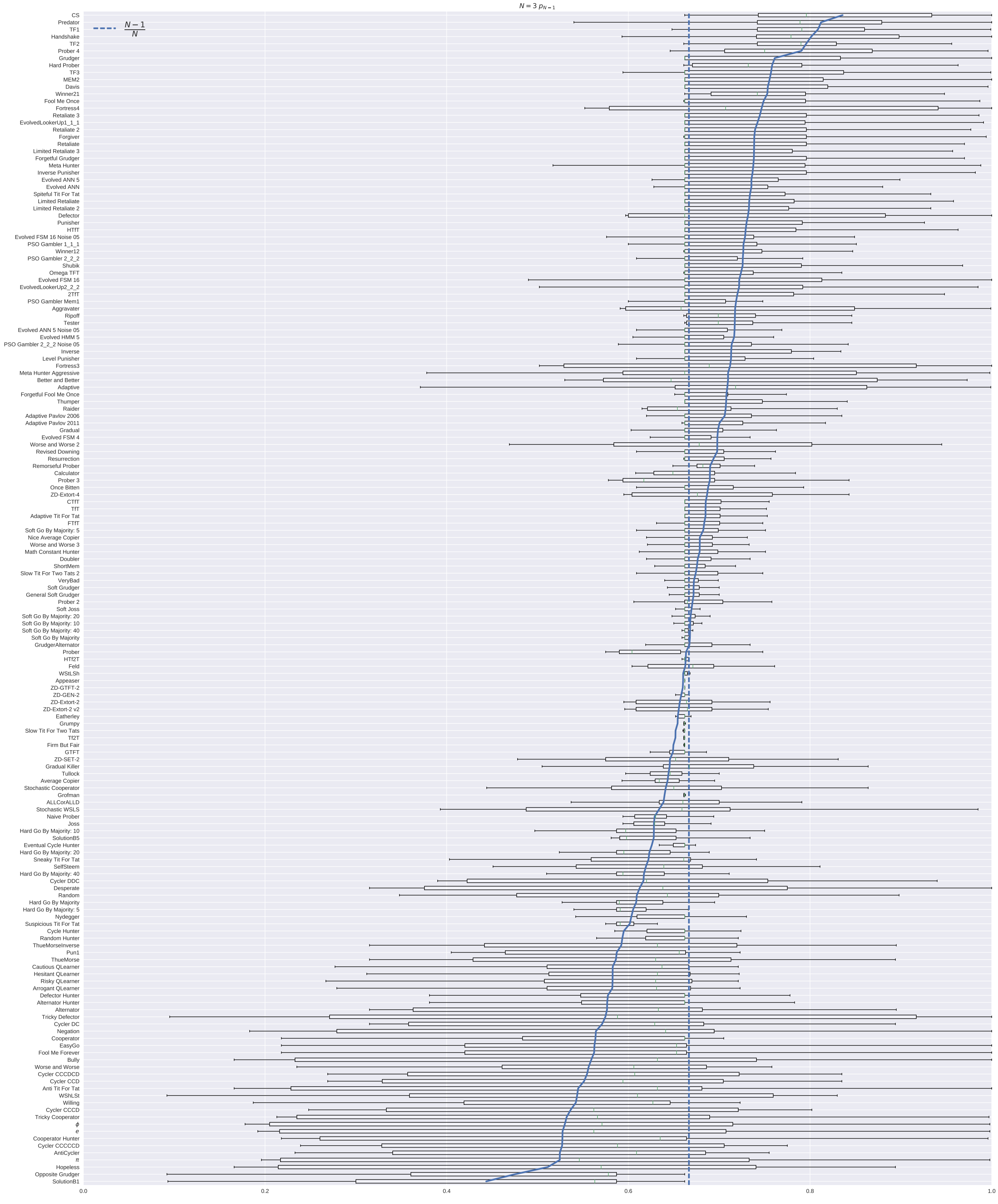


Figure 11: The fixation probability x_{N-1} for $N = 3$

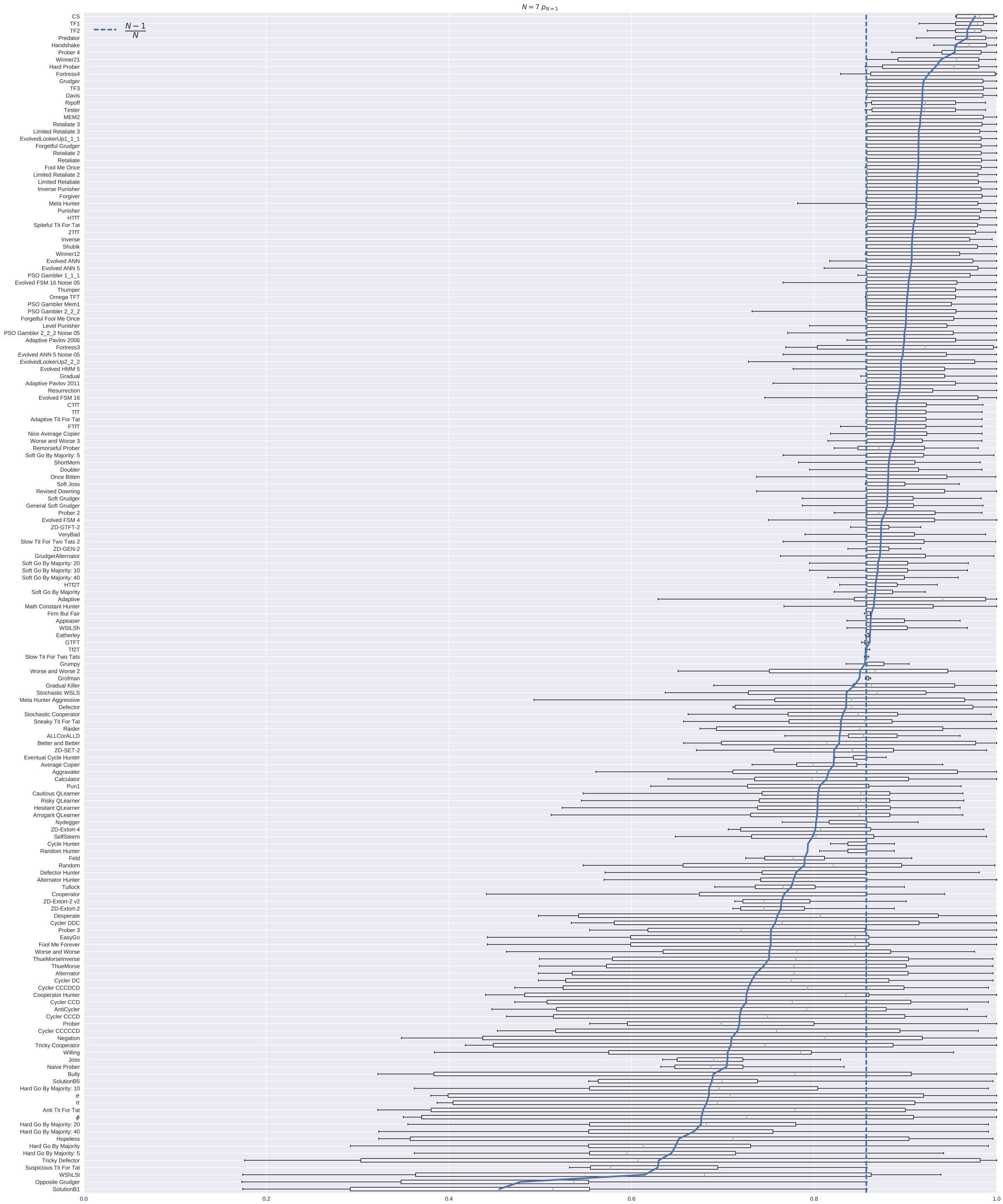


Figure 12: The fixation probability x_{N-1} for $N = 7$

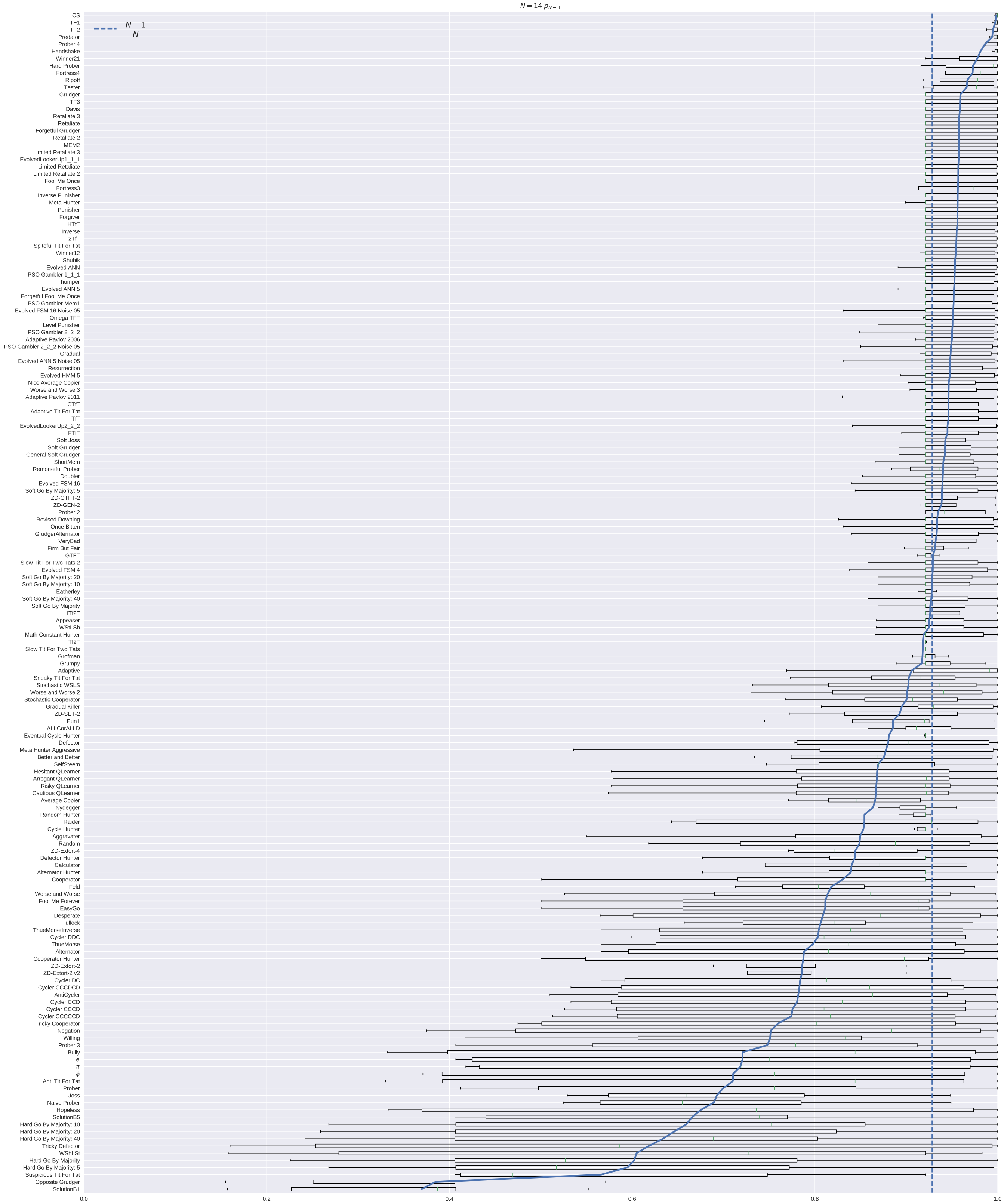


Figure 13: The fixation probability x_{N-1} for $N = 14$

harshly by defecting for the remainder of the match if the opponent has more than a small number of cumulative defections. Paradoxically it is necessary to defect (as a signal) in order to achieve mutual cooperation with opponents using the same strategy but not with other opponents.

A handshake requires at least one defection and there is selective pressure to defect as few times as possible to achieve the self-recognition mechanism. It is also unwise to defect on the first move as some strategies additionally retaliate first round defections. So the handshakes used by TF1 and TF2, and CS, are in some sense optimal. These discoveries may have significant ramifications regarding the evolution of cooperation and forgiveness in biological organisms such as antibacterial resistant bacteria and social interactions between humans.

It is evident through Sections 3.1, 3.2 and 3.3 that performance of strategies not only depends on the initial population distribution but also that there seems to be a difference depending on whether or not $N > 2$. This will be explored further in the next section, looking not only at x_1 and x_{N-1} but also consider $x_{N/2}$.

3.4 The effect of population size

Figures 14, 15 and 16 show the rank of each strategy based on mean fixation probabilities against population size. Tables 5, 6 and 7 show the same information for a selection of strategies:

- The strategies that ranked highly for $N = 2$;
- The strategies that ranked highly for $N = 14$;
- The Zero determinant strategies.

For all starting populations $i \in \{1, N/2, N-1\}$ the ranks of strategies are relatively stable across the different values of $N > 2$ however for $N = 2$ there is a distinct difference. This highlights that there is little that can be inferred about the evolutionary performance of a strategy in a large population from its performance in a small population.

This is confirmed by the performance of the Zero determinant strategies, whilst some do rank relatively highly for $N = 2$ (ZD-extort-4 has tank 16) this rank does not translate to larger populations.

Player	2	3	4	5	6	7	8	9	10	11	12	13	14
CS	1.0	1.0	2.0	11.0	9.0	11.0	13.0	21.0	16.0	22.0	17.0	25.0	23.0
Defector	2.0	43.0	80.0	91.0	89.0	87.0	87.0	103.0	97.0	105.0	94.0	103.0	101.0
Aggravater	3.0	50.0	89.0	99.0	102.0	103.0	108.0	113.0	114.0	115.0	115.0	116.0	117.0
Predator	4.0	8.0	24.0	35.0	28.0	33.0	31.0	43.0	36.0	43.0	34.0	45.0	35.0
Handshake	5.0	17.0	40.0	46.0	43.0	46.0	46.0	49.0	48.0	49.0	47.0	50.0	49.0
Evolved FSM 16	31.0	11.0	6.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
PSO Gambler 2_2_2	29.0	14.0	10.0	6.0	4.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
EvolvedLookerUp2_2_2	33.0	18.0	11.0	9.0	10.0	6.0	6.0	5.0	3.0	5.0	3.0	3.0	3.0
Evolved ANN	20.0	10.0	8.0	7.0	8.0	5.0	3.0	3.0	4.0	3.0	4.0	4.0	4.0
Evolved ANN 5	21.0	9.0	7.0	8.0	7.0	4.0	5.0	4.0	5.0	4.0	5.0	5.0	5.0
ZD-Extort-4	16.0	81.0	107.0	120.0	135.0	136.0	142.0	140.0	142.0	142.0	144.0	144.0	145.0
ZD-Extort-2 v2	41.0	105.0	126.0	140.0	152.0	152.0	153.0	152.0	153.0	153.0	153.0	152.0	153.0
ZD-Extort-2	43.0	107.0	125.0	139.0	151.0	151.0	152.0	153.0	152.0	152.0	152.0	153.0	152.0
ZD-SET-2	100.0	111.0	117.0	117.0	122.0	127.0	131.0	128.0	131.0	131.0	130.0	132.0	131.0
ZD-GTFT-2	112.0	92.0	82.0	80.0	81.0	82.0	84.0	72.0	81.0	71.0	78.0	72.0	70.0
ZD-GEN-2	113.0	96.0	87.0	83.0	85.0	88.0	90.0	82.0	87.0	82.0	86.0	83.0	91.0

Table 5: Ranks of some strategies according to x_1 for different population sizes

Figure 17 show the correlation coefficients of the ranks of strategies in differing population size. It is immediate to note that how well a strategy performs in any Moran process for $N > 2$ has little to do with the performance for $N = 2$. This illustrates why the strong performance of Zero determinant strategies predicted in [20] does not extend to larger populations. This was discussed theoretically in [1] however not observed empirically at the scale presented here.

4 Conclusion

A detailed empirical analysis of 164 strategies of the IPD within a pairwise Moran process has been carried out. All $\binom{164}{2} = 13,366$ possible ordered pairs of strategies have been placed in a Moran process with different starting values allowing the each strategy to attempt to invade the other. This is the largest such experiment carried out and has lead to many insights.

When studying evolutionary processes it is vital to consider $N > 2$ as the special case for $N = 2$ cannot be used to extrapolate performance in bigger populations. This was shown both observationally in Sections 3.2 and 3.3 but also by considering the correlation of the ranks in different population sizes in Section 3.4.

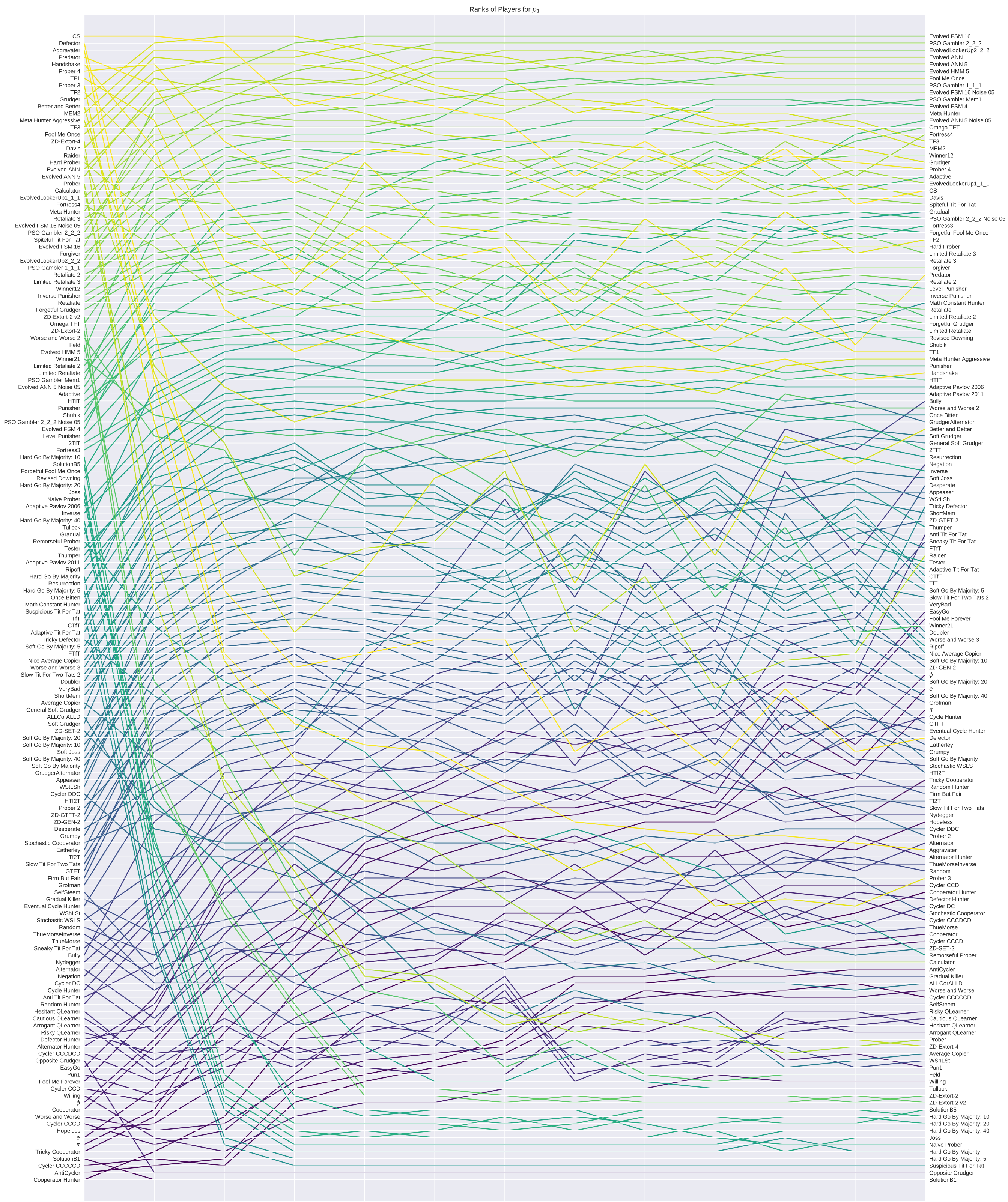


Figure 14: Ranks of all strategies according to x_1 for different population sizes

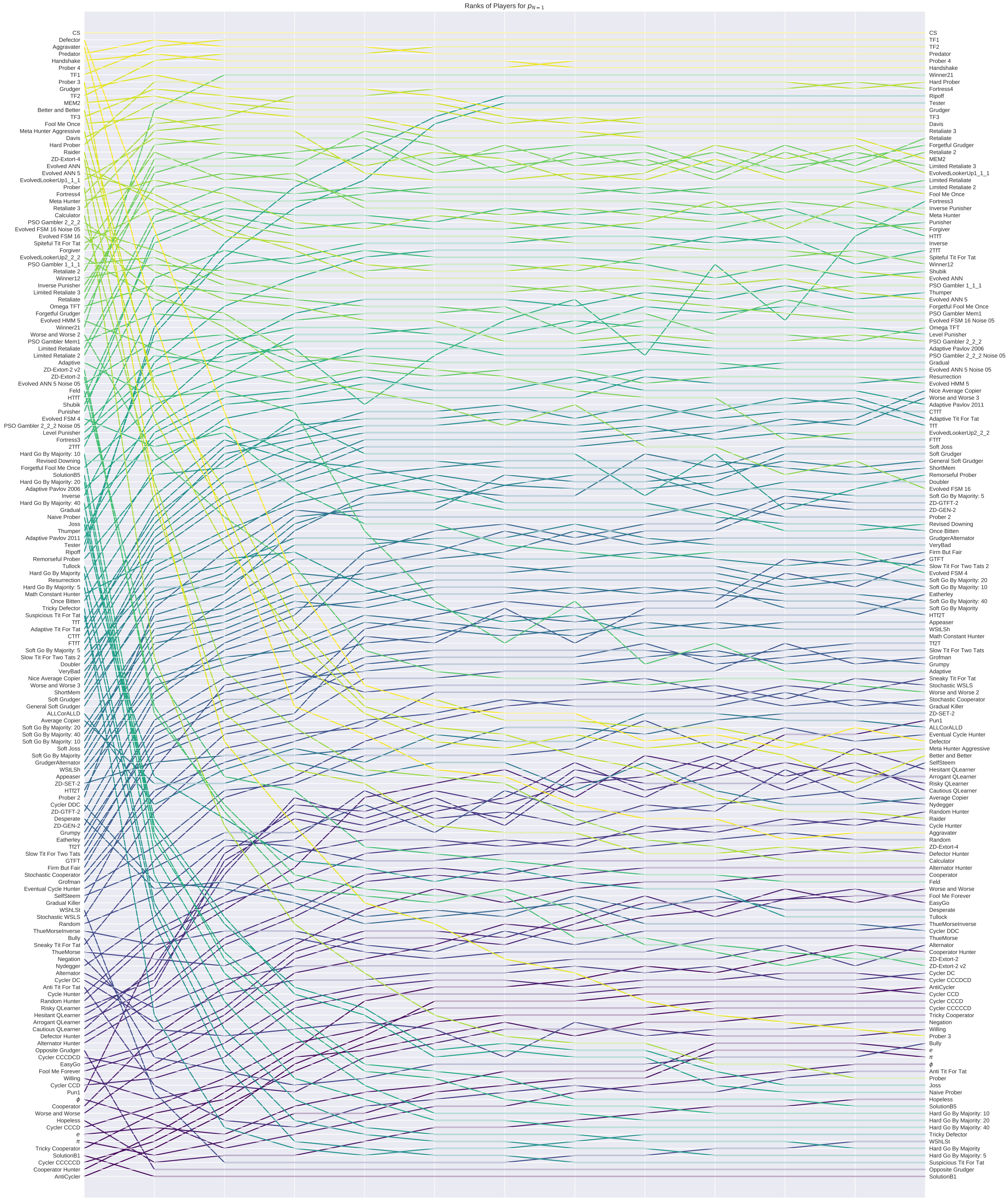


Figure 15: Ranks of all strategies according to x_{N-1} for different population sizes

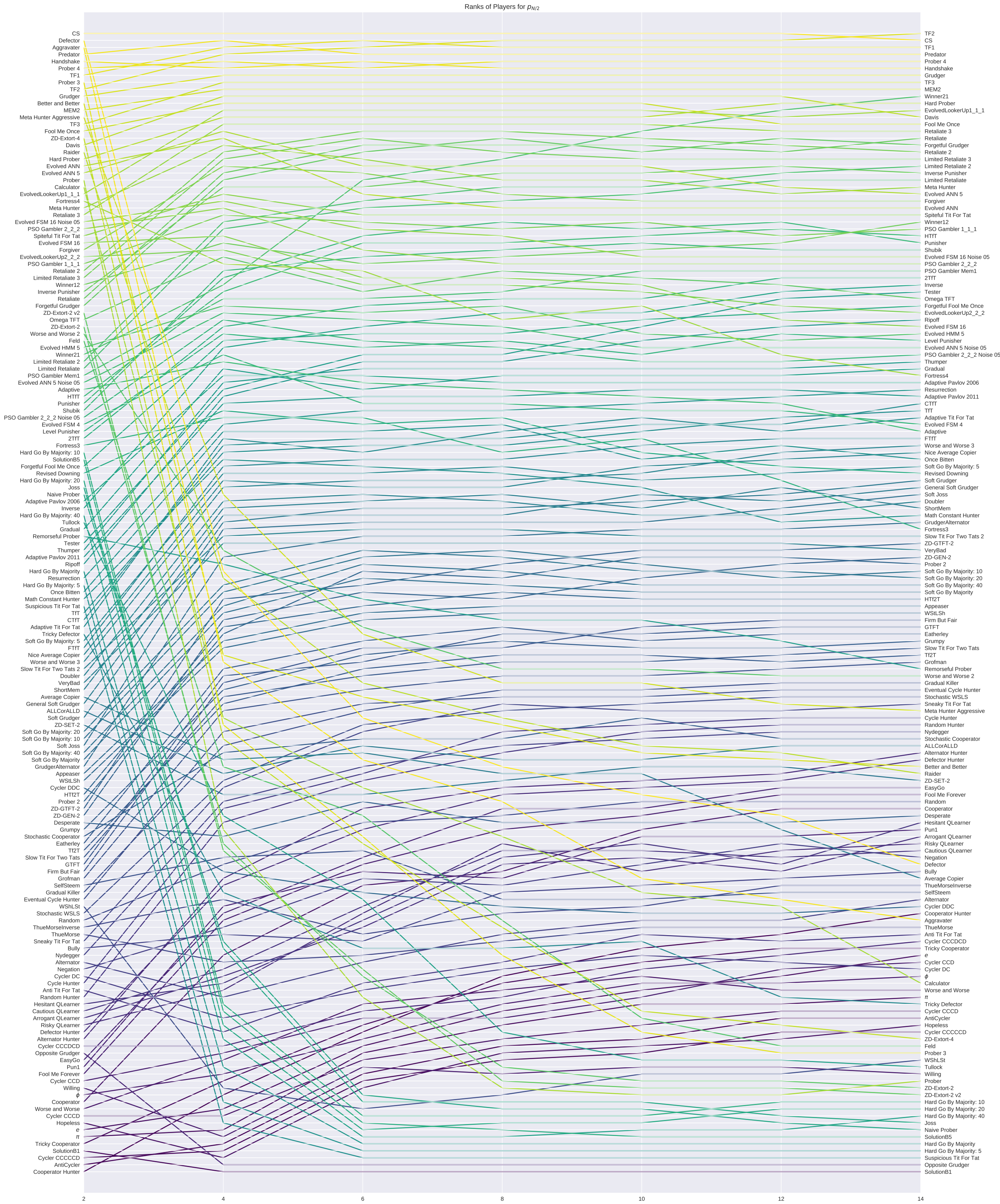


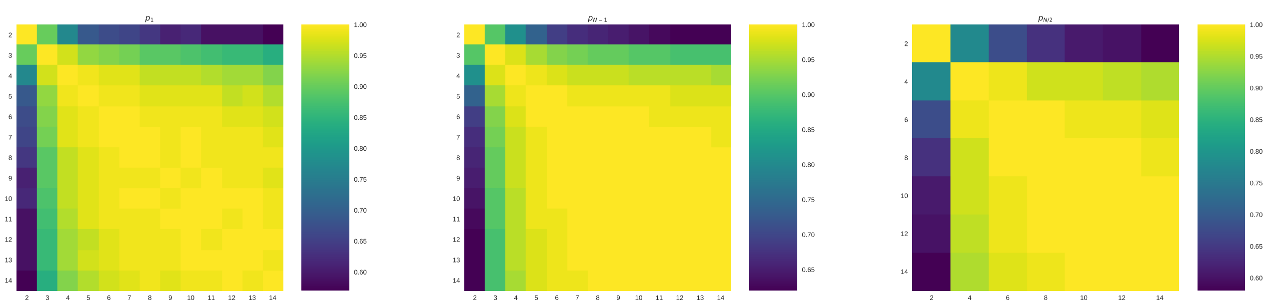
Figure 16: Ranks of all strategies according to $x_{N/2}$ for different population sizes

Player	2	3	4	5	6	7	8	9	10	11	12	13	14
CS	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Defector	2.0	29.0	55.0	79.0	94.0	97.0	98.0	98.0	102.0	101.0	103.0	100.0	102.0
Aggravater	3.0	42.0	71.0	97.0	101.0	106.0	107.0	111.0	113.0	113.0	116.0	115.0	115.0
Predator	4.0	2.0	3.0	3.0	3.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Handshake	5.0	4.0	5.0	5.0	5.0	5.0	5.0	6.0	6.0	6.0	6.0	6.0	6.0
TF1	7.0	3.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
TF2	10.0	5.0	4.0	4.0	4.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Prober 4	6.0	6.0	6.0	6.0	6.0	6.0	6.0	5.0	5.0	5.0	5.0	5.0	5.0
ZD-Extort-4	19.0	68.0	98.0	106.0	108.0	114.0	115.0	115.0	118.0	118.0	117.0	118.0	117.0
ZD-Extort-2 v2	49.0	98.0	111.0	121.0	123.0	124.0	124.0	130.0	130.0	132.0	134.0	132.0	134.0
ZD-Extort-2	50.0	97.0	112.0	123.0	124.0	125.0	123.0	126.0	131.0	131.0	132.0	133.0	133.0
ZD-SET-2	108.0	105.0	104.0	104.0	103.0	103.0	100.0	100.0	101.0	99.0	98.0	98.0	98.0
ZD-GTFT-2	112.0	95.0	88.0	84.0	75.0	72.0	71.0	73.0	71.0	71.0	67.0	68.0	68.0
ZD-GEN-2	114.0	96.0	89.0	86.0	77.0	75.0	72.0	74.0	72.0	72.0	68.0	69.0	69.0

Table 6: Ranks of some strategies according to x_{N-1} for different population sizes

Player	2	4	6	8	10	12	14
CS	1.0	1.0	1.0	1.0	1.0	1.0	2.0
Defector	2.0	78.0	99.0	106.0	110.0	113.0	120.0
Aggravater	3.0	91.0	105.0	111.0	122.0	125.0	128.0
Predator	4.0	2.0	4.0	4.0	4.0	4.0	4.0
Handshake	5.0	6.0	5.0	6.0	6.0	6.0	6.0
TF2	9.0	4.0	3.0	2.0	2.0	2.0	1.0
TF1	7.0	3.0	2.0	3.0	3.0	3.0	3.0
Prober 4	6.0	5.0	6.0	5.0	5.0	5.0	5.0
ZD-Extort-4	16.0	102.0	117.0	129.0	141.0	143.0	145.0
ZD-Extort-2 v2	41.0	118.0	135.0	151.0	152.0	152.0	153.0
ZD-Extort-2	43.0	117.0	136.0	149.0	151.0	151.0	152.0
ZD-SET-2	100.0	110.0	110.0	108.0	106.0	106.0	108.0
ZD-GTFT-2	112.0	82.0	80.0	77.0	75.0	75.0	74.0
ZD-GEN-2	113.0	85.0	81.0	82.0	79.0	77.0	76.0

Table 7: Ranks of some strategies according to $x_{N/2}$ for different population sizes



(a) Rank based on x_1 (b) Rank based on x_{N-1} (c) Rank based on $x_{N/2}$

Figure 17: Heatmap of correlation coefficients of rankings by population size

Memory one strategies do not perform well, as predicted by [20]. There are no memory one strategies in the top 5 performing strategies for $N > 3$. This is due to their lack of sophistication which allows them to recognise and adjust to their opponent. Some very sophisticated strategies proves to be high performers for invasion: these are infinite memory strategies which have been trained using a number of reinforcement learning algorithms. Interestingly they have been trained to perform well in tournaments and not Moran processes which highlights the potentially for improvement.

One of the major findings discussed in Section 3.3, is the ability of strategies with a handshake mechanism to resist invasion. This was not only identify for CS (an intelligently designed strategy) but also for two FSM strategies (TF1 and TF2) specifically trained through an evolutionary process. In these two cases, the handshake mechanism was a product of the evolutionary process. This has the potential to help with the understanding of organisms with a strong resistance to invasion such as anti antibacterial resistant bacteria [7]. With the knowledge that a handshake being likely to exist, perhaps it can be mimicked.

It is felt that these findings are important for the ongoing understanding of population dynamics and offer evidence for some of the shortcomings of short memory which has started to be recognised by the community [9].

All source code for this work has been written in a sustainable manner: it is open source, under version control and tested which ensures that all results can be reproduced [21, 23, 28]. The raw data as well as the processed data has also been properly archived and can be fond at [11].

There are various areas for further work to build on this. Firstly, an analysis of the effect of noise would offer insights about the stability of the findings. It would also be possible to consider three or more types of strategy in the population and finally mutation would also offer an interesting dimension to explore.

Acknowledgements

This work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University.

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

- The axelrod library (IPD strategies and Moran processes) [25].
- The matplotlib library (visualisation) [10].
- The pandas and numpy libraries (data manipulation) [17, 26].

References

[1] Christoph Adami and Arend Hintze. "Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything." In: *Nature communications* 4.1 (2013), p. 2193. ISSN: 2041-1723. DOI: 10.1038/ncomms3193. arXiv: arXiv:1208.2666v4. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3741637%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract.

[2] Michael Affenzeller et al. *Genetic algorithms and genetic programming: modern concepts and practical applications.* Crc Press, 2009.

[3] B. Allen et al. "Evolutionary dynamics on any population structure". In: 544 (Mar. 2017), pp. 227–230. DOI: 10.1038/nature21723. arXiv: 1605.06530 [q-bio.PE].

[4] Wendy Ashlock and Daniel Ashlock. "Changes in Prisoner ' s Dilemma Strategies Over Evolutionary Time With Different Population Sizes". In: (2006), pp. 1001–1008.

[5] R. Axelrod. "Effective Choice in the Prisoner's Dilemma". In: *Journal of Conflict Resolution* 24.1 (1980), pp. 3–25.

[6] Seung Ki Baek et al. "Comparing reactive and memory- one strategies of direct reciprocity". In: *Nature Publishing Group* (2016), pp. 1–13. DOI: 10.1038/srep25676. URL: http://dx.doi.org/10.1038/srep25676.

[7] Julian Davies and Dorothy Davies. "Origins and Evolution of Antibiotic Resistance". In: *Microbiol. Mol. Biol. Rev.* 74.3 (2010), pp. 417–433. ISSN: 1098-5557. DOI: 10.1128/mmbr.00016-10. URL: http://mmbr.asm.org/cgi/content/abstract/74/3/417.

[8] Merrill M. Flood. *Some Experimental Games.* 1958. DOI: 10.1287/mnsc.5.1.5.

[9] Christian Hilbe et al. "Memory- ¡i¿n¡/i¿ strategies of direct reciprocity". In: *Proceedings of the National Academy of Sciences* (2017), p. 201621239. ISSN: 0027-8424. DOI: 10.1073/pnas.1621239114. URL: http://www.pnas.org/lookup/doi/10.1073/pnas.1621239114.

[10] John D Hunter. "Matplotlib: A 2D graphics environment". In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95.

[11] Knight, Vincent and Harper, Marc and Glynatsi E., Nikoleta. *Data: Empirical Study of Invasion and Resistance for Iterated Prisoner's Dilemma Strategies*. May 2017. DOI: ?. URL: ?.

[12] Vincent Knight et al. "An Open Framework for the Reproducible Study of the Iterated Prisoner ' s Dilemma". In: (2016).

[13] Christopher Lee, Marc Harper, and Dashiell Fryer. "The Art of War: Beyond Memory-one Strategies in Population Games". In: *Plos One* 10.3 (2015), e0120625. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0120625. URL: http://dx.plos.org/10.1371/journal.pone.0120625.

[14] Jiawei Li and Graham Kendall. "A strategy with novel evolutionary features for the iterated prisoner's dilemma." In: *Evolutionary Computation* 17.2 (2009), pp. 257–274. ISSN: 1063-6560. DOI: 10.1162/evco.2009.17.2.257. URL: http://www.ncbi.nlm.nih.gov/pubmed/19413490.

[15] Jiawei Li, Graham Kendall, and Senior Member. "The effect of memory size on the evolutionary stability of strategies in iterated prisoner ' s dilemma". In: X.X (2014), pp. 1–8.

[16] LIFL. *PRISON*. 2008. URL: http://www.lifl.fr/IPD/ipd.frame.html.

[17] Wes McKinney et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. van der Voort S, Millman J. 2010, pp. 51–56.

[18] P.A.P. Moran. "Random Processes in Genetics". In: April (1957), pp. 60–71.

[19] Martin A Nowak. *Evolutionary Dynamics: Exploring the Equations of Life.* Cambridge: Harvard University Press. ISBN: 0674023382. DOI: 10.1086/523139.

[20] William H Press and Freeman J Dyson. "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent." In: *Proceedings of the National Academy of Sciences of the United States of America* 109.26 (2012), pp. 10409–13. ISSN: 1091-6490. DOI: 10.1073/pnas.1206569109. URL: http://www.pnas.org/content/109/26/10409.abstract.

[21] Andreas Prli and James B. Procter. "Ten Simple Rules for the Open Development of Scientific Software". In: *PLoS Computational Biology* 8.12 (2012), e1002802. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002802. URL: http://dx.plos.org/10.1371/journal.pcbi.1002802.

[22] Arthur Robson. *EFFICIENCY IN EVOLUTIONARY GAMES: DARWIN, NASH AND SECRET HANDSHAKE*. Working Papers. Michigan - Center for Research on Economic & Social Theory, 1989. URL: http://EconPapers.repec.org/RePEc:fth:michet:89-22.

[23] Geir Kjetil Sandve et al. "Ten Simple Rules for Reproducible Computational Research". In: *PLoS Computational Biology* 9.10 (2013), pp. 1–4. ISSN: 1553734X. DOI: 10.1371/journal.pcbi.1003285.

[24] Alexander J. Stewart and Joshua B. Plotkin. "Extortion and cooperation in the Prisoners Dilemma". In: *Proceedings of the National Academy of Sciences* 109.26 (2012), pp. 10134–10135. DOI: 10.1073/pnas.1208087109. eprint: http://www.pnas.org/content/109/26/10134.full.pdf. URL: http://www.pnas.org/content/109/26/10134.short.

[25] The Axelrod project developers. *Axelrod: v2.9.0*. Apr. 2016. DOI: 499122. URL: http://dx.doi.org/10.5281/zenodo.499122.

[26] Stfan van der Walt, S Chris Colbert, and Gael Varoquaux. "The NumPy array: a structure for efficient numerical computation". In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[27] Jeffrey West et al. "The prisoners dilemma as a cancer model". In: *Convergent Science Physical Oncology* 2.3 (2016), p. 035002. URL: http://stacks.iop.org/2057-1739/2/i=3/a=035002.

[28] Greg Wilson et al. "Best Practices for Scientific Computing". In: 12.1 (2014). DOI: 10.1371/journal.pbio.1001745.

A List of players

1. ϕ
2. π
3. e
4. ALLCorALLD
5. Adaptive
6. Adaptive Pavlov 2006
7. Adaptive Pavlov 2011
8. Adaptive Tit For Tat: 0.5
9. Aggravater

10. Alternator
11. Alternator Hunter
12. Anti Tit For Tat
13. AntiCycler
14. Appeaser
15. Arrogant QLearner
16. Average Copier
17. Better and Better
18. Bully
19. Calculator
20. Cautious QLearner
21. CollectiveStrategy (**CS**)
22. Contrite Tit For Tat (**CTfT**)
23. Cooperator
24. Cooperator Hunter
25. Cycle Hunter
26. Cycler CCCCCD
27. Cycler CCCD
28. Cycler CCCDCD
29. Cycler CCD
30. Cycler DC
31. Cycler DDC
32. Davis: 10
33. Defector
34. Defector Hunter
35. Desperate
36. Doubler
37. EasyGo
38. Eatherley
39. Eventual Cycle Hunter
40. Evolved ANN
41. Evolved ANN 5
42. Evolved ANN 5 Noise 05
43. Evolved FSM 16
44. Evolved FSM 16 Noise 05
45. Evolved FSM 4
46. Evolved HMM 5
47. EvolvedLookerUp1_1_1
48. EvolvedLookerUp2_2_2
49. FSM Player: [(0, 'C', 0, 'C'), (0, 'D', 3, 'C'), (1, 'C', 5, 'D'), (1, 'D', 0, 'C'), (2, 'C', 3, 'C'), (2, 'D', 2, 'D'), (3, 'C', 4, 'D'), (3, 'D', 6, 'D'), (4, 'C', 3, 'C'), (4, 'D', 1, 'D'), (5, 'C', 6, 'C'), (5, 'D', 3, 'D'), (6, 'C', 6, 'D'), (6, 'D', 6, 'D'), (7, 'C', 7, 'D'), (7, 'D', 5, 'C')], 0, C (**TF3**)
50. FSM Player: [(0, 'C', 13, 'D'), (0, 'D', 12, 'D'), (1, 'C', 3, 'D'), (1, 'D', 4, 'D'), (2, 'C', 14, 'D'), (2, 'D', 9, 'D'), (3, 'C', 0, 'C'), (3, 'D', 1, 'D'), (4, 'C', 1, 'D'), (4, 'D', 2, 'D'), (5, 'C', 12, 'C'), (5, 'D', 6, 'C'), (6, 'C', 1, 'C'), (6, 'D', 14, 'D'), (7, 'C', 12, 'D'), (7, 'D', 2, 'D'), (8, 'C', 7, 'D'), (8, 'D', 9, 'D'), (9, 'C', 8, 'D'), (9, 'D', 0, 'D'), (10, 'C', 2, 'C'), (10, 'D', 15, 'C'), (11, 'C', 7, 'D'), (11, 'D', 13, 'D'), (12, 'C', 3, 'C'), (12, 'D', 8, 'D'), (13, 'C', 7, 'C'), (13, 'D', 10, 'D'), (14, 'C', 10, 'D'), (14, 'D', 7, 'D'), (15, 'C', 15, 'C'), (15, 'D', 11, 'D')], 0, C (**TF2**)
51. FSM Player: [(0, 'C', 7, 'C'), (0, 'D', 1, 'C'), (1, 'C', 11, 'D'), (1, 'D', 11, 'D'), (2, 'C', 8, 'D'), (2, 'D', 8, 'C'), (3, 'C', 3, 'C'), (3, 'D', 12, 'D'), (4, 'C', 6, 'C'), (4, 'D', 3, 'C'), (5, 'C', 11, 'C'), (5, 'D', 8, 'D'), (6, 'C', 13, 'D'), (6, 'D', 14, 'C'), (7, 'C', 4, 'D'), (7, 'D', 2, 'D'), (8, 'C', 14, 'D'), (8, 'D', 8, 'D'), (9, 'C', 0, 'C'), (9, 'D', 10, 'D'), (10, 'C', 8, 'C'), (10, 'D', 15, 'C'), (11, 'C', 6, 'D'), (11, 'D', 5, 'D'), (12, 'C', 6, 'D'), (12, 'D', 9, 'D'), (13, 'C', 9, 'D'), (13, 'D', 8, 'D'), (14, 'C', 8, 'D'), (14, 'D', 13, 'D'), (15, 'C', 4, 'C'), (15, 'D', 5, 'C')], 0, C (**TF1**)
52. Feld: 1.0, 0.5, 200
53. Firm But Fair
54. Fool Me Forever
55. Fool Me Once
56. Forgetful Fool Me Once: 0.05
57. Forgetful Grudger
58. Forgiver
59. Forgiving Tit For Tat (**FTfT**)
60. Fortress3
61. Fortress4
62. GTFT: 0.33
63. General Soft Grudger: n=1,d=4,c=2
64. Gradual
65. Gradual Killer: ('D', 'D', 'D', 'D', 'D', 'C', 'C')
66. Grofman
67. Grudger
68. GrudgerAlternator
69. Grumpy: Nice, 10, -10
70. Handshake
71. Hard Go By Majority
72. Hard Go By Majority: 10
73. Hard Go By Majority: 20
74. Hard Go By Majority: 40
75. Hard Go By Majority: 5
76. Hard Prober
77. Hard Tit For 2 Tats (**HTf2T**)
78. Hard Tit For Tat (**HTfT**)
79. Hesitant QLearner
80. Hopeless
81. Inverse
82. Inverse Punisher
83. Joss: 0.9
84. Level Punisher
85. Limited Retaliate 2: 0.08, 15
86. Limited Retaliate 3: 0.05, 20
87. Limited Retaliate: 0.1, 20
88. MEM2
89. Math Constant Hunter

90. Meta Hunter Aggressive: 7 players
91. Meta Hunter: 6 players
92. Naive Prober: 0.1
93. Negation
94. Nice Average Copier
95. Nydegger
96. Omega TFT: 3, 8
97. Once Bitten
98. Opposite Grudger
99. PSO Gambler 1_1_1
100. PSO Gambler 2_2_2
101. PSO Gambler 2_2_2 Noise 05
102. PSO Gambler Mem1
103. Predator
104. Prober
105. Prober 2
106. Prober 3
107. Prober 4
108. Pun1
109. Punisher
110. Raider
111. Random Hunter
112. Random: 0.5
113. Remorseful Prober: 0.1
114. Resurrection
115. Retaliate 2: 0.08
116. Retaliate 3: 0.05
117. Retaliate: 0.1
118. Revised Downing: True
119. Ripoff
120. Risky QLearner
121. SelfSteem
122. ShortMem
123. Shubik
124. Slow Tit For Two Tats
125. Slow Tit For Two Tats 2
126. Sneaky Tit For Tat
127. Soft Go By Majority
128. Soft Go By Majority: 10
129. Soft Go By Majority: 20
130. Soft Go By Majority: 40
131. Soft Go By Majority: 5
132. Soft Grudger
133. Soft Joss: 0.9
134. SolutionB1
135. SolutionB5
136. Spiteful Tit For Tat
137. Stochastic Cooperator
138. Stochastic WSLS: 0.05
139. Suspicious Tit For Tat
140. Tester
141. ThueMorse
142. ThueMorseInverse
143. Thumper
144. Tit For 2 Tats (**Tf2T**)
145. Tit For Tat (**TfT**)
146. Tricky Cooperator
147. Tricky Defector
148. Tullock: 11
149. Two Tits For Tat (**2TfT**)
150. VeryBad
151. Willing
152. Win-Shift Lose-Stay: D (**WShLSt**)
153. Win-Stay Lose-Shift: C (**WStLSh**)
154. Winner12
155. Winner21
156. Worse and Worse
157. Worse and Worse 2
158. Worse and Worse 3
159. ZD-Extort-2 v2: 0.125, 0.5, 1
160. ZD-Extort-2: 0.1111111111111111, 0.5
161. ZD-Extort-4: 0.23529411764705882, 0.25, 1
162. ZD-GEN-2: 0.125, 0.5, 3
163. ZD-GTFT-2: 0.25, 0.5
164. ZD-SET-2: 0.25, 0.0, 2