A meta analysis of tournaments and an evaluation of performance in the Iterated Prisoner's Dilemma.

Nikoleta E. Glynatsi, Dr Vincent A. Knight

Abstract

The iterated prisoner's dilemma has been used for decades as a powerful model of behavioural interactions. From the celebrated performance of Tit for Tat, to the introduction of the zero-determinant family, to the use of sophisticated strategies such as neural networks, the literature has been exploring the performance of strategies in the game for years. Most of these strategies are now accessible due to an open source package. This manuscript make use of the open source and it's strategy data base to a conducts a meta analysis of iterated prisoner's dilemma tournaments. The aim is to evaluate the performance of these strategies and finally answer the question: which is the best strategy in the game?

1 Background

The iterated prisoner's dilemma is a repeated two player game that models situations in which self-interest clashes with collective interest. At each turn, the players simultaneously and independently make a choice between cooperation (C) and defection (D) whilst their prior interactions matter. The payoffs at each given turn are defined by the matrix,

$$\begin{pmatrix} R & S \\ T & P \end{pmatrix}$$

where $T > R > P > S$ and $2R > T + S$. The most common values used in the literature, and in this paper, are $R = 3, P = 1, T = 5, S = 0$.

Since the computer tournaments of R. Axelrod in 1980s several academic papers are published in the field regarding the performance of strategies in the iterated prisoner's dilemma. In the 80's following the strong performance of Tit For Tat in both Axelrod's computer tournaments [7, 8], and moreover in a series of evolutionary experiments [9], the strategy was thought as the most robust basic strategy in the iterated prisoner's dilemma. However, the strategy's poor performance in environments with noise [10, 13, 16, 21], made room for other protagonists, such as, Nice and Forgiving [10], Pavlov [18] and Generous Tit For Tat [19].

Yet again, in 2012 another set of strategies was introduced as the dominant set of strategies [20]. These were called zero-determinant strategies, and by forcing a linear relationship between the payoffs they can ensure that they will never receive less than their opponents. However, in [14] a tournament containing over 200 strategies, including the zero-determinant strategies introduced in [20], was performed and none of the zero-determinant strategies ranked in top spots. Instead, the top ranked strategies were a set of trained strategies based lookup tables, hidden markov models and finite state automata.

Thus, the following question is raised here: which are the true dominant strategies in the iterated prisoner's dilemma?

This manuscript uses the open source package Axelrod-Python [2] to simulate a large number of computer tournaments using as many strategies as possible from the literature. The aim is evaluate the performance of these strategies in a tournament and futhermore, explore the factors of their success. This is done not for standard round robin tournaments, but also for noisy, probabilistic ending and noisy probabilistic ending tournaments.

The structure of this manuscript is as follows. In Section 2 the data collection is covered. In Section 3, the best performed strategies for each type of tournament and overall are presented. Section 4, explores the traits which contribute to good performance and finally in Section 5 we conclude and summarise the results.

2 Data generating process

For the purposes of this manuscript a data set containing results on iterate prisoner's dilemma tournaments has been generated, which is available at. This was done using the open source package Axelrod-Python [2], more specifically, version 3.0.0. Axelrod-Python allows for different types of iterated prisoner's dilemma computer tournaments to be simulated whilst containing a list of over 190 strategies. Most of these are strategies described in the literature with a few exceptions being strategies that have been contributed specifically to the package. Though Axelrod-Python features several tournament types, this work considers only standard, noisy, probabilistic ending and noisy probabilistic ending tournaments.

Standard tournaments, are tournaments similar to that of Axelrod's in [7]. There are N strategies which all play an iterated game of n number of turns against each other. Note that self interactions and a match against a random strategy are not included. Similarly, **noisy tournaments** have N strategies and n number of turns but at each turn there is a probability p that a player's action will be flipped. **Probabilistic ending tournaments**, are of size N and after each turn a match between strategies ends with a given probability e. Finally, **noisy probabilistic ending** tournaments have both a noise probability p and an ending probability e. For smoothing the simulated results a tournament is repeated for r number of times and the ranks are based on the average score a strategy achieved and not by number of wins. A summary of each tournaments' parameters is given in Table 1.

tournament type	number of strategies	turns	noise probability	probability of match ending
standard	N	n	-	-
noisy	N	n	p	-
probabilistic ending	N	-	-	e
noisy probabilistic ending	N	-	p	e

Table 1: Tournament types' parameters.

The process of generating data implemented in this manuscript is given by Algorithm 1. For each trial a random size N is selected, and from the list of 195 strategies in [2], a random list of N strategies is chosen. For the given list of strategies a standard, noise, probabilistic ending and a noisy probabilistic ending tournament are performed and repeated r times. The parameters for the tournaments as well as the number of repetitions are selected once for each trial. The parameters and their respective minimum and maximum values are given by Table 2.

The source code for the data generating process as well as the source code for the analysis which will be discussed in the following sections have been written following best practices [3, 11]. It has been packaged and is available here.

Algorithm 1: Data generating Algorithm

foreach *seed* $\in [0, 12285]$ **do**

  $N \leftarrow$ randomly select integer $\in [N_{min}, N_{max}]$;

  players $\leftarrow$ randomly select N players;

  $r \leftarrow$ randomly select integer $\in [r_{min}, r_{max}]$;

  $n \leftarrow$ randomly select integer $\in [n_{min}, n_{max}]$;

  $p \leftarrow$ randomly select float $\in [p_{min}, p_{max}]$;

  $e \leftarrow$ randomly select float $\in [e_{min}, e_{max}]$;

  result standard $\leftarrow$ Axelrod.tournament(players, n, r);

  result noisy $\leftarrow$ Axelrod.tournament(players, n, p, r);

  result probabilistic ending $\leftarrow$ Axelrod.tournament(players, e, r);

  result noisy probabilistic ending $\leftarrow$ Axelrod.tournament(players, p, e, r);

return *result standard, result noisy, result probabilistic ending, result noisy probabilistic ending*;

parameter	parameter explanation	min value	max value
N	number of strategies	3	195
r	number of repetitions	10	100
n	number of turns	1	200
p	probability of flipping action at each turn	0	1
e	probability of match ending in the next turn	0	1

Table 2: Data generation parameters' values

A total of 12,285 trials of Algorithm 1 have been performed. For each trial the results for 4 different tournaments were collected, thus a total of 49,140 $(12,285 \times 4)$ tournament results have been retrieved. Each tournament outputs a result summary in the form of Table 3. The result summary has a length N because each row contains information for each strategy that participated in the tournament. The information include the strategy's rank (R), median score, the rate with which the strategy cooperated (C_r), it's wins and the probability that the strategy cooperated in the opening move. Moreover the summary result includes the rates of a strategy being in any of the four states (CC, CD, DC, DD), and the rate of which the strategy cooperated after each state.

						Rates							
Rank (R)	Name	Median score	Cooperation rating (C_r)	Win	Initial C	CC	CD	DC	DD	CC to C	CD to C	DC to C	DD to C
0	EvolvedLookerUp2 2 2	2.97	0.705	28.0	1.0	0.639	0.066	0.189	0.106	0.836	0.481	0.568	0.8
1	Evolved FSM 16 Noise 05	2.875	0.697	21.0	1.0	0.676	0.020	0.135	0.168	0.985	0.571	0.392	0.07
2	PSO Gambler 1 1 1	2.874	0.684	23.0	1.0	0.651	0.034	0.152	0.164	1.000	0.283	0.000	0.136
3	PSO Gambler Mem1	2.861	0.706	23.0	1.0	0.663	0.042	0.145	0.150	1.000	0.510	0.000	0.122
4	Winner12	2.835	0.682	20.0	1.0	0.651	0.031	0.141	0.177	1.000	0.441	0.000	0.462
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Table 3: Output result.

Several other measures have been manually calculated/retrieved and have been included in the data set. These include properties of the strategies as well as measures regarding the tournament, such as the parameters of

Table 2. All these measures are summarized and described Table 4.

measure	measure explanation	source	value type	min value	max value
normalized rank (r)	The rank of the strategy divided by the tournament size	result summary	float	0	1
memory depth	Strategy's memory size	strategy classifier from [2]	integer	0	∞
stochastic	If a strategy is stochastic	strategy classifier from [2]	boolean	False	True
makes use of game	If a strategy makes used of the game information	strategy classifier from [2]	boolean	False	True
makes use of length	If a strategy makes used of the number of turns	strategy classifier from [2]	boolean	False	True
SSE	A measure of how far a strategy is from extortionate behaviour	method described in [ref]	float	0	1
max cooperating rate (C_{max})	The biggest cooperating rate in the tournament	result summary	float	0	1
min cooperating rate (C_{min})	The smallest cooperating rate in the tournament	result summary	float	0	1
median cooperating rate (C_{median})	The median cooperating rate in the tournament	result summary	float	0	1
mean cooperating rate (C_{mean})	The mean cooperating rate in the tournament	result summary	float	0	1
C_r / C_{max}	A strategy's cooperating rate divided by the maximum	manually	float	0	1
C_r / C_{min}	A strategy's cooperating rate divided by the minimum	manually	float	0	1
C_r / C_{median}	A strategy's cooperating rate divided by the median	manually	float	0	1
C_r / C_{mean}	A strategy's cooperating rate divided by the mean	manually	float	0	1

Table 4: Manually calculated/retrieved measures.

These measures will be used in more detail in the following sections.

The data collection and the output from the generated tournaments have been covered in this section. The measures that will be used in the analysis of the following sections have been presented. These are also summarised in Table in the Appendix. In the following Section, the top performances at each tournament are overall are presented.

3 Top ranked strategies

This section evaluates the performance of 186 strategies. These can be find in the Appendix. The performance of each strategy will be evaluated for each type of tournament independently. In Section 3.1 the strategies are evaluated on their performance in standard tournaments, in Section 3.2 in noisy, and the rest in Sections 3.3 and 3.4.

Each tournament could have participated in multiple tournaments of the same type. On average a strategy participated in 5690 different tournaments of each type. For example Tit For Tat has participated in a total of 5569 standard tournaments. The strategy's normalised ranks distribution is given in Figure 1. As a result, of the multiple entries of strategies their performance in this manuscript is evaluated based on the median normalised rank denoted as $\bar{r}$. Note that a value of $\bar{r} = 0$ means that a strategy the winner of the tournament where a value of $\bar{r} = 1$ means that the strategy came last.

Following the analysis of each tournament type, in Section 3.5 the results are merged and the strategies are evaluated in their overall performances over all the different tournaments.

3.1 Standard tournaments

The results presented in this section are based on $12,285$ different standard tournaments. Each tournament has on average 122 participants, a length of 100 turns and was repeated 55 times.

The top 15 performances in standard tournaments are given by Table 5. The 10 out of the 15 are strategies introduced in [14]. These have been trained using FSM (finite state automata), HMM (hidden markov models), ANN (artificial neural networks), lookup tables (LookerUp) and stochastic lookup tables (Gambler).

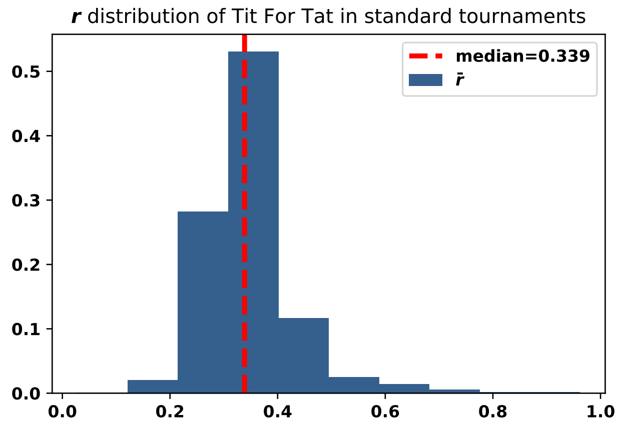

Figure 1: Tit For Tat's r distribution for standard tournaments.

Specifically, the have been trained using the strategy list of [2] in standard tournaments. Thus their performance was to be expected. DoubleCrosser, and Fool Me Once, are not from the literature but from [2]. DoubleCrosser is a strategy that makes use of the length of the match because is set to defect on the last two rounds. The strategy is expected to not perform as well in probabilistic ending tournaments. Finally, Winner 12 [15] and DBS are both from the the literature. DBS [6] is strategy specifically designed for noisy environments, however, it ranks highly only in standard ones.

Name	$\bar{r}$
Evolved HMM 5	0.00658
Evolved FSM 16	0.00990
EvolvedLookerUp2 2 2	0.01064
Evolved FSM 16 Noise 05	0.01639
PSO Gambler 2 2 2	0.02139
Evolved ANN	0.02874
Evolved ANN 5	0.03390
PSO Gambler 1 1 1	0.03723
Evolved FSM 4	0.04839
PSO Gambler Mem1	0.05000
Winner12	0.05946
Fool Me Once	0.06122
DBS	0.07087
DoubleCrosser	0.07190
BackStabber	0.07500

Table 5: Standard top performances

The distributions of $\bar{r}$ for top ranked strategies are given in Figure 2. The distributions are skewed towards zero and the highest median is at 0.075. These strategies would mostly rank high when they participated in a tournament. This is not true for the rest of the tournament types. In the following sections $\bar{r}$ and subsequently the performances are more variant.

3.2 Noisy tournaments

Similarly to Section 3.1 the results presented here are based on $12,285$ different noisy tournaments. The strategies have been ranked based on their $\bar{r}$ and the top 15 are given by Table 6. The distributions of their

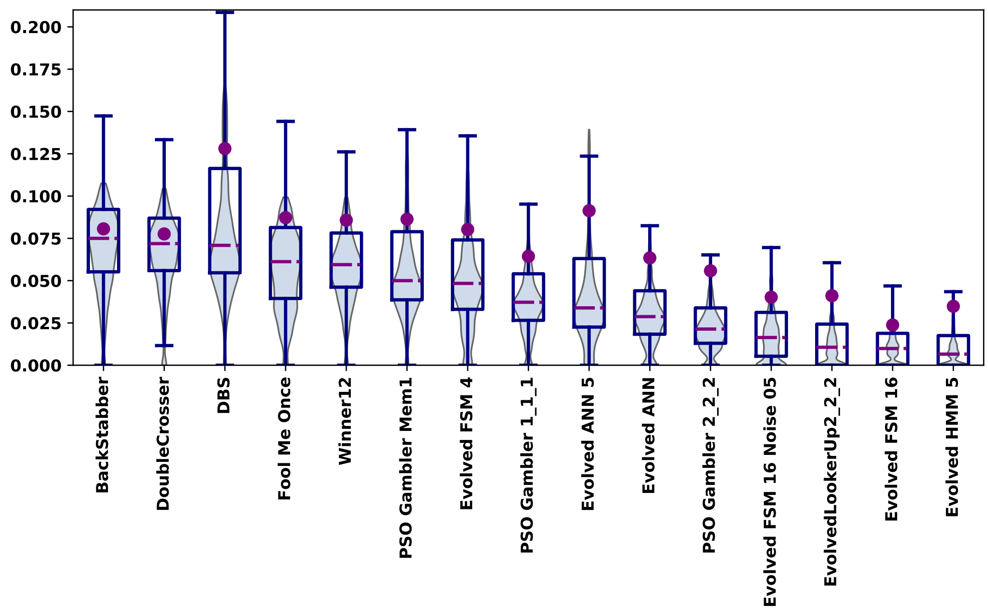

Figure 2: $\bar{r}$ distributions of top 15 strategies.

corresponding $\bar{r}$ is also given by Figure 3.

The top strategies include two well known strategies Tit For 2 Tats [8] and Hard Tit For 2 Tats [22] which are strategies that will defect only after the have received two defections from the opponent. The Retaliate strategies are set of strategies from [2] that start by cooperating but will retaliate once the opponent's wins and defections surpass a curtain threshold. Strategies ShortMem [12], Grumpy, e and ϕ are strategies that make decisions based on the cooperations to defections. Cycler Hunter tries to extort strategies that play cyclically and Risky QLearn uses a Q learning algorithm. Notably a very simple strategy, Cooperator, has been ranked third.

Name	$\bar{r}$
Grumpy	0.13953
e	0.19048
Cooperator	0.19565
Tit For 2 Tats	0.20520
Cycle Hunter	0.22222
Risky QLearner	0.22424
Retaliate 3	0.23077
Retaliate 2	0.23762
Retaliate	0.24309
Hard Tit For 2 Tats	0.24658
Limited Retaliate 3	0.25000
ShortMem	0.25272
Limited Retaliate	0.25698
Limited Retaliate 2	0.26027
ϕ	0.26201

Table 6: Noisy top performances

From Figure 2 it is evident that the normalised rank distributions in noisy environments are more variant and have higher median values compared to standard tournament. Moreover, the distributions are skewed both towards 0 and 1 which indicate that the top ranked strategies ranked first and last in several tournaments. However, the low values of the median shows that the have won most of the tournaments.

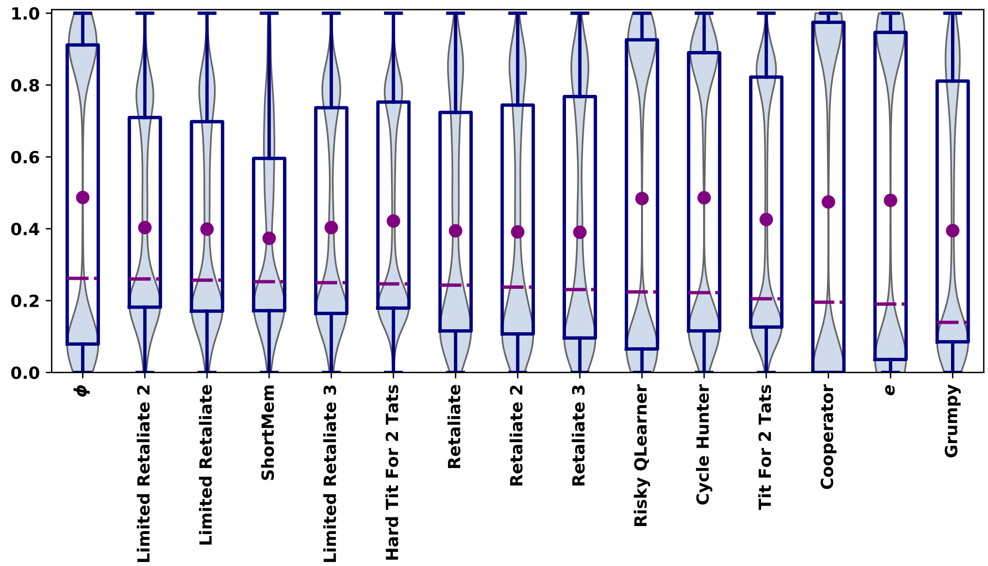

Figure 3: $\bar{r}$ distributions for best performed strategies in noisy tournaments.

3.3 Probabilistic ending tournaments

In this section the performances in probabilistic ending tournaments are discussed. The 15 top ranked strategies are given by Table 7 and their respective $\bar{r}$ distributions in Figure 4.

Fortress 3, Fortress 4 (both introduced in [4]), Raider [5] and Solution B1 [5] are strategies based on finite state automata introduced by Daniel and Wendy Ashlock. These strategies have been evolved using reinformed learning. However, there were trained to maximise their payoffs in tournaments with fixed turns, 150 specifically, and not in probabilistic ending tournaments.

Better and Better, Gradual Killer, Hard Prober are strategies from PRISON which is a research software tool for the prisoner's dilemma available at [1]. These, including Bully (Reverse Tit For Tat) [17], are strategies that are less cooperative strategies and more defective. Furthermore, Defector is a strategy which always defects. Interestingly, the last two strategies are EasyGo and Fool Me Forever. These strategies are actually the same. The will defect until their opponent defect, then they will cooperate until the end. Though we would have guessed that these two would have been taken advantage off both scored very highly and based on their distributions (Figure 4) there have managed to win tournaments.

The distributions of ranks in probabilistic ending tournaments are less variant than those of noisy tournaments. The uncertainty in the number of turns does not seem to be having the same effect. The medians are lower than 0.1 and the distributions are skewed towards 0. These suggest that the strategies performed well in the tournaments that they participated.

Name	$\bar{r}$
Fortress4	0.01266
Defector	0.01444
Better and Better	0.01587
Tricky Defector	0.01869
Fortress3	0.02198
Gradual Killer	0.02521
Aggravater	0.02797
Raider	0.03077
Cycler DDC	0.04545
Hard Prober	0.05085
SolutionB1	0.06040
Meta Minority	0.06040
Bully	0.06061
Fool Me Forever	0.07018
EasyGo	0.07065

Table 7: Probabilistic ending top performances

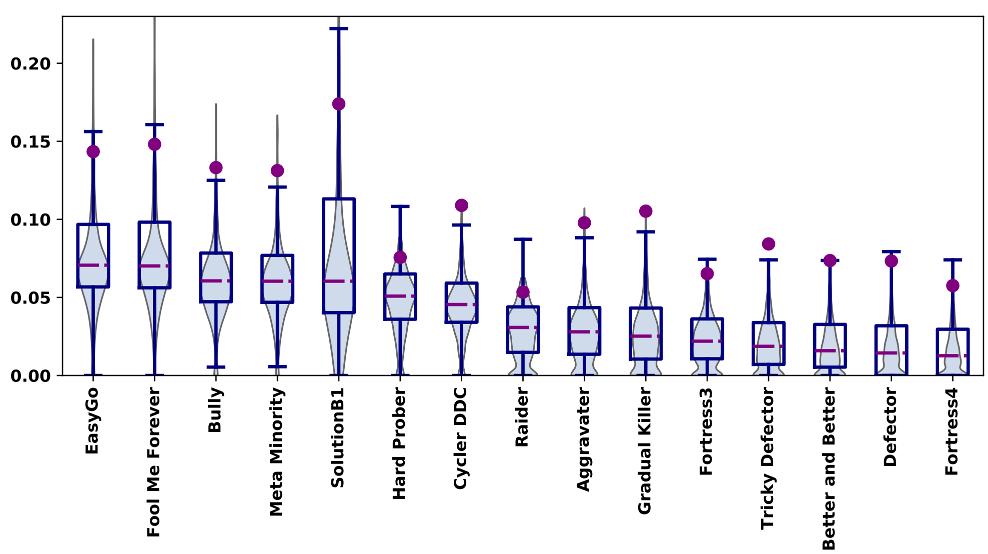

Figure 4: $\bar{r}$ distributions for best performed strategies in probabilistic ending tournaments.

3.4 Noisy & probabilistic ending tournaments

This section presents the results in tournaments were there are two uncertainties. There is the uncertainty that a player's action will be flipped and there is the uncertainty that there will not be future interactions. The 15 top ranked strategies are given in Table 8. Several of these strategies were highly ranked in noisy tournaments as well but the same does not hold for the strategies which were highly ranked in probabilistic ending tournaments.

Name	$\bar{r}$
Alternator	0.30390
ϕ	0.31025
e	0.31293
π	0.31818
Limited Retaliate	0.35294
Anti Tit For Tat	0.35429
Retaliate 3	0.35484
Limited Retaliate 3	0.35563
Retaliate	0.35588
Retaliate 2	0.35714
Limited Retaliate 2	0.36066
Hopeless	0.36913
Arrogant QLearner	0.40526
Cautious QLearner	0.40711
Risky QLearner	0.41989

Table 8: Noisy and probabilistic ending top performances

The top ranked strategy is Alternator a strategy that alternate between cooperation and defection. Alternator is followed by strategies that decide so that the ratio of cooperations and defections is close to the equivalent mathematical constants. Anti Tit For Tat is similar to Bully, which was covered in Section 3.2, but starts of with cooperation and Hopeless [23] a strategy that will only defect if and only if a mutual cooperation. The last spots are occupied by strategies based on the Q learning algorithm.

Compared to the distributions in noisy environments, Section 3.2, the distributions here are skewed towards the middle, Figure 5. The medians are close to 0.5 which seems to indicate a less dominant performance of strategies in this environment.

As yet, the performances of 186 different strategies have been evaluated in four different tournaments. In standard tournaments the top ranked spots were dominated by strategies that have been trained using reinforcement learning techniques and have been maximised in a standard tournament against the same list of opponents that is being used here. These strategies dominated most the tournaments that have been involved in. In noisy tournaments the top ranked strategies were all different to those of standard tournaments. Most of the top ranked strategies have been strategies that were trying to keep the defection and cooperation to a given ratio. These strategies had won most the tournaments in which they participated, however, they also ranked last in several tournaments. Once again in probabilistic ending tournaments the top ranked strategies were different to those before. More defecting strategies occupied the top ranks as well as finite state automata that have been introduced all by the same authors. Finally, in noisy probabilistic ending tournaments it was the only set of tournaments were the highly ranked strategies have been covered before, more specifically, in noisy tournaments. In these tournaments however the distributions of the ranks had much variation and some strategies even had a $\bar{r}$ close to 0.5, indicating that there were successfully only 50% of the time.

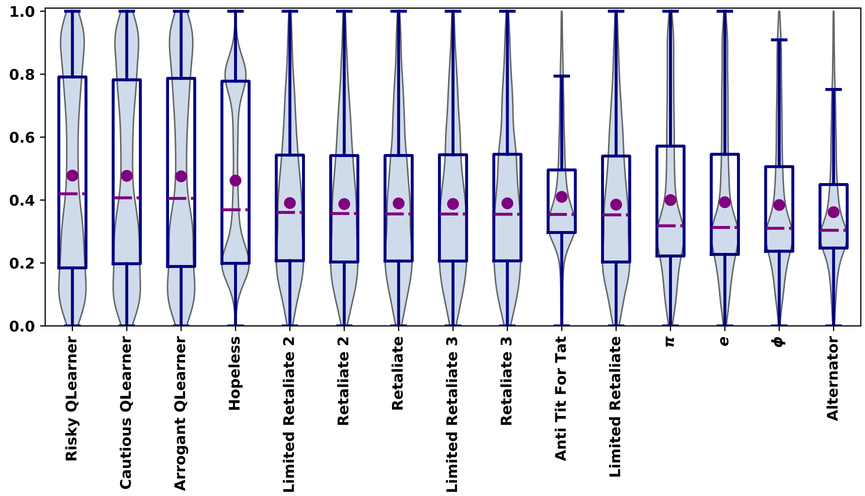

Figure 5: $\bar{r}$ distributions for best performed strategies in noisy probabilistic ending tournaments.

In the following section the performance of the strategies is evaluated again. However this time the results of the tournament types are included in the analysis.

3.5 Top performance in data set

The data set which was described in Section 2 contains results from 49,140 tournaments. The strategies have been ranked based on their $\bar{r}$ and the 15 top ranked strategies are given in Table 9. These include several strategies that have ranked in the top spots in the separate tournament analysis. The top 6 spaces are overtaken by Retaliate strategies, followed by BackStabber and DoubleCrosser. DoubleCrosser is just an extension of BackStabber. Nice Meta Winner and NMWE Memory One are strategies that are part of a team, Stein and Rapoport and Grudger are strategies from Axelrod's original tournament that came the 6th and 7th respectively. Forgeful Fool Me once is an extension to Grudger and PSO GAmer and Evolved are strategies again introduced in [14].

Name	$\bar{r}$
Limited Retaliate 3	0.285714
Retaliate 3	0.297872
Limited Retaliate 2	0.301370
Retaliate 2	0.304348
Limited Retaliate	0.309629
Retaliate	0.317073
BackStabber	0.322034
DoubleCrosser	0.327188
Nice Meta Winner	0.350000
PSO Gambler 2 2 2 Noise: 0.5	0.351104
Grudger	0.352941
Forgetful Fool Me Once	0.355140
NMWE Memory One	0.357576
Evolved HMM 5	0.358333
Stein and Rapoport	0.359375

Table 9: Top performances in data set

The distributions between the first six strategies are very similar. From then onwards the distributions are skewed to both 0 and 1. The analysis of the data set gives us an indication of what performance is successful in cases that the environment is not known to you and the Retaliate family seem to be doing the trick.

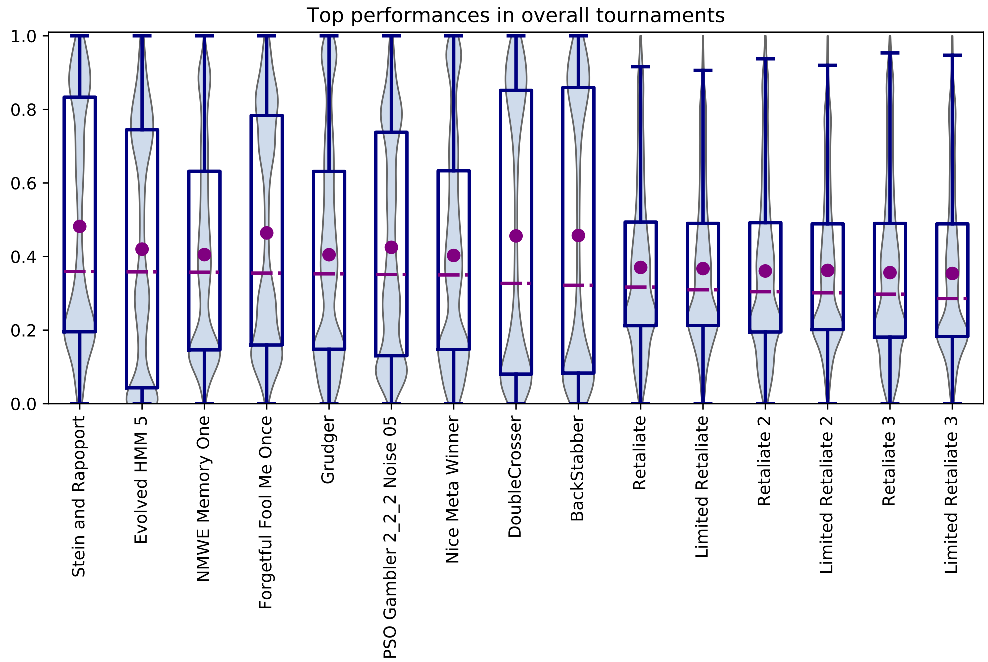

Figure 6: $\bar{r}$ distributions for best performed strategies in data set.

In this section the top ranked performances were presented. These one done just by simply taking into acount the ranks each strategy achieved in each tournament type and finally in the data set. Through the analysis only a few strategies seem to have been repeated, however, the winner were also completely different from type to type. Thus, what does make then successfully in those environments? The aim of the next section is that.

4 Evaluation of performance

5 Conclusion and Discussion

References

[1] Lifl (1998) prison. `http://www.lifl.fr/IPD/ipd.frame.html`. Accessed: 2017-10-23.

[2] The Axelrod project developers . Axelrod: 3.0.0, April 2016.

[3] Mark Aberdour. Achieving quality in open-source software. *IEEE software*, 24(1):58–64, 2007.

[4] Wendy Ashlock and Daniel Ashlock. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. In *2006 IEEE International Conference on Evolutionary Computation*, pages 297–304. IEEE, 2006.

[5] Wendy Ashlock, Jeffrey Tsang, and Daniel Ashlock. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 135–142. IEEE, 2014.

[6] Tsz-Chiu Au and Dana Nau. Accident or intention: that is the question (in the noisy iterated prisoner's dilemma). In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 561–568. ACM, 2006.

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

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

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

[10] J. Bendor, R. M. Kramer, and S. Stout. When in doubt... cooperation in a noisy prisoner's dilemma. *The Journal of Conflict Resolution*, 35(4):691–719, 1991.

[11] Fabien CY Benureau and Nicolas P Rougier. Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. *Frontiers in neuroinformatics*, 11:69, 2018.

[12] A Carvalho, H Rocha, F Amaral, and F Guimaraes. Iterated prisoners dilemma-an extended analysis. *Iterated Prisoners Dilemma-An extended analysis*, 2013.

[13] C. Donninger. *Is it Always Efficient to be Nice? A Computer Simulation of Axelrod's Computer Tournament*. Physica-Verlag HD, Heidelberg, 1986.

[14] Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoners dilemma. *PloS one*, 12(12):e0188046, 2017.

[15] Philippe Mathieu and Jean-Paul Delahaye. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation*, 20(4):12, 2017.

[16] P. Molander. The optimal level of generosity in a selfish, uncertain environment. *The Journal of Conflict Resolution*, 29(4):611–618, 1985.

[17] John H Nachbar. Evolution in the finitely repeated prisoner's dilemma. *Journal of Economic Behavior & Organization*, 19(3):307–326, 1992.

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

[19] Martin A Nowak and Karl Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355(6357):250, 1992.

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

[21] R. Selten and P. Hammerstein. Gaps in harley's argument on evolutionarily stable learning rules and in the logic of tit for tat. *Behavioral and Brain Sciences*, 7(1):115116, 1984.

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

[23] Pieter Van den Berg and Franz J Weissing. The importance of mechanisms for the evolution of cooperation. *Proceedings of the Royal Society B: Biological Sciences*, 282(1813):20151382, 2015.