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

Marc Harper Vincent Knight Nikoleta E. Glynatsi

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. We find 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) [7] 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).

The Moran Process [17] is a model of evolutionary population dynamics that has been used to gain insights about the evolutionary stability and fixation of strategies in a number of settings. Similarly since the first Iterated Prisoner's Dilemma (IPD) tournament described in [5], the Prisoner's dilemma has been used to understand the evolution of cooperative behaviour in complex systems. Several earlier works have studied iterated games in the context of the prisoner's dilemma [18, 23].

This manuscript provides a detailed numerical analysis of **164** complex and adaptive strategies for the IPD. This is made possible by the Axelrod library [24], an effort to provide software for reproducible research for the IPD. The library now contains over 150 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. The library can conduct matches, tournaments and population dynamics with variations including noise and spatial structure.

We present fixation probabilities for all pairs of strategies in the library, identifying those are effective invaders and those resistant to invasion, for population sizes $N = 2$ to $N = 14$.

In particular we address the following questions:

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

While our results agree with some of the published literature, we find that:

1. Zero determinant strategies are not particularly effective for $N > 2$
2. Long memory strategies are generally more effective than short memory strategies
3. Complex strategies can be effective

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 3.

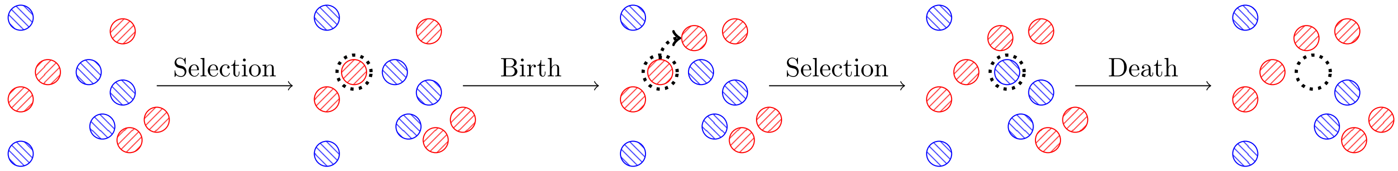


Figure 1: A diagrammatic representation of a Moran process

The Moran process was initially introduced in [17]. 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 [26] 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 [13] 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 [19] and highly-praised [23] it was argued that generous ZD strategies are robust against invading strategies. However, in [11] a strategy using machine learning techniques was capable of resisting invasion and also able to invade any memory one strategy. Recent work [8] 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 recognized 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.

The contribution of this work is a detailed and extensive analysis of absorption probabilities for 164 strategies. These strategies and the numerical simulations are from [24] which is an open source research library written for the study of the IPD. The strategies and simulation frameworks are automatically tested to an extraordinarily high degree of coverage in accordance with best research software practices. 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.

Section 2 will explain the methodological approach used, Section 3 will validate the methodology by comparing simulated results to analytical results in some cases. The main results of this manuscript are presented in Section 4 which will present a detailed analysis of all the data generated. Finally, Section 5 will conclude and offer future avenues for the work presented here.

2 Methodology

To carry out this large numerical experiment, 164 strategies are used from [24]. These include 161 default strategies in the library at the time (excluding strategies classified as having a long run time and those that make use of the length of the game) as well as the following 6 finite state machine machine strategies [4]:

Appendix A shows all the players in question. More information about each player can be obtained in the documentation for [24]. There are 43 stochastic and 124 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.

Memory Depth	0	1	2	3	4	5	6	9	10	11	12	16	20	40	200	∞
Count	3	34	12	8	2	6	1	1	5	1	1	2	2	2	1	86

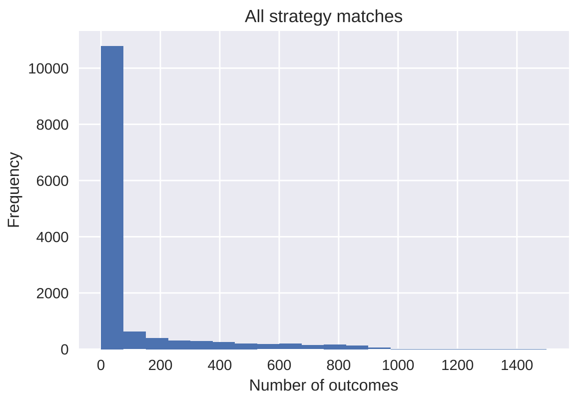
Table 1: Memory depth

Each strategy pair is run for 1000 runs of the Moran process to fixation with starting population distributions of

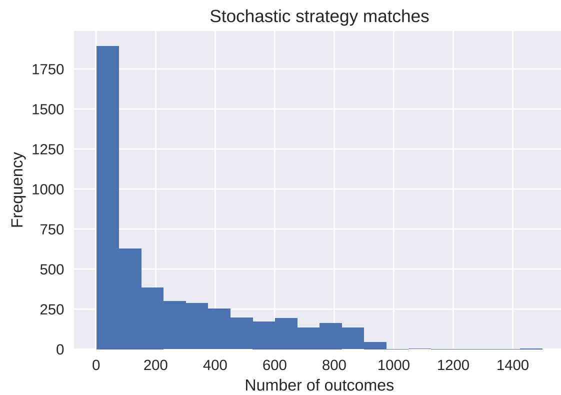
$(1, N-1)$, $(N/2, N/2)$ and $(N-1, 1)$, for N from 2 through 14. The fixation probability is then empirically computed for each combination of starting distribution and value of N.

Our software can carry out exact simulations of the Moran process. Since some of the strategies have a high computational cost or are stochastic, we sample 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.

Figure 2 shows the distribution of the number of outcomes between all strategy pairs.



(a) All matches



(b) Stochastic matches

Figure 2: The distribution of the number of unique outcomes used as the cached results

Section 3 will validate the methodology used here against known theoretic results.

3 Validation

As described in [18] 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 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. We have:

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

where:

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

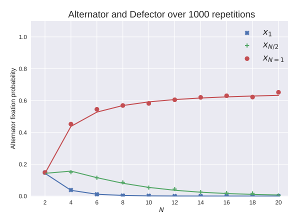
A neutral strategy will have fixation probability $x_i = i/N$. Alternatively, we can frame the outcomes in terms of relative fitness. For a strategy of relative fitness r the fixation probability is well-known to be

$$x_i = \frac{1 - r^{-i}}{1 - r^{-N}}$$

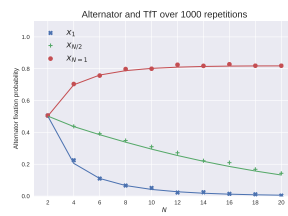
We can use this formula to compute a value r_i that produces the observed value x_i, noting that for arbitrary strategies the value of this effective relative fitness is dependent on i and N.

Comparisons of $x_1, x_{N/2}, x_{N-1}$ are shown in Figure 17. 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.

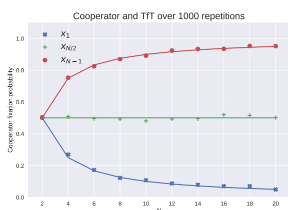
(a) Alternator and Cooperator



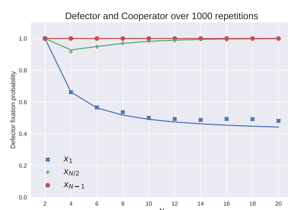
(b) Alternator and Defector



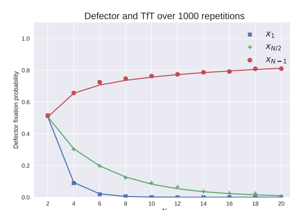
(c) Alternator and Tit For Tat



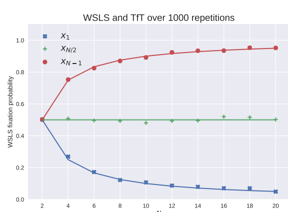
(d) Cooperator and Tit For Tat



(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

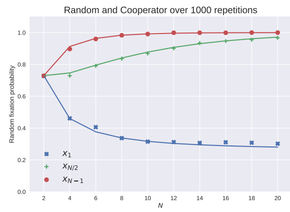
(i) Defector and Win Stay Lose Shift

Figure 3: Comparison of theoretic and actual Moran Process fixation probabilities for **deterministic** strategies

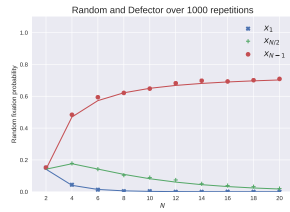
Figure 18 shows the fixation probabilities for stochastic strategies. These are no longer a good match which highlights the weakness of the analytical formulae that relies on the average payoffs. A detailed analysis of the 164 strategies considered, using direct Moran processes will be shown in the next Section.

4 Empirical results

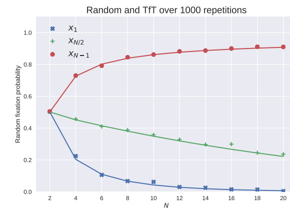
This section outlines the data analysis carried out:



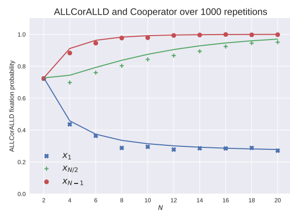
(a) Random and Cooperator



(b) Random and Defector

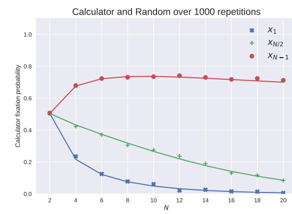


(c) Random and Tit For Tat

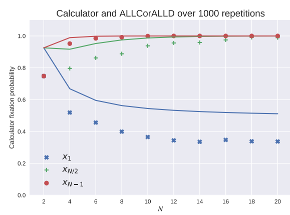


(d) All C or all D and Cooperator

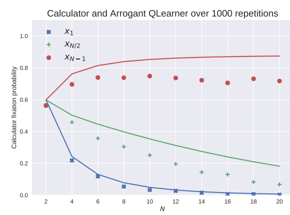
(e) All C or all D and Defector



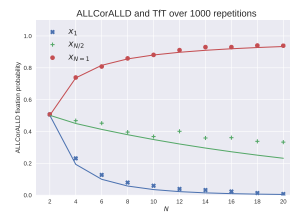
(f) Calculator and Random



(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 4: Comparison of theoretic and actual Moran Process fixation probabilities for **stochastic** strategies

- Section 4.1 considers the specific case of $N = 2$.
- Section 4.2 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 4.3 similarly investigates the ability to defend against an invasion.
- Section 4.4 investigates the relationship between performance for differing population sizes.
- Section 4.5 calculates the relative fitness of all strategies.

4.1 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.

Overall the main fixation probabilities of interest are x_1 and x_{N-1}, these reflect a strategy's ability to invade or resist invasion; for $N = 2$ these two cases coincide. Figure 5 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 [12]. This strategy was specifically designed for evolutionary processes.
2. The finite state machine strategy
3. The Defector: it always defects. As it has little potential interaction with itself, recall that $N = 2$ is considered, its aggressiveness is rewarded.

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

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

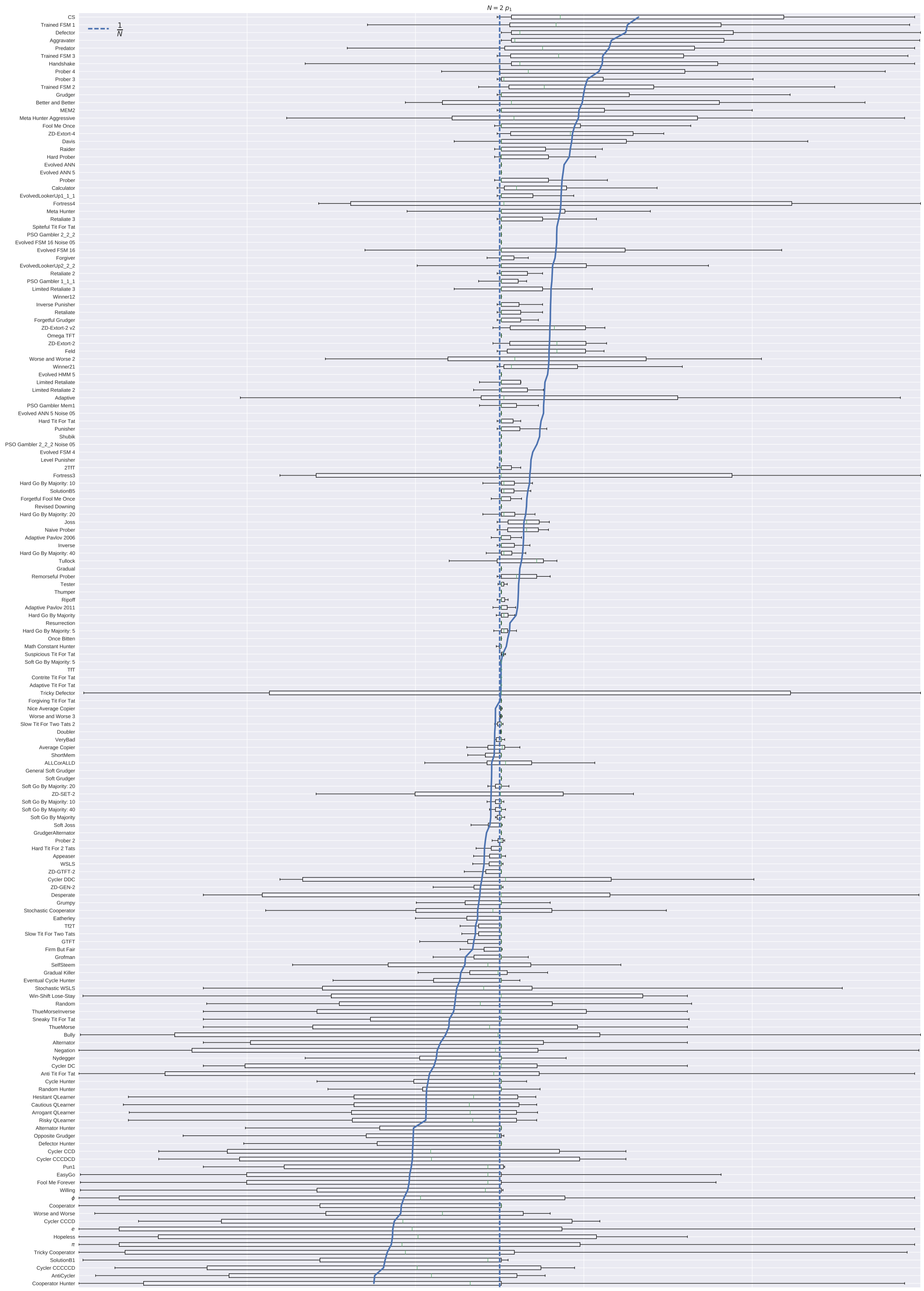


Figure 5: The fixation probabilities for $N = 2$

As will be demonstrated in Section 4.4 the results for $N = 2$ differ from those of larger N. Hence our results do not concur with the literature which suggests that Zero Determinant strategies should be effective for larger population sizes, but we note that those analysis run each match to stationarity, while our matches run for a fixed number of rounds.

In the next sections we pay close attention to strategies who are strong invaders/resistors.

Player	Mean p_1	Memory Depth	Stochastic
CS	0.664994	∞	False
Trained FSM 1	0.651840	1	False
Defector	0.649650	0	False
Aggravater	0.632742	∞	False
Predator	0.629908	9	False

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

4.2 Strong invaders

In this section we focus 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_i. The fixation probabilities are shown in Figures 6, 7 and 8 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	0.450319	∞	False
Grudger	0.433417	∞	False
MEM2	0.430000	∞	False
Fool Me Once	0.426276	∞	False
Prober 4	0.426055	∞	False

(a) $N = 3$

Player	Mean p_1	Memory Depth	Stochastic
Evolved FSM 16	0.254233	16	False
Fool Me Once	0.249706	∞	False
Evolved ANN 5	0.248791	∞	False
PSO Gambler 2_2_2	0.248755	∞	True
Evolved ANN	0.248687	∞	False

(b) $N = 7$

Player	Mean p_1	Memory Depth	Stochastic
Evolved FSM 16	0.212454	16	False
PSO Gambler 2_2_2	0.206442	∞	True
Evolved ANN	0.204902	∞	False
Evolved ANN 5	0.203945	∞	False
EvolvedLookerUp2_2_2	0.203264	∞	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:

- Prober 4, complex strategy with an initial 20 move sequence of cooperations and defections [14]. This initial sequence serves as approximate handshake.
- 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 [13].
- Fool Me Once, forgives one defection but defects forever once a second defection takes place.

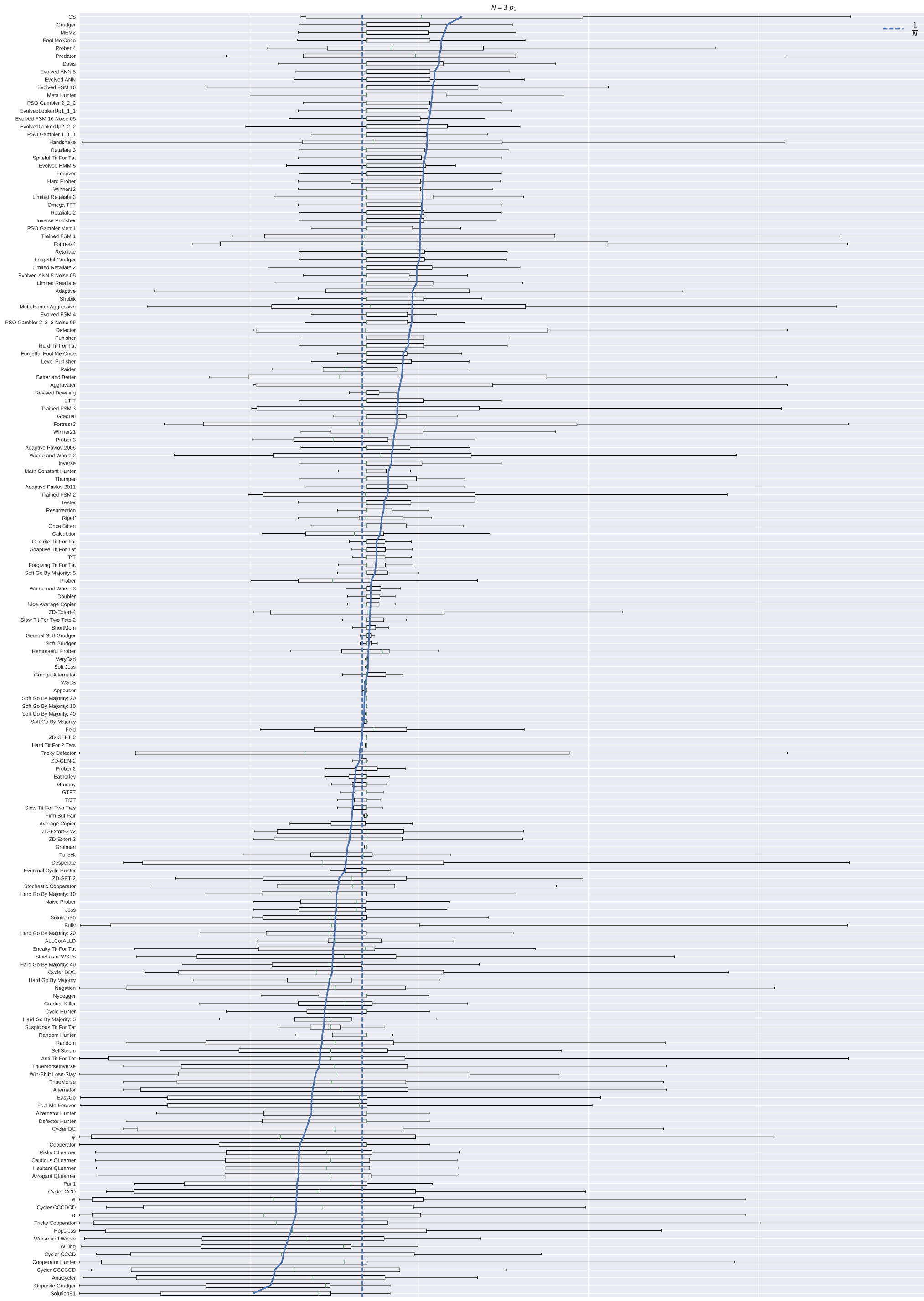


Figure 6: The fixation probability x_1 for $N = 3$

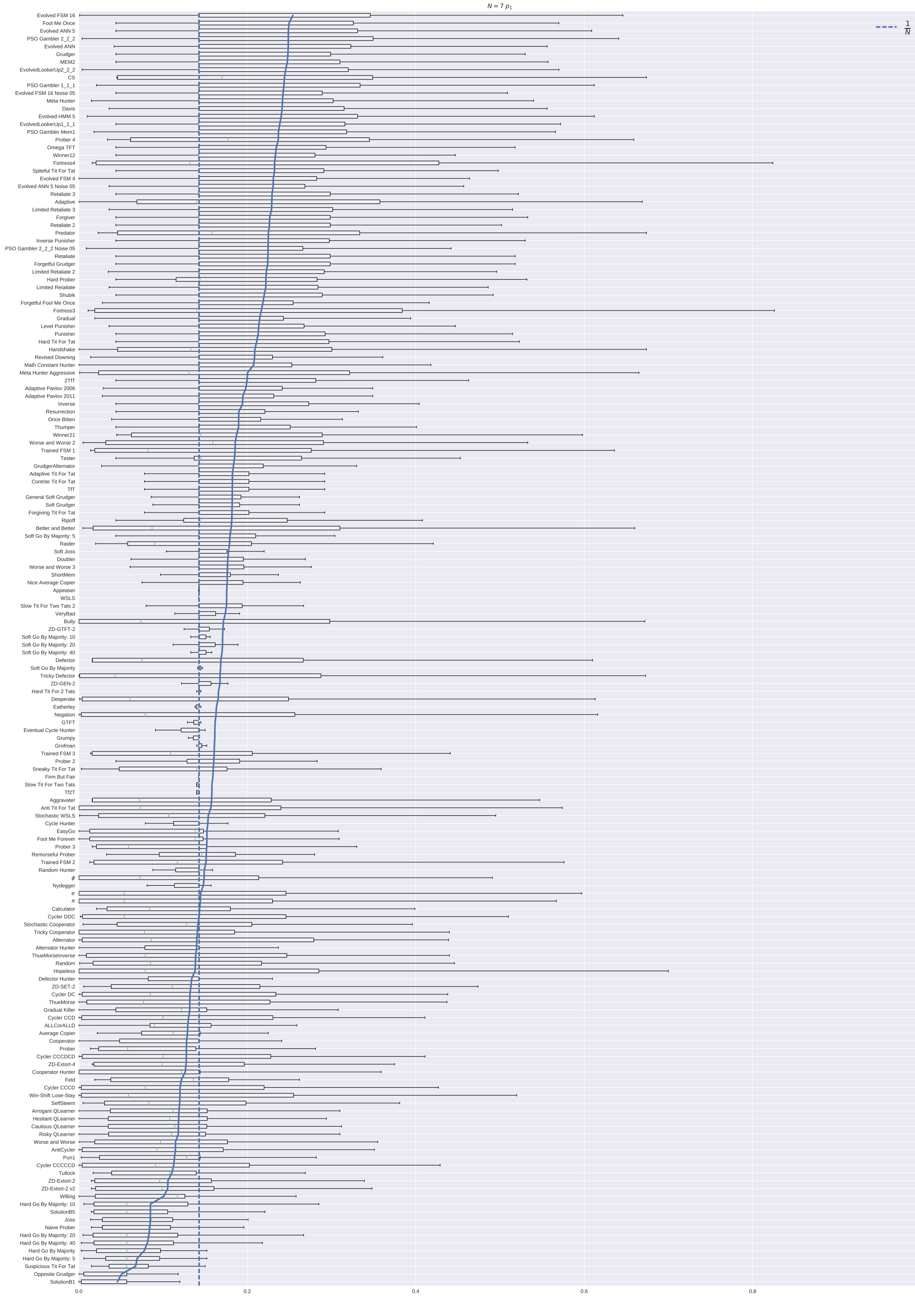


Figure 7: The fixation probability x_1 for $N = 7$

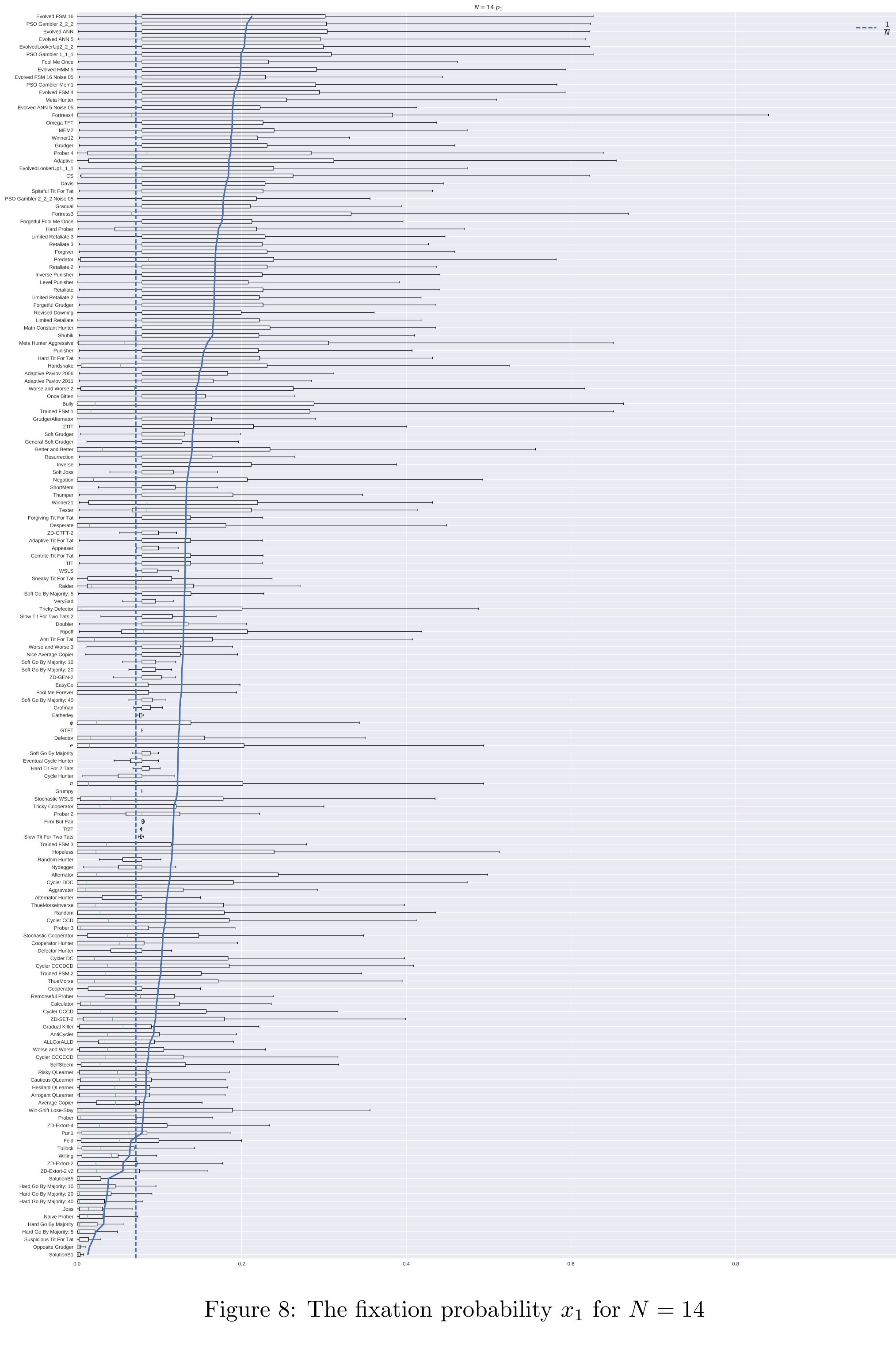


Figure 8: The fixation probability x_1 for $N = 14$

- 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 [10].
- 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 machine strategy that has been trained using an evolutionary algorithm.

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.

4.3 Strong resistors

In addition to identifying good invaders, we also identify strategies resistant to invasion by other strategies 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 9, 7 and 11 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 4.1 perform well for high N.

Player	Mean p_{N-1}	Memory Depth	Stochastic
CS	0.836853	∞	False
Predator	0.813092	9	False
Handshake	0.799092	∞	False
Prober 4	0.791718	∞	False
Grudger	0.762706	∞	False

(a) $N = 3$

Player	Mean p_{N-1}	Memory Depth	Stochastic
CS	0.977018	∞	False
Predator	0.968221	9	False
Prober 4	0.955294	∞	False
Handshake	0.952607	∞	False
Winner21	0.939460	2	False

(b) $N = 7$

Player	Mean p_{N-1}	Memory Depth	Stochastic
CS	0.998491	∞	False
Predator	0.994147	9	False
Prober 4	0.986994	∞	False
Handshake	0.979761	∞	False
Winner21	0.977853	2	False

(c) $N = 14$

Table 4: Properties of top five resistors

There are 3 strategies that are only top performers in x_{N-1}:

- Handshake: a slightly less aggressive version of the Collective strategy [21]. 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.
- Winner 21: a strategy that makes its decision deterministically based on 1 round of its own strategy and 2 of the opponent's strategy [15].

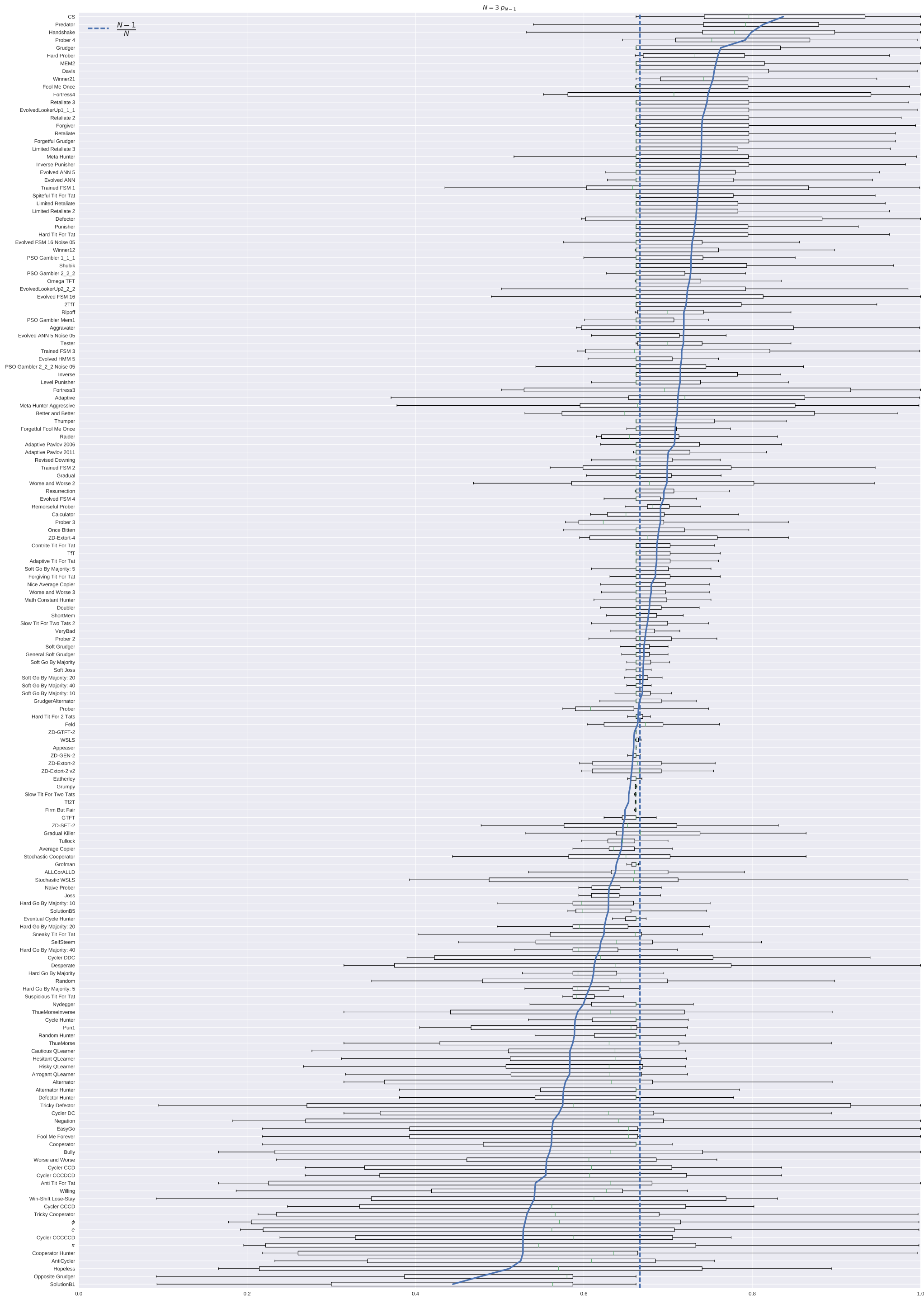


Figure 9: The fixation probability x_{N-1} for $N = 3$

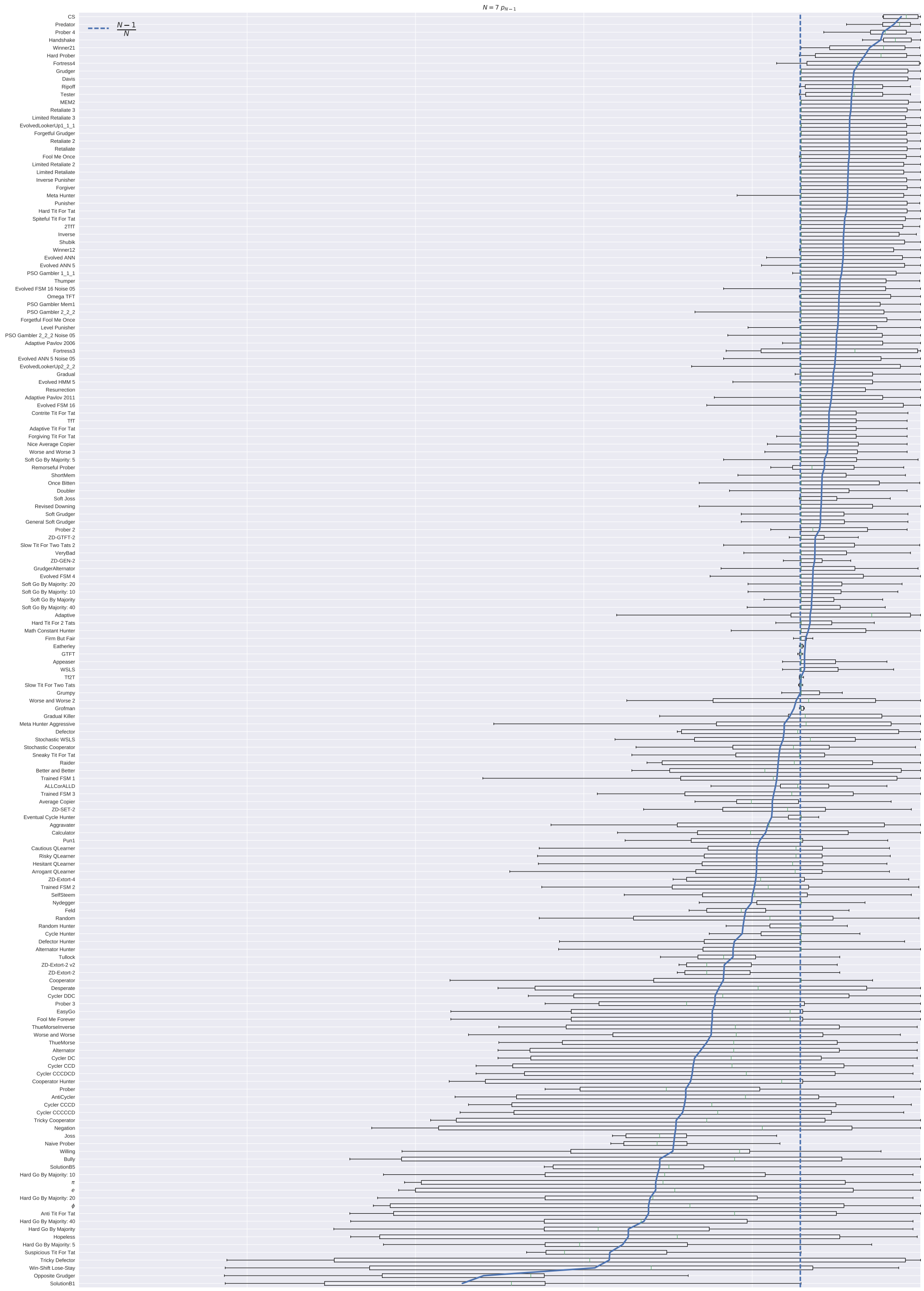


Figure 10: The fixation probability x_{N-1} for $N = 7$

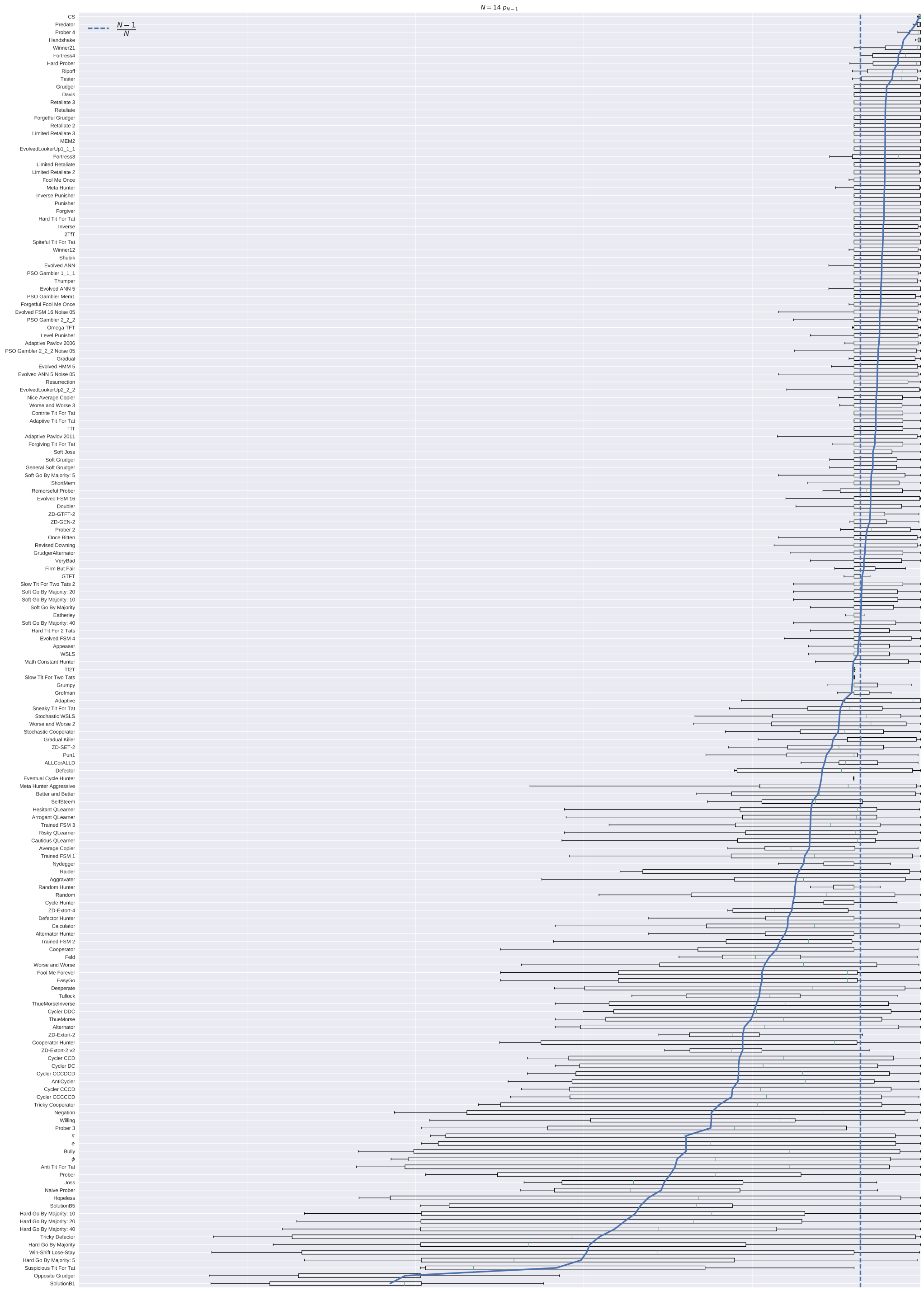


Figure 11: The fixation probability x_{N-1} for $N = 14$

- Predator: a finite state machine described in [4]

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 we note that it is possible to design a strategy with a "stochastic handshake" [11].

It is evident through Sections 4.1, 4.2 and 4.3 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.

4.4 The effect of population size

Figures 12, 13 and 14 show the median rank of each strategy against population size. 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 confirms what has been discussed in previous sections.

Tables 5, 6 and 7 show the same information for the strategies that rated high for $N = 2$ and $N = 14$.

Player	2	3	4	5	6	7	8	9	10	11	12	13	14
CS	1.0	1.0	2.0	8.0	4.0	9.0	12.0	20.0	14.0	21.0	15.0	23.0	22.0
Trained FSM 1	2.0	29.0	55.0	71.0	67.0	57.0	54.0	71.0	60.0	68.0	53.0	59.0	53.0
Defector	3.0	41.0	75.0	88.0	86.0	84.0	85.0	99.0	94.0	104.0	90.0	100.0	96.0
Aggravater	4.0	48.0	88.0	98.0	101.0	102.0	107.0	112.0	111.0	114.0	113.0	113.0	116.0
Predator	5.0	6.0	22.0	33.0	25.0	29.0	29.0	41.0	32.0	41.0	31.0	41.0	33.0
Evolved FSM 16	31.0	10.0	8.0	4.0	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	12.0	9.0	7.0	8.0	4.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Evolved ANN	20.0	9.0	6.0	5.0	7.0	5.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Evolved ANN 5	21.0	8.0	5.0	6.0	6.0	3.0	5.0	4.0	4.0	4.0	5.0	4.0	4.0
EvolvedLookerUp2_2_2	33.0	15.0	11.0	10.0	9.0	8.0	6.0	6.0	5.0	5.0	4.0	5.0	5.0

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

Player	2	3	4	5	6	7	8	9	10	11	12	13	14
CS	1.0	1.0	2.0	8.0	4.0	9.0	12.0	20.0	14.0	21.0	15.0	23.0	22.0
Trained FSM 1	2.0	29.0	55.0	71.0	67.0	57.0	54.0	71.0	60.0	68.0	53.0	59.0	53.0
Defector	3.0	41.0	75.0	88.0	86.0	84.0	85.0	99.0	94.0	104.0	90.0	100.0	96.0
Aggravater	4.0	48.0	88.0	98.0	101.0	102.0	107.0	112.0	111.0	114.0	113.0	113.0	116.0
Predator	5.0	6.0	22.0	33.0	25.0	29.0	29.0	41.0	32.0	41.0	31.0	41.0	33.0
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
Predator	5.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Prober 4	8.0	4.0	4.0	4.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Handshake	6.0	3.0	3.0	3.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Winner21	44.0	9.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0

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	2.0	4.0	12.0	14.0	15.0	22.0
Trained FSM 1	2.0	55.0	67.0	54.0	60.0	53.0	53.0
Defector	3.0	75.0	86.0	85.0	94.0	90.0	96.0
Aggravater	4.0	88.0	101.0	107.0	111.0	113.0	116.0
Predator	5.0	22.0	25.0	29.0	32.0	31.0	33.0
CS	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Predator	5.0	2.0	2.0	2.0	2.0	2.0	2.0
Prober 4	8.0	3.0	3.0	3.0	3.0	3.0	3.0
Handshake	7.0	5.0	4.0	4.0	4.0	4.0	4.0
Grudger	11.0	4.0	5.0	5.0	5.0	5.0	5.0

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

Tables 8a, 8b and 8c show the correlation coefficients of the ranks of strategies in differing population size. This is shown graphically in Figure 15. 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 [19] does not extend to larger populations. This was discussed theoretically in [1] however not observed empirically at the scale presented here.

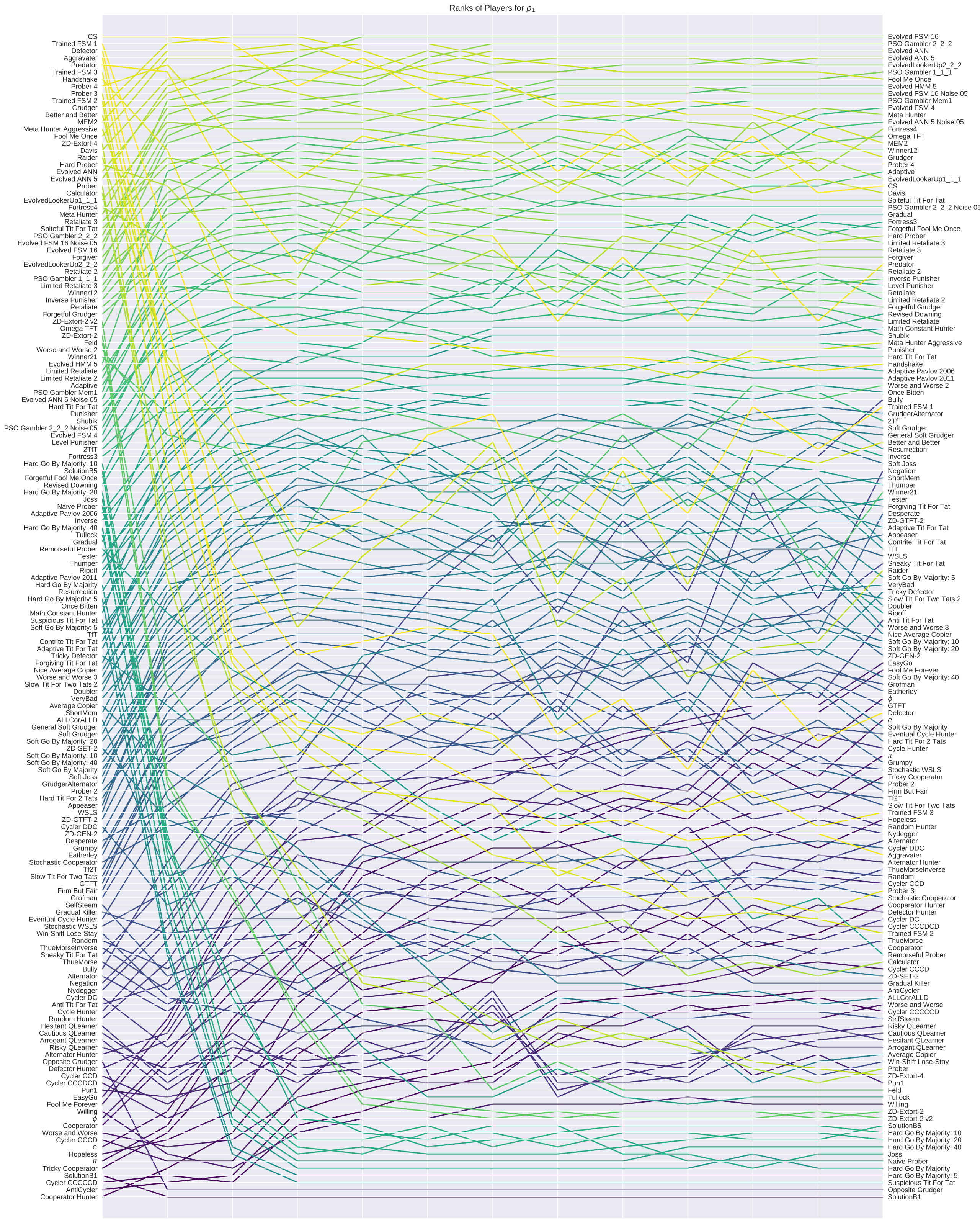


Figure 12: Ranks of all strategies according to x_1 for different population sizes

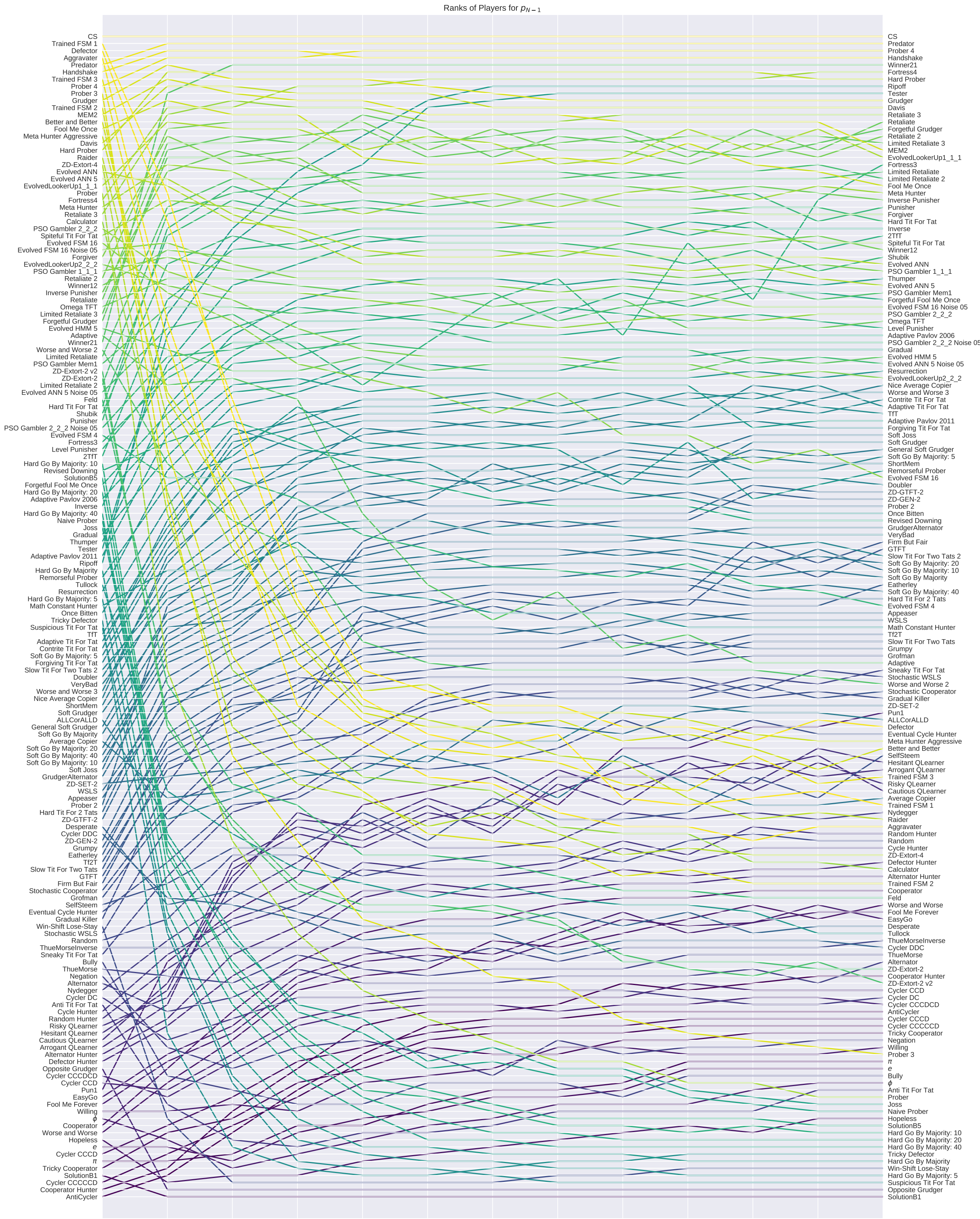


Figure 13: Ranks of all strategies according to x_{N-1} for different population sizes

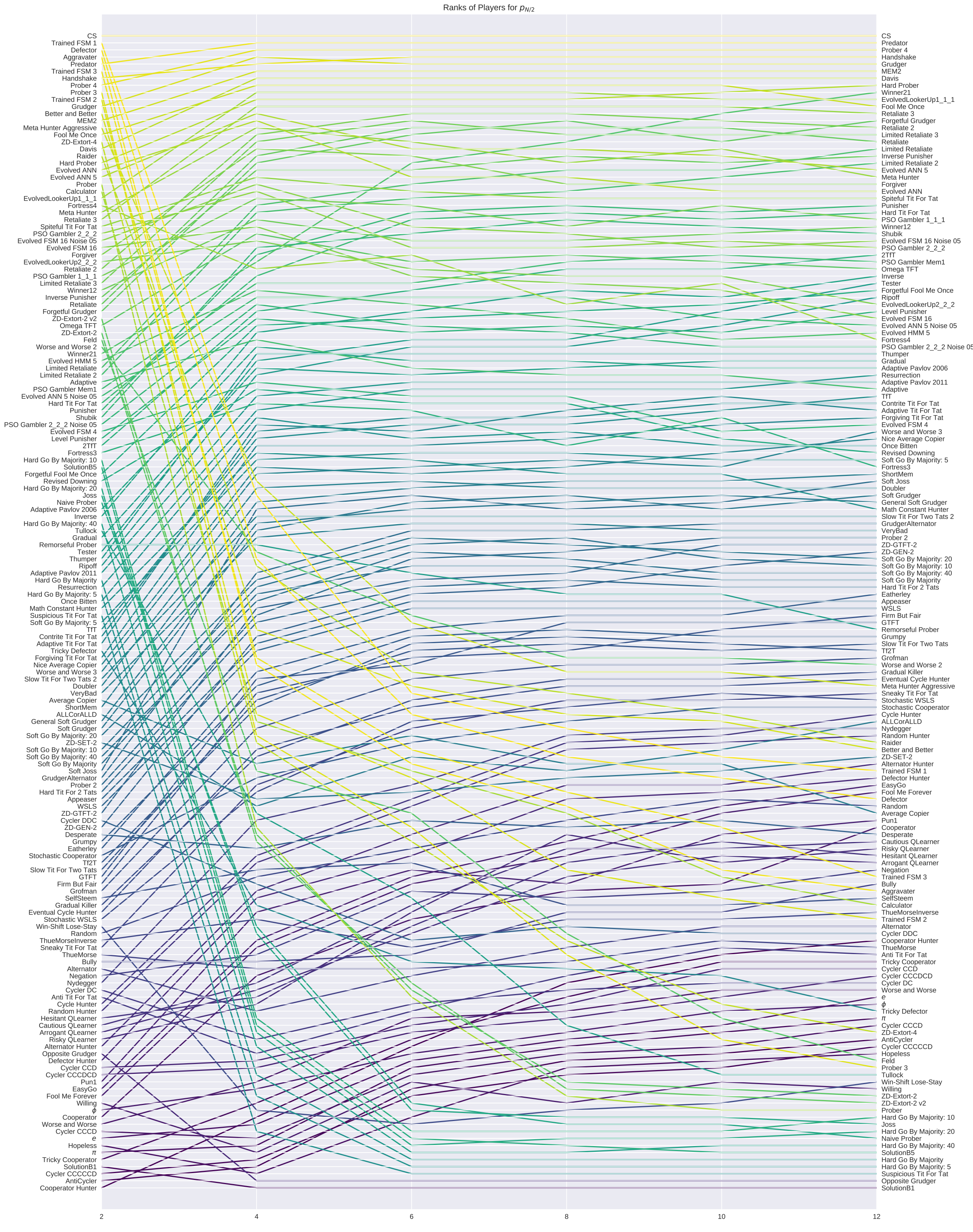


Figure 14: Ranks of all strategies according to $x_{N/2}$ for different population sizes

N	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1.00	0.89	0.75	0.66	0.64	0.63	0.61	0.57	0.58	0.55	0.56	0.55	0.55
3	0.89	1.00	0.96	0.93	0.91	0.90	0.89	0.87	0.87	0.86	0.85	0.85	0.84
4	0.75	0.96	1.00	0.99	0.98	0.97	0.96	0.96	0.95	0.95	0.94	0.94	0.93
5	0.66	0.93	0.99	1.00	0.99	0.99	0.98	0.98	0.98	0.98	0.96	0.96	0.95
6	0.64	0.91	0.98	0.99	1.00	1.00	0.99	0.99	0.99	0.99	0.98	0.98	0.97
7	0.63	0.90	0.97	0.99	1.00	1.00	1.00	0.99	1.00	0.99	0.99	0.99	0.98
8	0.61	0.89	0.96	0.98	0.99	1.00	1.00	0.99	1.00	0.99	0.99	0.99	0.99
9	0.57	0.87	0.96	0.98	0.99	0.99	0.99	1.00	1.00	1.00	0.99	0.99	0.99
10	0.58	0.87	0.95	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
11	0.55	0.86	0.95	0.98	0.99	0.99	0.99	1.00	1.00	1.00	0.99	1.00	0.99
12	0.56	0.85	0.94	0.96	0.98	0.99	0.99	0.99	1.00	0.99	1.00	1.00	1.00
13	0.55	0.85	0.94	0.96	0.98	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00
14	0.55	0.84	0.93	0.95	0.97	0.98	0.99	0.99	0.99	0.99	1.00	1.00	1.00

(a) Correlation coefficients for ranks for invasion

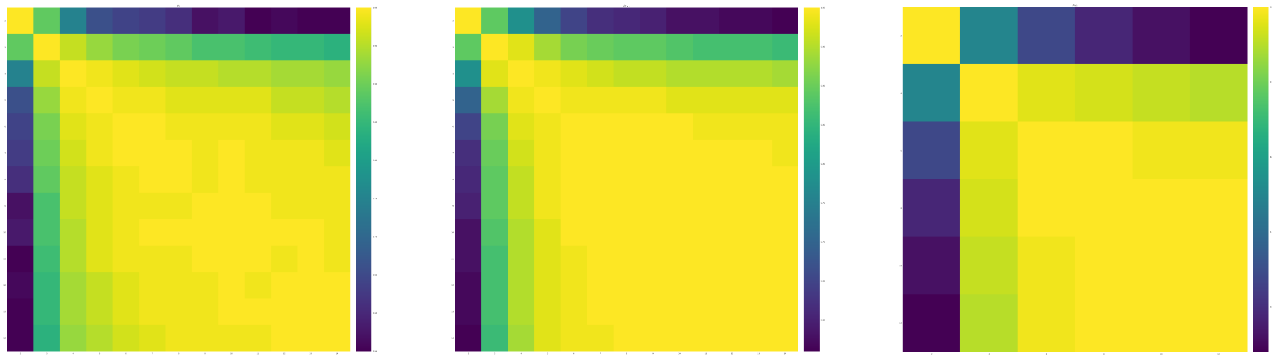
N	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1.00	0.89	0.78	0.70	0.65	0.62	0.61	0.60	0.58	0.58	0.57	0.57	0.56
3	0.89	1.00	0.98	0.94	0.91	0.90	0.89	0.89	0.88	0.87	0.87	0.87	0.86
4	0.78	0.98	1.00	0.99	0.98	0.97	0.96	0.96	0.95	0.95	0.95	0.95	0.94
5	0.70	0.94	0.99	1.00	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.98	0.98
6	0.65	0.91	0.98	0.99	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99
7	0.62	0.90	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
8	0.61	0.89	0.96	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	0.60	0.89	0.96	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	0.58	0.88	0.95	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	0.58	0.87	0.95	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	0.57	0.87	0.95	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	0.57	0.87	0.95	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14	0.56	0.86	0.94	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00

(b) Correlation coefficients for ranks for resistance

N	2	4	6	8	10	12
2	1.00	0.75	0.64	0.59	0.56	0.54
4	0.75	1.00	0.98	0.97	0.96	0.95
6	0.64	0.98	1.00	1.00	0.99	0.99
8	0.59	0.97	1.00	1.00	1.00	1.00
10	0.56	0.96	0.99	1.00	1.00	1.00
12	0.54	0.95	0.99	1.00	1.00	1.00

(c) Correlation coefficients for ranks for coexistance

Table 8: Correlation coefficients of rankings by population size



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

Figure 15: Heatmap of correlation coefficients of rankings by population size

4.5 Relative fitness

Under the assumption of a constant relative fitness r between two strategies [18] the formula for x_i (for given N, r is:

$$x_i = x_i(r) = \frac{1 - \frac{1}{r^i}}{1 - \frac{1}{r^N}} \tag{10}$$

Figure 16 shows this function for $N = 10$ and $i \in \{1, 5, 10\}$.

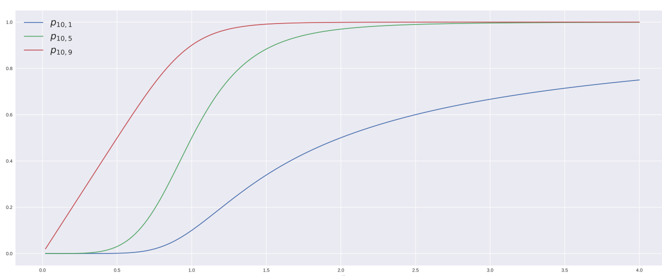


Figure 16: $x_i(r)$

The first and second derivative of (10) is given by equations (11) and (12).

$$\frac{dx_i}{dr} = \frac{r^{N-i-1}}{r^{2N} - 2r^N + 1}\left(-Nr^i + N + ir^N - i\right) \tag{11}$$

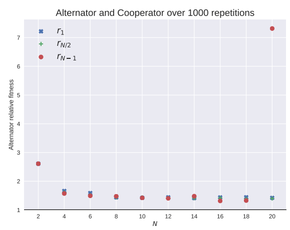
$$\frac{d^2x_i}{dr^2} = \frac{r^{N-i-2}}{\left(r^N - 1\right)^3}\left(2N^2\left(r^i - 1\right) + N\left(r^N - 1\right)\left(N\left(r^i - 1\right) - 2i + r^i - 1\right) - i\left(i + 1\right)\left(r^N - 1\right)^2\right) \tag{12}$$

Using these, Halley's method [2] can be used to efficiently numerically invert $x_i(r)$ to obtain a theoretic relative fitness r that gives the calculated $x_i(r)$ between two strategies for a given N, i.

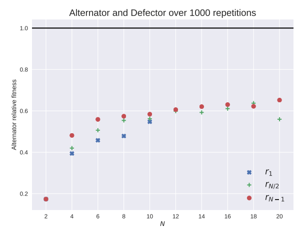
5 Conclusion

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.

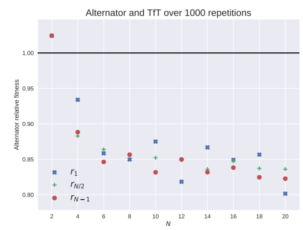
This is the largest such experiment carried out and has lead to many insights.



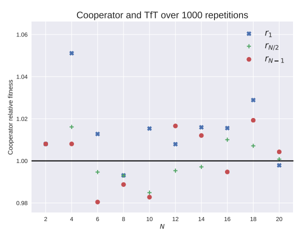
(a) Alternator and Cooperator



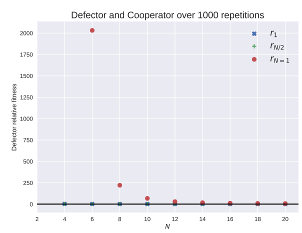
(b) Alternator and Defector



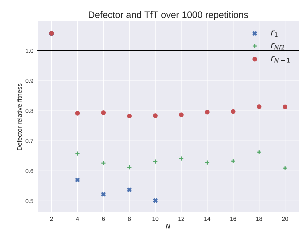
(c) Alternator and Tit For Tat



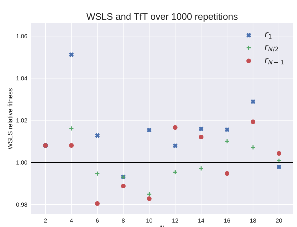
(d) Cooperator and Tit For Tat



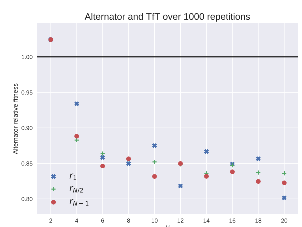
(e) Defector and Cooperator



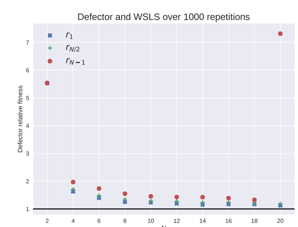
(f) Defector and Tit For Tat



(g) Win Stay Lose Shift and Tit For Tat

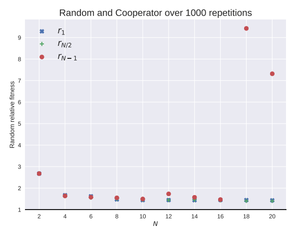


(h) Alternator and Win Stay Lose Shift

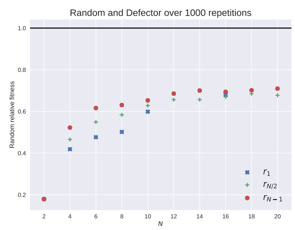


(i) Defector and Win Stay Lose Shift

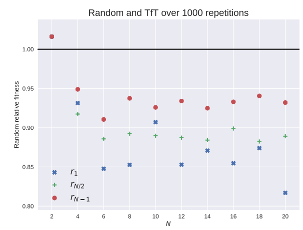
Figure 17: Estimated relative fitness for **deterministic** strategies



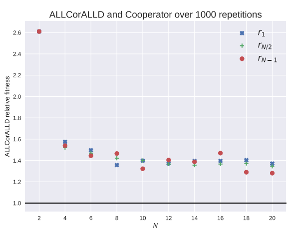
(a) Random and Cooperator



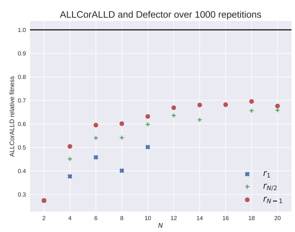
(b) Random and Defector



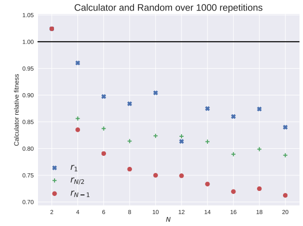
(c) Random and Tit For Tat



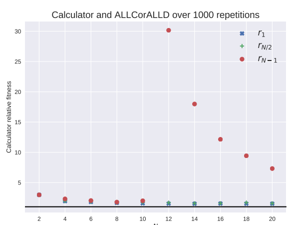
(d) All C or all D and Cooperator



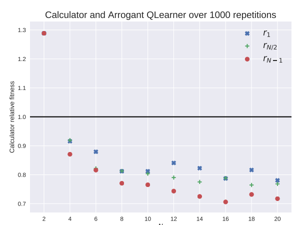
(e) All C or all D and Defector



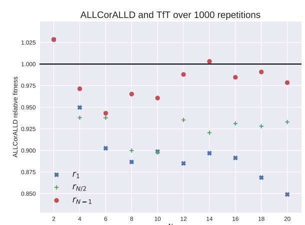
(f) Calculator and Random



(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 18: Estimated relative fitness for **stochastic** strategies

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 4.2 and 4.3 but also by considering the correlation of the ranks in different population sizes in Section 4.4.

Memory one strategies do not perform well, as predicted by [19]. However, 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.

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 [8].

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 [20, 22, 27]. The raw data as well as the processed data has also been properly archived.

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) [24].
- The matplotlib library (visualisation) [9].
- The pandas and numpy libraries (data manipulation) [16, 25].

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] Author G Alefeld. "On the Convergence of Halley ' s Method". In: 88.7 (2012), pp. 530–536.

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

[8] 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.

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

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

[11] 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.

[12] 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`.

[13] 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.

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

[15] Philippe Mathieu and Jean-Paul Delahaye. “New Winning Strategies for the Iterated Prisoner’s Dilemma (Extended Abstract)”. In: *14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)* (2015), pp. 1665–1666. ISSN: 15582914.

[16] 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.

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

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

[19] 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`.

[20] 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`.

[21] 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`.

[22] 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`.

[23] 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`.

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

[25] 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.

[26] 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`.

[27] 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 (**Trained FSM 1**)
50. 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')], 1, C (**Incorrect Trained FSM 1**)
51. 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 (**Trained FSM 2**)
52. 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')], 1, C (**Incorrect Trained FSM 2**)
53. 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 (**Trained FSM 3**)
54. 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')], 1, C (**Incorrect Trained FSM 3**)
55. Feld: 1.0, 0.5, 200
56. Firm But Fair
57. Fool Me Forever
58. Fool Me Once
59. Forgetful Fool Me Once: 0.05
60. Forgetful Grudger
61. Forgiver
62. Forgiving Tit For Tat (**FTfT**)
63. Fortress3
64. Fortress4
65. GTFT: 0.33
66. General Soft Grudger: n=1,d=4,c=2
67. Gradual
68. Gradual Killer: ('D', 'D', 'D', 'D', 'D', 'C', 'C')
69. Grofman
70. Grudger
71. GrudgerAlternator

72. Grumpy: Nice, 10, -10
73. Handshake
74. Hard Go By Majority
75. Hard Go By Majority: 10
76. Hard Go By Majority: 20
77. Hard Go By Majority: 40
78. Hard Go By Majority: 5
79. Hard Prober
80. Hard Tit For 2 Tats (**HTf2T**)
81. Hard Tit For Tat (**HTfT**)
82. Hesitant QLearner
83. Hopeless
84. Inverse
85. Inverse Punisher
86. Joss: 0.9
87. Level Punisher
88. Limited Retaliate 2: 0.08, 15
89. Limited Retaliate 3: 0.05, 20
90. Limited Retaliate: 0.1, 20
91. MEM2
92. Math Constant Hunter
93. Meta Hunter Aggressive: 7 players
94. Meta Hunter: 6 players
95. Naive Prober: 0.1
96. Negation
97. Nice Average Copier
98. Nydegger
99. Omega TFT: 3, 8
100. Once Bitten
101. Opposite Grudger
102. PSO Gambler 1_1_1
103. PSO Gambler 2_2_2
104. PSO Gambler 2_2_2 Noise 05
105. PSO Gambler Mem1
106. Predator
107. Prober
108. Prober 2
109. Prober 3
110. Prober 4
111. Pun1
112. Punisher
113. Raider
114. Random Hunter
115. Random: 0.5
116. Remorseful Prober: 0.1
117. Resurrection
118. Retaliate 2: 0.08
119. Retaliate 3: 0.05
120. Retaliate: 0.1
121. Revised Downing: True
122. Ripoff
123. Risky QLearner
124. SelfSteem
125. ShortMem
126. Shubik
127. Slow Tit For Two Tats
128. Slow Tit For Two Tats 2
129. Sneaky Tit For Tat
130. Soft Go By Majority
131. Soft Go By Majority: 10
132. Soft Go By Majority: 20
133. Soft Go By Majority: 40
134. Soft Go By Majority: 5
135. Soft Grudger
136. Soft Joss: 0.9
137. SolutionB1
138. SolutionB5
139. Spiteful Tit For Tat
140. Stochastic Cooperator
141. Stochastic WSLS: 0.05
142. Suspicious Tit For Tat
143. Tester
144. ThueMorse
145. ThueMorseInverse
146. Thumper
147. Tit For 2 Tats (**Tf2T**)
148. Tit For Tat (**TfT**)
149. Tricky Cooperator
150. Tricky Defector
151. Tullock: 11
152. Two Tits For Tat (**2TfT**)
153. VeryBad
154. Willing
155. Win-Shift Lose-Stay: D (**WShLSt**)
156. Win-Stay Lose-Shift: C (**WStLSh**)
157. Winner12
158. Winner21
159. Worse and Worse
160. Worse and Worse 2
161. Worse and Worse 3
162. ZD-Extort-2 v2: 0.125, 0.5, 1
163. ZD-Extort-2: 0.1111111111111111, 0.5
164. ZD-Extort-4: 0.23529411764705882, 0.25, 1
165. ZD-GEN-2: 0.125, 0.5, 3
166. ZD-GTFT-2: 0.25, 0.5 (**ZD-GTFT**)
167. ZD-SET-2: 0.25, 0.0, 2